

On the Algorithmic Linearizability of Nonlinear Ordinary Differential Equations

Dmitry A. Lyakhov

Computational Sciences Group, KAUST, Thuwal, Kingdom of Saudi Arabia

Vladimir P. Gerdt

*Group of Algebraic and Quantum Computations, Joint Institute for Nuclear Research, Dubna, Russian Federation;
Peoples' Friendship University of Russia (RUDN University), Moscow, Russian Federation*

Dominik L. Michels

*Computational Sciences Group, KAUST, Thuwal, Kingdom of Saudi Arabia;
Computer Science Department, Stanford University, Stanford (CA), United States of America*

Abstract

Solving nonlinear ordinary differential equations is one of the fundamental and practically important research challenges in mathematics. However, the problem of their algorithmic linearizability so far remained unsolved. In this contribution, we propose a solution of this problem for a wide class of nonlinear ordinary differential equation of arbitrary order. We develop two algorithms to check if a nonlinear differential equation can be reduced to a linear one by a point transformation of the dependent and independent variables. In this regard, we have restricted ourselves to quasi-linear equations with a rational dependence on the occurring variables and to point transformations. While the first algorithm is based on a construction of the Lie point symmetry algebra and on the computation of its derived algebra, the second algorithm exploits the differential Thomas decomposition and allows not only to test the linearizability, but also to generate a system of nonlinear partial differential equations that determines the point transformation and the coefficients of the linearized equation. The implementation of our algorithms is discussed and evaluated using several examples.

Keywords: Algorithmic linearization test, determining equations, differential Thomas decomposition, Lie symmetry algebra, ordinary differential equations, point transformation, power series solutions, Tremblay-Turbiner-Winternitz system.

Email addresses: dmitry.lyakhov@kaust.edu.sa (Dmitry A. Lyakhov), gerdt@jinr.ru (Vladimir P. Gerdt), dominik.michels@kaust.edu.sa, michels@cs.stanford.edu (Dominik L. Michels)

URL: <http://www.lyakhov.com/> (Dmitry A. Lyakhov), <http://compalg.jinr.ru/CAGroup/Gerdt/> (Vladimir P. Gerdt), <http://dmichels.de/> (Dominik L. Michels)

1. Introduction

The automation of solving nonlinear ordinary differential equations (ODEs) belongs to the most fundamental and practically important research challenges in mathematics and in the computational sciences. Such equations are typically solved numerically or by approximate analytical methods since obtaining their explicit solution is usually very difficult in practice or even impossible. One of the important approaches for solving a nonlinear ODE explicitly considers the existence of an invertible linearizing transformation of the variables and its construction. The reduction of a nonlinear ODE to a linear one makes its explicit integration much easier and often allows for obtaining an exact solution.

The linearization problem for a second-order ODE

$$y'' + f(x, y, y') = 0 \quad (1)$$

was solved by Lie (1883), who applied his general theory of integration of ODEs by means of a group of point transformations. He proved that f is at most cubic in y' for a linearizable equation and derived the necessary and sufficient conditions of linearizability. These conditions have the form of two explicit and easily verifiable equations (22) containing differential polynomials in the coefficients of f as a polynomial in y' :

$$f = F_3(x, y)(y')^3 + F_2(x, y)(y')^2 + F_1(x, y)y' + F_0(x, y). \quad (2)$$

Lie's ideas and methods were extended and applied to third-order equations $y''' = f(x, y, y', y'')$ by Ibragimov (2005) and later to fourth-order equations $y'''' = f(x, y, y', y'', y''')$ (Ibragimov (2008)). In these contributions, all possible structures of the candidates for the linearization were found and the explicit form of necessary and sufficient linearizability conditions of the coefficients of those structures were derived. Therefore, given an ODE of second, third or fourth order, to check whether it is linearizable by a point transformation or not, it is sufficient to verify whether the relevant explicit linearizability conditions are satisfied or not.

In practice, such a verification typically needs a computer-based symbolic algebraic computation for the simplification of the resulting expressions. An additional point to emphasize is that if the ODE contains parameters and/or arbitrary functions, then the linearizability conditions imply the algebraic and/or differential constraints on these parameters and/or functions that provide the linearization. Generally, however, these constraints may include the point transformation functions, and it may be highly conjectural to solve the constraints and to find a linearizing point transformation.

In this paper we suggest two algorithmic linearization tests applicable to a quasi-linear ODE¹ of any order greater or equal to two with a rational dependence on the other derivatives and the independent variable. The first linearization test is applicable to ODEs which do not contain parameters and arbitrary functions. This test is based on the construction of the Lie point symmetry algebra for the input ODE. The relevant mathematical methods are described in several textbooks (see, for example, Bluman (2001); Ibragimov (2009); Olver (1993, 1995); Ovsyannikov (1992); Schwarz (2008)). To detect linearizability we compute the maximal abelian dimension of the Lie symmetry algebra and make use of the results of Mahomed (1990). Unlike the first

¹Quasi-linear ODE means that it depends linearly on the highest order derivative, i.e. solvable with respect to its highest order derivative.

test, our second test exploits the differential Thomas decomposition (for more details, see Bachler (2012); Gerdt (2008); Li (1999); Robertz (2014); Thomas (1937, 1962); Wang (2001)), a universal algorithmic tool for the algebraic analysis of polynomially-nonlinear systems of partial differential equations (PDEs), and allows not only for the detection of linearizability but also for the derivation of necessary and sufficient linearizability conditions for arbitrary functions or parameters occurring in these equations. An example of such a problem is given by Eq. (1) whose linearizability conditions are given by Eq. (2).

Linearizability problems for nonlinear ODEs were studied by the prominent Russian mathematician Nail Ibragimov (2005, 2008), who contributed immeasurably to group analysis of differential equations and to the popularization of symmetry methods. He and his co-authors were able to obtain explicit formulas for linearizable equations by point and contact transformations, which could be reproduced automatically by our second algorithm. Sadly, we have to mention that during the writing of this paper he passed away. We dedicate this paper to his memory.

Second linearization algorithms gives more information and suitable for finding the transformation and the coefficients. However, the first linearization test is computationally more efficient and it is therefore advisable to apply it first when considering higher-order equations, and then, in the case of linearizability, apply the second test in order to construct the linearizing point transformation and the reduced linear form of the ODE.

This paper is an extended version of our recent contribution (Lyakhov (2017)). For our first linearizing test we presented there (Lyakhov (2017), Sect. 4), we now present one more implementation based on the algorithms described in the recent paper by Lisle (2017). It is implemented in the freely available Maple library LAVF (LIE ALGEBRA OF VECTOR FIELDS). Our benchmark shows (Sect. 5, Table 1) that the new implementation clearly outperforms the other two implementations. Moreover, we applied our second linearizing algorithm to the third-order ODE arising in the study (Gubbiotti (2017)) of the superintegrable Hamiltonian system known in the literature as the Tremblay-Turbiner-Winternitz system. In (Gubbiotti (2017)) it was shown that the ODE admits linearization by a certain point transformation. Our algorithm constructed another linearizing transformation, and we included this application of our second algorithm as an additional example (Ex. 12) to our Sect. 5).

This paper is organized as follows. In Sect. 2 we briefly describe the mathematical objects we deal with before presenting our algorithms in Sect. 3. Their implementation in Maple is then described in Sect. 4 and its application is illustrated in Sect. 5 using several examples including the Tremblay-Turbiner-Winternitz system. Finally, a conclusion is provided in Sect. 6.

2. Underlying Equations

In this paper we consider ODEs of the form

$$y^{(n)} + f(x, y, y', \dots, y^{(n-1)}) = 0, \quad y^{(k)} := \frac{d^k y}{dx^k}, \quad (3)$$

where f is a rational function of its arguments.² As additional arguments, the function f may also include parameters and/or arbitrary functions in x and/or y . Given an ODE of the form (3),

²We can also assume that f is a rational function of $(y', \dots, y^{(n-1)})$ with coefficients $\alpha(x, y)$ which satisfy some given polynomially nonlinear differential equations.

our aim is to check the existence of an invertible transformation³

$$u = \phi(x, y), \quad t = \psi(x, y) \quad (4)$$

which maps (3) into a linear n -th order homogeneous equation

$$u^{(n)}(t) + \sum_{k=0}^{n-1} a_k(t) u^{(k)}(t) = 0, \quad u^{(k)} := \frac{d^k u}{dt^k}. \quad (5)$$

The invertibility of (4) is provided by the inequation

$$J := \phi_x \psi_y - \phi_y \psi_x \neq 0, \quad (6)$$

where we use the standard notion for derivatives

$$\phi_x = \frac{\partial \phi}{\partial x}, \quad \psi_y = \frac{\partial \psi}{\partial y}.$$

If such a transformation exists for $n \geq 3$, then it can always be chosen (cf. Olver (1995), Thm. 6.54; Ibragimov (2009), Thm. 6.6.3) in a way that (5) takes the Laguerre-Forsyth normal form⁴

$$u^{(n)}(t) + \sum_{k=0}^{n-3} a_k(t) u^{(k)}(t) = 0. \quad (7)$$

A first-order ODE $y' = f(x, y)$ is always linearizable, but its linearization procedure is as hard as the integration of the equation (cf. Arnold (1992), Ch. 2, Thm. 1). For $n = 2$ any homogeneous linear equation

$$y''(x) + a(x) y'(x) + b(x) y(x) = 0$$

can be transformed by a substitution

$$t = \varphi(x), \quad \varphi'(x) \neq 0, \quad u = \sigma(x) y, \quad \sigma(x) \neq 0$$

to the simplest second order equation (Ibragimov (2009), Thm. 3.3.1)

$$u''(t) = 0. \quad (8)$$

One way to check the linearizability of Eq. (3) is to follow the classical approach by Lie (1883) to study the symmetry properties of Eq. (3) under the *infinitesimal* transformation

$$\tilde{x} = x + \varepsilon \xi(x, y) + \mathcal{O}(\varepsilon^2), \quad \tilde{y} = y + \varepsilon \eta(x, y) + \mathcal{O}(\varepsilon^2). \quad (9)$$

The *invariance condition* for Eq. (3) under the transformation (9) is given by the equality

$$\mathcal{X}(y^{(n)} + f(x, y, \dots, y^{(n-1)}))|_{y^{(n)} + f(x, y, \dots, y^{(n-1)})=0} = 0, \quad (10)$$

³Since we consider local linearizability, hereafter we assume that all functions are smooth in the vicinity of the linearization point.

⁴It is important to use the Laguerre-Forsyth normal form for reasons of computational efficiency of the algorithm **LinearizationTest II** due to finite dimensionality of the linearizing differential system (see Def. 6).

where the *symmetry operator* reads

$$X := \xi \partial_x + \sum_{k=0}^n \eta^{(k)} \partial_{y^{(k)}}, \quad \eta^{(k)} := D_x \eta^{(k-1)} - y^{(k)} D_x \xi, \quad (11)$$

$\eta^{(0)} := \eta$ and $D_x := \partial_x + \sum_{k \geq 0} y^{(k+1)} \partial_{y^{(k)}}$ is the total derivative operator with respect to x . Here we use the commonly accepted standard notation in group analysis (Ibragimov (2009), Sect. 1.4), where all differential functions depend only on a finite number of differential variables $y^{(k)}$, and

$$\partial_x = \frac{\partial}{\partial x}, \quad \partial_{y^{(k)}} = \frac{\partial}{\partial y^{(k)}}.$$

Example 1. As a toy example we consider the second-order trivial ODE

$$y''(x) = 0. \quad (12)$$

By a substitution of the arbitrary point transformation

$$\left[u = f(x, y), \quad t = g(x, y), \quad J = f_x g_y - f_y g_x \neq 0 \right],$$

it is transformed into

$$u''(t) + A_3 \cdot (u')^3 + A_2 \cdot (u')^2 + A_1 \cdot u' + A_0 = 0.$$

The symmetry property implies

$$A_3 = -\frac{\partial^2 g}{\partial y^2} \frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial y^2} \frac{\partial g}{\partial y} = 0,$$

$$A_2 = -\frac{\partial^2 g}{\partial y^2} \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x \partial y} \frac{\partial g}{\partial y} + 2 \frac{\partial^2 f}{\partial y \partial y} \frac{\partial g}{\partial x} - 2 \frac{\partial^2 g}{\partial x \partial y} \frac{\partial f}{\partial y} = 0,$$

$$A_1 = -\frac{\partial^2 g}{\partial x^2} \frac{\partial f}{\partial y} + \frac{\partial^2 f}{\partial x^2} \frac{\partial g}{\partial y} + 2 \frac{\partial^2 f}{\partial x \partial y} \frac{\partial g}{\partial x} - 2 \frac{\partial^2 g}{\partial x \partial y} \frac{\partial f}{\partial x} = 0,$$

$$A_0 = -\frac{\partial^2 g}{\partial x^2} \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2} \frac{\partial g}{\partial x} = 0.$$

Instead of searching for the Lie group itself we rewrite this system in terms of generators of the one-parameter group of symmetry transformations (with group parameter a):

$$g(x, y) := x + a \cdot \xi(x, y) + o(a),$$

$$f(x, y) := y + a \cdot \eta(x, y) + o(a).$$

Substitution to symmetry conditions and differentiation by a at $a = 0$ leads to the linear PDE system

$$\frac{\partial^2 \eta}{\partial x^2} = 0, \quad -\frac{\partial^2 \xi}{\partial x^2} + 2 \frac{\partial^2 \eta}{\partial x \partial y} = 0, \quad \frac{\partial^2 \xi}{\partial y^2} = 0, \quad -\frac{\partial^2 \eta}{\partial y^2} + 2 \frac{\partial^2 \xi}{\partial x \partial y} = 0,$$

which can be easily solved by hand:

$$\begin{aligned}\xi(x, y) &= (C_7 \cdot x + C_8) \cdot y + C_5 \cdot x^2 + C_3 \cdot x + C_4, \\ \eta(x, y) &= (C_5 \cdot y + C_6) \cdot x + C_7 \cdot y^2 + C_1 \cdot y + C_2.\end{aligned}$$

It corresponds to the 8-parameter Lie group of symmetries, which contains all linear mappings of plane, projections, and shifts. Geometrically, the symmetry group represents all possible transformations of the set of straight lines on a plane into itself that is given by solutions of Eq. (12).

The invariance condition (10) means that its left-hand side vanishes when Eq. (3) holds. Then the application of (11) to the left-hand side of Eq. (3) and the substitution of

$$y^{(n)} = -f(x, y, \dots, y^{(n-1)})$$

in the resulting expression leads to the equality $g = 0$ with the polynomial dependence of g on the derivatives $y', \dots, y^{(n-1)}$. Since, by the transformation (9), the functions ξ and η do not depend on these derivatives, the equality $g = 0$ holds if and only if all coefficients in $y', \dots, y^{(n-1)}$ are equal to zero. This leads to an overdetermined system of linear PDEs in ξ and η called *determining system*. Its solution yields a set of symmetry operators whose cardinality we denote by m . This set forms a basis of the m -dimensional Lie symmetry algebra

$$[\mathcal{X}_i, \mathcal{X}_j] = \sum_{k=1}^m C_{i,j}^k \mathcal{X}_k, \quad 1 \leq i < j \leq m. \quad (13)$$

We denote the Lie symmetry algebra by L and $m = \dim(L)$. Its *derived algebra* $L' \subset L$ is a subalgebra that consists of all commutators of pairs of elements in L .

It was shown (Lie (1883), Ch. 12, p. 298, ‘‘Satz’’ 3) that the Lie point symmetry algebra of an n -order ODE has a dimension m satisfying

$$n = 1, m = \infty; \quad n = 2, m \leq 8; \quad n \geq 3, m \leq n + 4.$$

Later, the interrelations between n and m were established that provide the linearizability of (3) by a point transformation (4) in the absence of parameters and arbitrary functions. Here we present the two theorems that describe such interrelations and form the basis of our first linearization test.

Theorem 2. (Mahomed (1990), Thm. 1) *A necessary and sufficient condition for the linearization of (3) with $n \geq 3$ via a point transformation is the existence of an abelian n -dimensional subalgebra of (13).*

Theorem 3. (Mahomed (1990), Sect. 2–4 and Thms. 6,8; Schwarz (2008), Thm. 5.19)⁵ *Eq. (3) with $n \geq 2$ is linearizable by a point transformation if and only if one of the following conditions is fulfilled:*

1. $n = 2, m = 8$;
2. $n \geq 3, m = n + 4$;

⁵Cf. Thm. 6.39 in (Olver (1995)) regarding $n = 2$.

3. $n \geq 3$, $m \in \{n+1, n+2\}$ and (13) admits an abelian subalgebra of dimension n .

These theorems show that the verification of the third condition requires, in addition to the determination of m , a computation to check the existence of an abelian Lie symmetry subalgebra of dimension n . To our knowledge, there is only an algorithm described in the literature (Ceballos (2012)) for the computation of the *maximal abelian dimension*, i.e. dimension of the maximal abelian subalgebra of a finitely-dimensional Lie algebra given by its structure constants. The algorithm is reduced to solving the quadratically nonlinear system of multivariate polynomial equations providing vanishing of the Lie bracket between two arbitrary vectors in the Lie algebra. Clearly, the computing time of the algorithm is at least exponential in the dimension m of the algebra.

Instead, to verify the third condition in Thm. 3 we devise a much more efficient algorithm. Our algorithm relies on the following statement which is a corollary to Thms. 2 and 3.

Corollary 4. (Lyakhov (2017)) The third condition is equivalent to

3'. $n \geq 3$, $m \in \{n+1, n+2\}$ and the derived algebra of (13) is abelian of dimension n .

Proof. Under the third condition, since $L' \subset L$, Eq. (3) is linearizable by Thm. 1. Let Eq. (3) be linearizable. The symmetry Lie algebra of (5) is *similar* and hence *isomorphic* to that of (3) (cf. Ovsyannikov (1992), Ch. 2, §7.9). It is easy to see that a linear n -th order equation (5) with variable coefficients admits the Lie point symmetry group

$$\{\bar{t} = t, \bar{u} = u + c_i \cdot v_i(t) \ (i = 1, \dots, n), \bar{u} = c_{n+1} \cdot u\}, \quad (14)$$

where c_i, c_{n+1} are constants (the group parameters) and $v_i(t)$ are the fundamental solutions of (5). The Lie group (14) has the $(n+1)$ -dimensional Lie algebra (cf. Schwarz (2008), Thm. 5.19)

$$L_{n+1} := \{\mathcal{X}_i := v_i(t) \partial_u \ (i = 1, \dots, n), \mathcal{X}_{n+1} := u \partial_u\}. \quad (15)$$

If a linear n -th order Eq. (5) has constant coefficients, then in addition to (15) the Lie point symmetry group (14) includes the translation $\bar{t} = t + c_{n+2}$ and, hence, its Lie algebra, in addition to (15), has one more element:

$$L_{n+2} := L_{n+1} \cup \{\mathcal{X}_{n+2} := \partial_t\}. \quad (16)$$

Furthermore, $[\mathcal{X}_{n+1}, \mathcal{X}_{n+2}] = 0$, and for all $i \in \{1, \dots, n\}$:

$$[\mathcal{X}_i, \mathcal{X}_{n+1}] = \mathcal{X}_i, \quad [\mathcal{X}_i, \mathcal{X}_{n+2}] = -v_i'(t) \partial_u = \sum_{j=1}^n \alpha_j \mathcal{X}_j,$$

where α_j are constants. Therefore, both Lie algebras (15) and (16) have abelian derived algebras of dimension n . \square

It is important to emphasize that m can be algorithmically computed without solving the determining system what is generally impossible. To do such computation it is necessary first to complete the last system to involution (for the theory of completion to involution we refer to the paper of Seiler (2010)). Then one can apply Algorithm 3 by Lisle (2017) or use the algorithmic ideas of Reid (1991a,b) based on construction of power series solutions to determining system.

We exploit both these approaches in our implementation (Sect. 4) of the algorithm **LinearizationTest I** described in Sect. 3.1. In doing so, in our implementation of the first approach we use the Maple library LAVF (see Sect. 4, Item 1). To implement the second approach (see Sect. 4, Item 2) we use the *differential Thomas decomposition* (Bachler (2012); Robertz (2014)) for a *degree-reverse lexicographical ranking* to complete the determining system to involution and then we compute the *differential dimension polynomial* (Lange-Hegermann (2014)) for the output Janet basis.

The differential Thomas decomposition was suggested in the papers of Thomas (1937, 1962) as a generalization of the Riquier-Janet theory of *passive* linear and *orthonomic* PDE systems (see also Seiler (2010) and the references therein) to polynomially-nonlinear systems of general form. The Thomas decomposition provides a universal algorithmic tool (Bachler (2012); Robertz (2014)) to study a *differential system*, which is defined as follows.

Definition 5. (Bachler (2012); Gerdt (2008); Li (1999); Robertz (2014); Thomas (1937, 1962); Wang (2001)) A differential system is a system $S := \{S^=, S^\neq\}$ of differential equations and (possibly) inequations of the form

$$S^= := \{g_1 = 0, \dots, g_s = 0\}, S^\neq := \{h_1 \neq 0, \dots, h_t \neq 0\},$$

where s is a positive integer as well as t if $S^\neq \neq \emptyset$, and g_i, h_j are elements in the differential polynomial ring in finitely many differential indeterminates (dependent variables) over the differential field of characteristics zero.

The Thomas decomposition applied to a differential system S yields a finite set of passive (involution) and *differentially triangular* differential systems called *simple* (see Bachler (2012); Gerdt (2008); Li (1999); Robertz (2014); Thomas (1937, 1962); Wang (2001)) that partition the solution set of the input differential system. Algebraically, this provides a characterizable decomposition (Hubert (2003)) of the radical differential ideal $\sqrt{\mathcal{I}}$ where \mathcal{I} is the differential ideal generated by the polynomials in $S^=$.

Unlike the **LinearizationTest I** where one can use, due to the linearity of determining systems, any procedure of completion to involution (e.g. the standard form algorithm of Reid (1991a)), our second algorithm **LinearizationTest II** (Sect. 3.2) is oriented to the Thomas decomposition.

To apply it, we need to formulate the conditions for the functions $\phi(x, y)$, $\psi(x, y)$ in (4) and for the coefficients $a_k(t)$ in (7) (if $n \geq 3$) such that these conditions hold if and only if (3) is linearizable. In addition to the input differential system, the Thomas decomposition is determined by a *ranking*, that is, a linear ordering on the partial derivatives compatible with derivations (Bachler (2012); Gerdt (2008); Li (1999); Robertz (2014); Thomas (1937, 1962); Wang (2001)) (in our case with ∂_x and ∂_y).

By differentiating the equality $u(\psi(x, y(x))) = \phi(x, y(x))$, that follows from (4), n times with

respect to x , we obtain the following equations:

$$\begin{aligned}
u'(t) &= \frac{\phi_x + \phi_y y'}{\psi_x + \psi_y y'}, \\
u''(t) &= \frac{J}{(\psi_x + \psi_y y')^3} y'' + \frac{P_2(y')}{(\psi_x + \psi_y y')^3}, \\
&\vdots \\
u^{(n)}(t) &= \frac{J}{(\psi_x + \psi_y y')^{n+1}} y^{(n)} + \frac{P_n(y', \dots, y^{(n-1)})}{(\psi_x + \psi_y y')^{2n-1}}.
\end{aligned} \tag{17}$$

Here J is the Jacobian (6), P_k ($k = (2, \dots, n)$) are polynomials in their arguments whose coefficients are differential polynomials in ϕ and ψ , for example,

$$P_2(y') = (\psi_x + \psi_y y')(\phi_{xx} + \phi_{xy} y' + \phi_{yy} (y')^2) - (\phi_x + \phi_y y')(\psi_{xx} + \psi_{xy} y' + \psi_{yy} (y')^2).$$

Now we replace the derivatives $u^{(k)}$ occurring in (7) (or the second-order derivative in (8) if $n = 2$) with the appropriate right-hand sides in Eqs. (17) and solve the obtained equality with respect to $y^{(n)}$ (or y''). As a result, we obtain the equality

$$y^{(n)} + \frac{R(y', \dots, y^{(n-1)})}{J \cdot (\psi_x + \psi_y y')^{n-2}} = 0, \tag{18}$$

where R is a polynomial in $y', \dots, y^{(n-1)}$ whose coefficients for $n \geq 3$ are the differential polynomials not only in ϕ and ψ but also in $a_k(t) = a_k(\psi(x, y(x)))$, the coefficients in Eq. (7).

Denote by M and N the numerator and denominator of the function f in Eq. (3). Then, after elimination of $y^{(n)}$ from the equation system (3), (18) and clearing denominators in the rational functions of the obtained equality we obtain equation

$$R \cdot N - J \cdot M \cdot (\psi_x + \psi_y y')^{n-2} = 0. \tag{19}$$

This equation is a polynomial in $y', y'', \dots, y^{(n-1)}$, and there are no constraints on these variables. Therefore, the equation holds if and only if all coefficients of the polynomial in the left-hand side vanish. This condition gives a partial differential equation system in ϕ , ψ and a_k . If the function f in Eq. (3) depends on parameters and/or undetermined functions in (x, y) , then Eq. (19) contains these parameters/functions.⁶

Let $S^=$ be the set of equations obtained from Eq. (19) by equating the coefficients of the polynomial (in $y', \dots, y^{(n-1)}$) in the left-hand side to zero. If $n \geq 3$ we enlarge $S^=$ with the set of equations

$$S^= = S^= \cup_{k=0}^{n-3} \{\psi_y(a_k)_x - \psi_x(a_k)_y = 0\}. \tag{20}$$

The equation $\psi_y(a_k)_x - \psi_x(a_k)_y = 0$ means that a_k is a function of t in accordance to (7). It is easy to see by differentiating the equality $a_k(t) = a_k(\psi(x, y))$ as follows:

$$(a_k)_x = (a_k)_t \psi_x, \quad (a_k)_y = (a_k)_t \psi_y, \quad \iff \quad \psi_y(a_k)_x - \psi_x(a_k)_y = 0.$$

⁶One can always consider parameters as functions in x and y with zero derivatives.

Since we admit the invertible transformations (4) only, one has to add to the enlarged equation set $S^=$ the inequation

$$S^\neq := \{J \neq 0\} \quad (21)$$

where J is the Jacobian (6).

Thereby, the main object of our construction and the statements on its relation to the linearization are given as follows.

Definition 6. *The differential system (see Def. 5) made up of the above constructed PDE set $S^=$ and of the inequation set $S^\neq = \{J \neq 0\}$ will be called linearizing differential system.*

The linearization of the differential system follows a direct procedure of transformation to a linear equation (if $J \neq 0$). It is clear that its consistency implies the linearizability criterion.

Theorem 7. *Eq. (3) is linearizable via a point transformation (4) if and only if the linearizing differential system is consistent, i.e. has a solution.*

Corollary 8. *Eq. (3) is nonlinearizable via a point transformations (4) if and only if the result of the Thomas decomposition algorithm (Bachler (2012), Alg. 2.25; Robertz (2014), Alg. 2.2.56) applied to the linearizing differential system is the empty set.*

Proof. Bachler (2012), Remark 2.3 and Robertz (2014), Remark 2.2.58. □

3. Linearization Tests

In this section we present our algorithms **LinearizationTest I** and **LinearizationTest II**. These algorithms, given an input equation (3), verify its linearizability by the point transformation (4). In so doing, the first test is applicable only to an ODE without parameters and undetermined functions in the variables x and y . The second algorithm admits a rational dependence of the function f in Eq. (3) on such parameters and functions.

3.1. Linearization Test I

Our first test, presented below, is based on the computation of the Lie symmetry algebra and its analysis. In line 2 we compute the determining system for (3). It is the straightforward procedure outlined in the preceding section and described in most textbooks on Lie symmetry analysis, in particular in (Bluman (2001); Ibragimov (2009); Olver (1993, 1995); Ovsyannikov (1992); Schwarz (2008)). As a routine, this procedure is present in most computer algebra packages specialized to such an analysis, for example, in the Maple packages DESOLV (Carminati (2000)), DESOLVII (Vu (2012)), and SADE (Filho (2011)).

Since the determining system is linear, one can use any algorithm for its completion to involution in line 3, (cf. Reid (1991a); Seiler (2010), Sect. 10.7). However, we prefer to use the differential Thomas algorithm here (Bachler (2012), Sect. 3; Robertz (2014) Sect. 2.2).

The dimension m of the Lie algebra (13) (line 4) is the dimension of the solution space of the determining system and can be computed in several ways (cf. Seiler (2010), Sect. 8.2 and 9.3). Having computed the Janet involutive form of the determining system, it is easy to compute the dimension of its solution via an algorithmic construction of the *differential dimension polynomial* (Lange-Hegermann (2014)).

We refer to [papers](#) (Lisle (2017); Reid (1991b)) for the subalgorithm providing computation of the Lie symmetry algebra (line 8), i.e. for the computation of the structure constants $C_{i,j}^k$ in

Algorithm 1 *LinearizationTest I*(q)

Input: q , a nonlinear differential equation of form (3)

Output: True, if q is linearizable and False, otherwise

```
1:  $n := \text{order}(q)$ ;  
2:  $DS := \text{DeterminingSystem}(q)$ ;  
3:  $IDS := \text{InvolutiveDeterminingSystem}(DS)$ ;  
4:  $m := \text{dim}(\text{LieSymmetryAlgebra})(IDS)$ ;  
5: if  $n = 1 \vee (n = 2 \wedge m = 8) \vee (n > 2 \wedge m = n + 4)$  then  
6:   return True;  
7: else if  $n > 2 \wedge (m = n + 1 \vee m = n + 2)$  then  
8:    $L := \text{LieSymmetryAlgebra}(IDS)$ ;  
9:    $DA := \text{DerivedAlgebra}(L)$ ;  
10:  if  $DA$  is abelian and  $\text{dim}(DA) = n$  then  
11:    return True;  
12:  end if  
13: end if  
14: return False;
```

Algorithm 2 *LinearizationTest II*(q, P, H)

Input: q , a nonlinear differential equation of form (3) of order ≥ 2 ; P , a set of parameters; H , a set of undetermined functions in (x, y)

Output: set G of differential systems for functions ϕ and ψ in (4) and (possibly) in elements of P and H if (3) is linearizable, and the empty set, otherwise

```
1:  $n := \text{order}(q)$ ;  
2:  $G := \emptyset$ ;  
3:  $M := \text{numerator}(f)$ ;  $N := \text{denominator}(f)$ ;  
4:  $J := \phi_x \psi_y - \phi_y \psi_x$ ; {Jacobian (6)}  
5: if  $n = 2$  then  
6:    $r := u''(t) = 0$ ; {ODE (8)}  
7:    $A := \emptyset$ ;  
8: else  
9:    $r := u^{(n)}(t) + \sum_{k=0}^{n-3} a_k(t)u^{(k)}(t) = 0$ ; {ODE (7)}  
10:   $A := \{a_0, \dots, a_{n-3}\}$ ;  
11: end if  
12:  $r \xrightarrow{\text{by (4)}} y^{(n)} + \frac{R(y', \dots, y^{(n-1)})}{J \cdot (\psi_x + \psi_y y')^{(n-2)}} = 0$ ; {Eq. (18)}  
13:  $T := R \cdot N - M \cdot J \cdot (\psi_x + \psi_y y')^{(n-2)} = 0$ ; {Eq. (19)}  
14:  $S^\square := \{c = 0 \mid c \in \text{coeffs}(T, \{y', \dots, y^{(n-1)}\})\}$ ;  
15:  $S^\square := S^\square \cup_{p \in P} \{p_x = 0, p_y = 0\}$ ;  
16:  $S^\square := S^\square \cup_{a \in A} \{a_x \psi_y - a_y \psi_x = 0\}$ ; {Eq. (20)}  
17:  $S^\neq := \{J \neq 0\}$ ; {Ineq. (21)}  
18:  $G := \text{ThomasDecomposition}(S^\square, S^\neq)$ ;  
19: return  $G$ ;
```

Eq. (13). The last subalgorithm **DerivedAlgebra** in line 9 does the straightforward computation of the derived algebra via the structure constants.

Correctness and termination. For the subalgorithms both these properties are either obvious (as for **DerivedAlgebra**) or shown in the papers we referred to in the description of the subalgorithms above. Therefore, the whole algorithm **LinearizationTest I** terminates, and its correctness is provided by Thms. 2 and 3, and Cor. 4.

3.2. Linearization Test II

Our second test is based on the differential Thomas decomposition (Bachler (2012); Robertz (2014)). It admits the rational dependence of Eq. (3) on a finite set of parameters (constants) and/or undetermined functions in (x, y) . In the absence of parameters/functions the corresponding sets are inputted as the empty ones.

In lines 3–17 of the algorithm **LinearizationTest II** the input linearizing differential system (Def. 6) is constructed for the Thomas decomposition computed in line 18. This construction is done in correspondence with the formulas (6)–(7), (8), and (18)–(21).

Furthermore, if the output set of the Thomas decomposition is nonempty, then Eq. (3) is linearizable by Thm. 7. In this case the simple systems in the decomposition provide a partition of the solution space of the linearizing differential system and their solutions determine the invertible point transformation (4) and the coefficients $a_k(t)$ of the linearized form (7) or (8). In addition, if there are parameters and/or undetermined functions in (3), then the output differential systems of the Thomas decomposition provide the compatibility conditions to these parameters/functions imposed by the linearization.

Correctness and termination are provided by those of the Thomas decomposition (Bachler (2012), Sect. 3.4; Robertz (2014), Thr. 2.2.57).

4. Implementation

We implemented both linearization tests in Maple and our implementation runs on version 16 and the current 2017 version.

First, we describe three different implementations of the algorithm **LinearizationTest I**.

1. For the generation of the determining system DS (line 2) we use the routine *Determining-PDE* of the package `PDETools` built-in Maple. By doing so, the output system is reduced taking into account its integrability conditions. Hence, we can skip the completion of the determining system to involution (line 3). Furthermore, to determine the Lie symmetry algebra for the input ODE we use the Maple library `LAVF` implementing algorithms described by Lisle (2017). These algorithms improve those in (Reid (1991a,b)). The library `LAVF` is freely downloadable from the web page indicated in the footnote of page 497 of Lisle (2017) and consists of three packages: `VectorFields`, `LinearHomogeneousPDE` and `LieAlgebrasOfVectorFields`. Computation of the Lie symmetry algebra is done by invoking the command `LieAlgebrasOfVectorFields:-LAVF`. In order to construct the derived algebra (line 9) we invoke the routine *DerivedAlgebra* which is a part of the built-in package `DifferentialGeometry:-LieAlgebras`. To check whether the computed Lie algebra is abelian we used the built-in Maple routine `IsAbelian` included in the package `GroupTheory`.

2. To generate the determining system in line 2, we use the routine *gendef* of the Maple package DESOLV (Carminati (2000); Vu (2012)). Then, to complete the system *DS* to involution (line 3), we choose the *orderly* (“*DegRevLex*”) ranking (cf. Robertz (2014), Def. A.3.2) such that

$$\partial_x > \partial_y, \quad \xi > \eta,$$

and apply the routine *DifferentialThomasDecomposition* of the Maple package `DifferentialThomas`. This package `DifferentialThomas` (2012) is freely available. To compute the dimension of the Lie symmetry algebra (line 4), we invoke the routine *DifferentialSystemDimensionPolynomial*. Since in our case the solution space of the determining system is finitely dimensional, the last routine outputs just the dimension of the solution space. We implemented the subalgorithm **LieSymmetryAlgebra** computing the structure constants of the Lie symmetry by using the algorithmic approach developed in (Reid (1991a,b)). Our implementation takes the Janet involutive form of the determining system outputted by the package `DifferentialThomas` and exploits its routine *PowerSeriesSolution*.

3. This implementation differs from the previous one by the replacement of the subalgorithm **DerivedAlgebra** (in line 9) with the Maple implementation of Ceballos (2012) for the detection of a n -dimensional abelian subalgebra of the Lie symmetry algebra (line 8). By Thm. 2, the existence of such a subalgebra yields the criterion of linearization.

The rows in the table marked with **I** show the timings measured with three different implementations of **LinearizationTest I**. In the row marked with **I₁** we show the computation times based on the use of the Maple package LAVF by Lisle (2017) for the computation of the Lie symmetry algebra.

In our implementation of **LinearizationTest II** we compute the expressions (17) to obtain the left-hand side in (19) (line 13) that is a polynomial in $y', y'', \dots, y^{(n)}$. Then equating of all coefficients in the polynomial to zero (line 14) and enlarging it with additional equations (lines 15 and 16) and the Jacobian inequation (line 17) yields the input linearizing differential system for the subroutine *DifferentialThomasDecomposition* (line 18). By default, we choose the orderly ranking on the partial derivatives of the functions ϕ and ψ , and of those in the sets A (line 10):

$$\partial_x > \partial_y, \quad \xi > \eta > a_0 > \dots > a_{n-2}.$$

If the input ODE (3) contains (nonempty) sets of parameters and/or undetermined functions, then their rankings are less than those of a_{n-2} in order to derive the compatibility conditions for parameters and functions.

The comparative efficiency of the first algorithm could also be understood by means of complexity theory. The first algorithm in comparison to the second one is dealing with a fixed number of dependent and independent variables. It is well known that the complexity of completion algorithms in differential algebra grows drastically as the number of variables increases (Grigoriev (2009); Gustavson (2018)).

5. Examples

In this section we demonstrate our algorithmic linearization tests using several examples. All timings given below were obtained with Maple 2018⁷ running on a desktop computer with an Intel(R)Xeon(R) CPU E5-2687W v4 CPU clocked at 3.00 GHz and 128 GB RAM.

⁷Since its 2018 version, the Differential Thomas tool is available as a built-in package of the standard Maple distribution (Gerdt (2019)).

Example 9. (Lie (1883)) Consider the second-order Eq. (1) in which f is given by (2) with undetermined functions F_k , $0 \leq k \leq 3$. Algorithm **LinearizationTest I** is not applicable to this case, so we apply the algorithm **LinearizationTest II** with block elimination ranking such that

$$\partial_x > \partial_y, \quad \phi > \psi \gg F_3 > F_2 > F_1 > F_0.$$

Then the routine `DifferentialThomasDecomposition` of the package `DifferentialThomas` (2012) outputs three differential systems with disjoint solutions space in about 0.3 sec.:

$$S_1 := \{S_1^{\bar{=}}, S_1^{\neq}\}, \quad S_2 := \{S_2^{\bar{=}}, S_2^{\neq}\}, \quad S_3 := \{S_3^{\bar{=}}, S_3^{\neq}\}.$$

Cor. 8 guarantees that there are linearizable equations among the equations in family (1)–(2). One of the output differential systems, namely S_1 , is a generic simple system (see Robertz (2014), Def. 2.2.67). It has eight equations, and the last two of them that contain solely functions F_0, F_1, F_2, F_3 are the compatibility conditions for these functions whose solutions admit linearization. These conditions have the following form:

$$\begin{aligned} & 3(F_3)_{xx} - 2(F_2)_{xy} + (F_1)_{yy} - 3F_1(F_3)_x + 2F_2(F_2)_x \\ & - 3F_3(F_1)_x + 3F_0(F_3)_y + 6F_3(F_0)_y - F_2(F_1)_y = 0, \\ & (F_2)_{xx} - 2(F_1)_{xy} + 3(F_0)_{yy} - 6F_0(F_3)_x + F_1(F_2)_x \\ & - 3F_3(F_0)_x + 3F_0(F_2)_y + 3F_2(F_0)_y - 2F_1(F_1)_y = 0. \end{aligned} \quad (22)$$

These equations are exactly the linearizability conditions for (1)–(2) obtained by Lie (1883) (also see cf. Ibragimov (2009), Thm. 6.5.2). The inequations in the S_1 system are

$$S_1^{\neq} = \{J \neq 0, \psi_x \neq 0, \psi_y \neq 0\}.$$

The two other differential systems S_2 and S_3 have the following inequations:

$$S_2^{\neq} = \{\phi_x \neq 0, \psi_y \neq 0\}, \quad S_3^{\neq} = \{\phi_y \neq 0, \psi_x \neq 0\}. \quad (23)$$

Each of these systems has eight equations as S_1 . Every equation in S_2 as well as in S_3 is valid on all common solutions to the equations in S_1 (cf. Robertz (2014), Cor. 2.2.66). In doing so,

$$\psi_x = 0 \in S_2^{\bar{=}}, \quad \psi_y = 0 \in S_3^{\bar{=}}, \quad (24)$$

and hence each of (23) and (24) implies $J \neq 0$. Therefore, algorithm **LinearizationTest II** reproduces Lie's classical results on the necessary and sufficient conditions for the linearization of the second-order ODEs from family (1)–(2).

Example 10. (Euler (2003), Eq. 2.50) The third order ODE

$$y''' - 6\frac{y'}{x^2} + 3\frac{(y')^2}{x} - \frac{1}{2}(y')^3 = 0. \quad (25)$$

is linearizable by the generalized Sundman transformation (a kind of a nonlocal transformation, which is not a point one, in general). Here we check its linearizability via the point transformation (4). Eq. (25) admits both our tests since it does not contain parameters and/or undetermined functions. Our implementation of algorithm **LinearizationTest I** returns false in 0.05 sec. and that of **LinearizationTest II** returns the empty set in 0.3 sec.

Example 11. We consider the fourth-order ODE

$$2x^2y y'''' + x^2y^2 + h(x,y)y'y'' + 16xy y'' + 6x^2(y'')^2 + 48xy'y'' + 24yy'' + 24(y')^2 = 0, \quad (26)$$

where $h(x,y)$ is an undetermined function. To find all values of this function providing linearization, we again apply algorithm **LinearizationTest II**. The package `DifferentialThomas` for the orderly ranking satisfying

$$\partial_x > \partial_y, \quad \phi > \psi > a > b > h$$

outputs in 3.1 sec. two differential systems S_1 and S_2 (see (28) and (30)). Each system has only one equation containing $h(x,y)$:

$$h(x,y) - 8x^2 = 0. \quad (27)$$

The linearizability of (26) under condition (27) was established in paper of Ibragimov (2008), and our computation shows that there are no other linearizable equations of family (26). Moreover, the simple systems S_1 and S_2 allow for the explicit construction of the linearizing point transformation (4) and the coefficients $a_0(t)$ and $a_1(t)$ in the Laguerre-Forsyth form (7) of the image of (26) under mapping (4):

$$u^{(4)}(t) + a(t)u(t) + b(t)u'(t) = 0.$$

To show this, consider first the equations in S_1^- :

$$\begin{aligned} y\phi_y - 2\phi - 2\phi_{xxx} &= 0, \quad \phi_{xxy} = 0, \quad x^2\phi_{xy} - 2\phi_y = 0, \quad x\phi_{xy} - 2\phi_y = 0, \\ y\phi_{yy} - \phi_y &= 0, \quad a\psi_x^4 - 1 = 0, \quad \psi_y = 0, \quad a_x = 0, \quad a_y = 0, \quad b = 0, \quad h - 8x^2 = 0, \end{aligned} \quad (28)$$

and its inequations

$$S_1^\# = \{a \neq 0, \phi_y \neq 0\}. \quad (29)$$

The equation system (28) can easily be integrated by hand or using the Maple routine `pdsolve`. The general solution to (28), in addition to (27), reads

$$\begin{aligned} \phi_1 &:= c_1 x^2 y^2 \\ &+ \sin\left(\frac{x}{\sqrt{2}}\right) \left(c_2 \exp\left(-\frac{x}{\sqrt{2}}\right) + c_3 \exp\left(\frac{x}{\sqrt{2}}\right) \right) + \cos\left(\frac{x}{\sqrt{2}}\right) \left(c_4 \exp\left(-\frac{x}{\sqrt{2}}\right) + c_5 \exp\left(\frac{x}{\sqrt{2}}\right) \right), \\ \psi_1 &:= \frac{x}{c_6^{1/4}} + c_7, \quad a_1 := c_6, \quad b_1 := 0, \end{aligned}$$

where c_k ($k \in \{1, \dots, 7\}$) are arbitrary constants and the subscript 1 represents the obtained solution to the differential system S_1 . Ineq. (29) implies $c_1 \neq 0$ and $c_6 \neq 0$.

The second differential system S_2 is generic, and its set of equations is given by

$$\begin{aligned} 32ya\phi_y - 64a^3(\phi + \phi_{xxx}) - 96a^2a_x\phi_{xx} - 36a_x^2\phi_{xx} - 3a_x^3\phi_x &= 0, \\ 128x^2a^3\phi_{xxy} + 15x^2a_x^3\psi_y - 144xa_x^2\phi_y + 288a^2a_x\phi_y &= 0, \\ 16x^2a^2\phi_{xy} - 3x^2a_x^2\phi_y + 24xa_x\phi_y - 32a^2\phi_y &= 0, \\ xa\phi_{xy} + 3xa_x\phi_y - 16a\phi_y = 0, y\phi_{yy} - \phi_y = 0, a\psi_x^4 - 1 = 0, \psi_y = 0, \\ 8a_x a_{xx} - 7a_x^2 = 0, a_y = 0, b = 0, h - 8x^2 = 0. \end{aligned} \quad (30)$$

The set of inequations in S_2 consists of three elements:

$$S_2^\neq = \{a \neq 0, \phi_y \neq 0, a_x \neq 0\}. \quad (31)$$

The system (30) of differential equations is also easily solvable. Its general solution reads

$$\begin{aligned} \phi_2 &:= \frac{\phi_1}{(c_6x+c_7)^3}, & \psi_2 &:= c_8 - \frac{1}{c_6(c_6x+c_7)}, \\ a_2 &:= (c_6x+c_7)^8, & b_2 &:= 0, & h &:= 8x^2, \end{aligned} \quad (32)$$

where ϕ_1 is the above presented solution to (28) with arbitrary constants c_i ($1 \leq i \leq 8$).

The constraints that follow from (31) are those in S_1 , $c_1 \neq 0, c_6 \neq 0$, and the additional inequation $c_6x + c_7 \neq 0$ rules out singularity in (32).

The obtained explicit solutions to S_1 and S_2 form disjointed sets, since $a_x = 0$ for a solution to S_1 and $a_x \neq 0$ for that to S_2 . The disjointness of solution sets for the output simple systems is guaranteed by the Thomas decomposition algorithm (Bachler (2012); Gerdt (2008); Li (1999); Robertz (2014); Thomas (1937, 1962)). In the given case a solution to S_1 provides a mapping of (26) into the linear ODE

$$u^{(4)}(t) + c_6u(t) = 0$$

with constant coefficients, whereas a solution to S_2 maps (26) into an equation with variable coefficients

$$u^{(4)}(t) + (c_6x + c_7)^8u(t) = 0.$$

In (Ibragimov (2008)), the simplest form of the linearizing transformation (4) was found:

$$t = x, \quad u = x^2y^2,$$

which maps (26) and (27) into $u^{(4)} + u = 0$ and corresponds to the solution of S_1 with $c_1 = c_6 = 1$ and $c_2 = c_3 = c_4 = c_5 = c_7 = 0$.

Example 12. (Tremblay-Turbiner-Winternitz system) These equations come from the theory of superintegrable Hamiltonian systems, which recently have found the application in quantum mechanics as new exactly solvable models. As it is shown in (Gubbiotti (2017)), the Tremblay-Turbiner-Winternitz system can be described by two third-order equations (Gubbiotti (2017), Eq. (89),(90)), in which the first one is given by

$$y(x)y'''(x) + y'(x)(16\omega^2y(x) + 3y''(x)) = 0, \quad \omega \in \mathbb{R}_{>0}, \quad (33)$$

and can be reduced to

$$u'''(t) = -16\omega^2u'(t) \quad (34)$$

using the transformation

$$u = \frac{y^2}{2}, \quad t = x. \quad (35)$$

This reduction yields the general solution of (33) respectively (Gubbiotti (2017), Eq. (89))

$$y(x) = \sqrt{a_1 + a_2 \cos(4\omega x) + a_3 \sin(4\omega x)}, \quad (36)$$

where a_1, a_2, a_3 are arbitrary constants. Eq. (36) can be substituted into (Gubbiotti (2017), Eq. (90)) which is also linearizable and can be solved similarly. In this regard, we omit the

details here for brevity and refer to Gubbiotti (2017). In the following, we are sketching the procedure how to treat Eq. (33) using our approach.

If one applies the algorithm **LinearizationTest I** to Eq. (33) when $\omega = 1$, then in about 0.2 sec. it outputs `True`. It should be noted that Eq. (34) is not in the Laguerre-Forsyth form (7). Thus, the algorithm **LinearizationTest II** constructs a point transformation (4) distinct from (35) and, respectively, another linear form

$$u'''(t) + a_0(t)u(t) = 0 \quad (37)$$

of (33) by solving the simple differential system computed in line 18 of the algorithm.

The output simple system is computed in about 0.4 sec. Its equations and inequations read

$$16\omega^2\phi_y^2\phi_x - 2\phi_y^2\phi_{xxx} + 6\phi_y\phi_{xy}\phi_{xx} - 3\phi_x\phi_{xy}^2 = 0, \quad (38a)$$

$$16\omega^2\phi_y^2 - 2\phi_y\phi_{xxy} + 3\phi_{xy}^2 = 0, \quad \phi_y\psi_{xx} - \phi_{xy}\psi_y = 0, \quad (38b)$$

$$y\phi_{yy} - \phi_y = 0, \quad \phi_y \neq 0, \quad (38c)$$

$$\psi_y = 0, \quad \psi_x \neq 0, \quad a_0 = 0. \quad (38d)$$

The general solution to (38c) and (38d) is given by

$$\phi(x, y) := F_1(x) + F_2(x)y^2, \quad \psi(x, y) := F_3(x) \quad (39)$$

where $F_i(x)$ ($i = 1, 2, 3$) are arbitrary functions such that $F_2(x)$ and $F_3(x)$ do not vanish identically. The substitution of Eqs. (39) with $F_1(x) = 0$ into Eqs. (38a)–(38b) reduces them to ordinary differential equations whose general solution is readily found with the built-in Maple solver `dsolve` and gives the following structure of (39)

$$\phi(x, y) := \frac{16\omega^2 y^2}{[c_1 \sin(2\omega x) - c_2 \cos(2\omega x)]^2}, \quad \psi(x, y) := c_3 + \frac{c_4}{c_1 \tan(2\omega x) + c_2}. \quad (40)$$

with arbitrary constants c_1, c_2, c_3, c_4 . The setting $c_1 := 4, c_2 := 0, c_3 := 0, c_4 := 4$ gives the following linearizing transformation determined by a particular solution to the system (38)

$$u := \frac{\omega^2 y^2}{\sin^2(2\omega x)}, \quad t := \frac{1}{\tan(2\omega x)}. \quad (41)$$

Since $a_0 = 0$ in (38d), Eq. (37) reads $u'''(t) = 0$, and its general solution is given by

$$u(t) = A_2 t^2 + A_1 t + A_0, \quad A_0, A_1, A_2 \text{ are constants.}$$

By Eqs. (41), in terms of the initial variables this solution is given by

$$y(x) = \frac{1}{\omega} \sqrt{A_2 \cos^2(2\omega x) + A_1 \cos(2\omega x) \sin(2\omega x) + A_0 \sin^2(2\omega x)}$$

and coincides with (36) if one sets

$$a_1 := \frac{A_0 + A_1}{2\omega^2}, \quad a_2 := \frac{A_1}{2\omega^2}, \quad a_3 := \frac{A_0 - A_1}{2\omega^2}.$$

Example 13. As a serial example for benchmarking, we consider

$$(y^2)^{(n)} + y^2 = 0, \quad n \in \mathbb{N}_{\geq 3}. \quad (42)$$

Obviously, Eq. (42) becomes $u^{(n)} + u = 0$ via transformation (4) of the form $t = x$ and $u = y^2$. We use this example as a benchmark for a comparative experimental analysis of the time behavior of our algorithms when the order of the ODE grows. In so doing, we measure the CPU time for the three different implementations of algorithm **LinearizationTest I** (see Sect. 4).

Table 1 presents the CPU times, and we use the subscript in the first column to mark implementation of the first algorithm described in the corresponding enumerated items of Sect. 4. “OOM” is an acronym for runs “Out Of Memory”. As one can see, our first linearization test (I) is substantially faster than the second test (II), and the implementation marked with subscript 1 is the fastest. However, algorithm **LinearizationTest II**, being the slowest, outputs much more information on the linearization. This fact was illustrated by Examples 9, 11 and 12.

Table 1: CPU times (sec.)

Test	Order n of the ODE (42)										
	3	4	5	6	7	8	9	10	11	12	13
I ₁	0.3	0.4	0.4	0.5	0.6	0.8	1.1	1.4	1.8	2.3	3.0
I ₂	0.0	0.5	1.1	2.0	3.9	5.8	9.5	20.2	35.7	50.3	71.2
I ₃	0.2	0.7	1.3	2.7	4.9	8.9	14.3	39.4	70.7	120.2	234.3
II	0.5	1.8	9.4	30.1	209.9	789.8	2011	8200	25217	OOM	OOM

Example 14. As another serial example for benchmarking, we consider three ODEs of fifth order:

$$(I) : (Y)^{(5)} + Y = 0, \quad (II) : (Y)^{(5)} + xY = 0, \quad (III) : (Y)^{(5)} + Y^2 = 0, \quad (43)$$

where $Y = P_n(y)$ is an arbitrary dense polynomial of n -th order, generated by the Maple command `randpoly`. Table 2 presents the CPU times for the algorithm **LinearizationTest I**.

Table 2: CPU times (sec.)

Test	Order n of the polynomial $P_n(y)$ in (43)										
	1	10	20	30	40	50	60	70	80	90	100
I	0.4	8.0	22.1	37.3	63.4	89.6	119.5	162.1	217.7	261.6	317.8
II	0.5	7.8	21.6	36.3	64.5	90.4	124.2	163.1	215.8	262.8	325.8
III	0.14	7.9	23.5	37.1	67.1	94.4	126.9	170.3	231.6	274.2	351.1

Example 15. (Ibragimov (2005)) The general problem of third-order linearizable equations is a bit more complicated. It contains two differential algebraic forms of candidates for linearization. The first form corresponds to the particular case $g_y = 0$. It is described by

$$y''' = (A_1 y' + A_0) y'' + B_3 (y')^3 + B_2 (y')^2 + B_1 y' + B_0.$$

We apply the algorithm **LinearizationTest II** with block elimination ranking such that

$$\partial_x > \partial_y, \quad \phi > \psi \gg A_1 > A_0 > B_3 > B_2 > B_1 > B_0.$$

Then the routine DifferentialThomasDecomposition of the package DifferentialThomas (2012) outputs three differential systems with disjoint solutions space in about 4.3 sec.

The first system is generic one and contains compatibility conditions which coincides exactly with paper of Ibragimov (2005):

$$\begin{aligned} \frac{\partial A_0}{\partial y} - \frac{\partial A_1}{\partial x} &= 0, \quad \frac{\partial}{\partial y} \left(3B_1 - A_0^2 - 3 \frac{\partial A_0}{\partial x} \right) = 0, \\ 3 \frac{\partial A_1}{\partial x} + A_0 A_1 - 3B_2 &= 0, \quad 3 \frac{\partial A_1}{\partial y} + A_1^2 - 9B_3 = 0, \\ \left(9B_1 - 6 \frac{\partial A_0}{\partial x} - 2A_0^2 \right) \frac{\partial A_1}{\partial x} + 9 \frac{\partial}{\partial y} \left(\frac{\partial B_1}{\partial x} - A_1 B_0 \right) + 3 \frac{\partial B_1}{\partial y} A_0 - 27 \frac{\partial^2 B_0}{\partial y^2} &= 0. \end{aligned}$$

For the general case

$$y''' = \frac{-3(y'')^2 + (C_2(y')^2 + C_1 y' + C_0)y'' + D_5(y')^5 + D_4(y')^4 + D_3(y')^3 + D_2(y')^2 + D_1 y' + D_0}{y' + r},$$

the differential Thomas decomposition with block elimination ranking

$$\partial_x > \partial_y, \quad \phi > \psi \gg C_2 > C_1 > C_0 > D_5 > D_4 > D_3 > D_2 > D_1 > D_0.$$

yields four differential system⁸

$$\begin{aligned} S_1 &= \tilde{S}_1 \cup \{M = 0, r \neq 0\}, \quad S_2 = \tilde{S}_2 \cup \{M \neq 0, r = 0\}, \\ S_3 &= \tilde{S}_3 \cup \{M \neq 0, r \neq 0\}, \quad S_4 = \tilde{S}_4 \cup \{M = 0, r = 0\}, \end{aligned}$$

where

$$\begin{aligned} M &= 36D_3 \frac{\partial r}{\partial x} + 3C_0 D_3 + 36 \frac{\partial C_2}{\partial x} \frac{\partial r}{\partial x} + 3C_0 \frac{\partial C_2}{\partial x} - 18 \frac{\partial D_5}{\partial y} r^4 - 72 \frac{\partial D_5}{\partial x} r^3 \\ &+ 18 \frac{\partial D_4}{\partial y} r^3 - 5C_2^3 r^3 + 72 \frac{\partial D_4}{\partial x} r^2 - 18 \frac{\partial D_3}{\partial y} r^2 - 18 \frac{\partial D_3}{\partial x} r - 18 \frac{\partial^2 C_2}{\partial x^2} r \\ &+ 12C_1 C_2 \frac{\partial r}{\partial x} + C_0 C_1 C_2 - 5C_0 C_2^2 r + 63C_2 D_5 r^4 - 39C_1 D_5 r^3 - 48C_2 D_4 r^3 \\ &- 90D_5 r^2 \frac{\partial r}{\partial x} + 15C_0 D_5 r^2 + 10C_1 C_2^2 r^2 + 15C_2 \frac{\partial C_2}{\partial x} r^2 + 24C_1 D_4 r^2 \\ &+ 33C_2 D_3 r^2 - 12 \frac{\partial C_1}{\partial x} C_2 r - 5C_1^2 C_2 r - 18C_2 D_2 r - 9C_1 D_3 r - 15C_1 \frac{\partial C_2}{\partial x} r. \end{aligned}$$

The above output of the differential Thomas decomposition differs from the system (2.33)–(2.40) given in (Ibragimov (2005), Thm.2.2). It contains eight equations and no inequations in the differential system. Let denote it by T . To prove that we obtained an equal result, we split the system T into four disjoint systems

$$\begin{aligned} T_1 &= T \cup \{M = 0, r \neq 0\}, \quad T_2 = T \cup \{M \neq 0, r = 0\}, \\ T_3 &= T \cup \{M \neq 0, r \neq 0\}, \quad T_4 = T \cup \{M = 0, r = 0\}, \end{aligned}$$

Application of the Thomas Decomposition to each system T_i yields the involutive differential system which coincides exactly with S_i .

⁸We do not explicitly write out the resulting equations here for brevity.

6. Conclusions

For the first time, the problem of the linearization test was algorithmically solved for a wide class of ordinary differential equation of arbitrary order. In doing so, we have restricted ourselves to the quasi-linear equations with a rational dependence on the other variables and to point transformations, and designed two algorithmic tests in order to check linearizability.

The main benefits of these restrictions are (i) the algorithmic construction of the Lie symmetry algebra for the input equation and (ii) the reduction of the number of coefficients in the linearized equation due to the Laguerre-Forsyth canonical form (7). The benefit (i) allowed us to design an efficient algorithm **LinearizationTest I** which checks the linearizability of the equations. The second benefit (ii) provides the feasibility of the algorithm **LinearizationTest II** because of the overdeterminacy (cf. Seiler (2010), Sect. 7.5) of a linearizing differential system. This overdeterminacy simplifies the consistency analysis of the linearizing system answering the same question as the first test.

Moreover, due to finite-dimensionality of the solution space (cf. Olver (1995), Prop. 6.57) of a linearizing system, the Thomas decomposition algorithm outputs overdetermined subsystems, as those in Examples 11 and 12. In practice, the overdeterminacy of the outputted simple systems of the Thomas decomposition of a linearizing differential system makes them easily solvable, much like the determining systems in the Lie symmetry analysis. Thereby, with the algorithm **LinearizationTest II** one can not only detect linearizability, but also find the linearizing transformation (4) and the coefficients in the linear form of the equation.

The Thomas decomposition for linearizing differential systems, even in the case of its inconsistency, may be time and space consuming, especially for higher-order ODEs. That is why, in practice, it is advisable to check the linearizability of the equation under consideration by the first algorithm before applying the second one. In the case when Eq. (3) contains parameters and/or arbitrary functions, there is no choice and one has to use the second algorithm. The first algorithm could also improve work of the second one. Depending on the Lie Algebra of the differential equations under consideration, we can immediately conclude which type of linear equation (trivial, constant coefficients, or the general case) it could be mapped into. It allows to add constraints on coefficients a_k directly for linearizing differential systems in a way to increase computational efficiency of the Thomas decomposition.

Algorithm **LinearizationTest II** may also improve the built-in Maple solver *dsolve* of differential equations. For example, *dsolve* applied to the equation

$$y''' + \frac{3y'}{y}(y'' - y') - 3y'' + 2y' - y = 0 \quad (44)$$

outputs its solution implicitly in the complicated form of double integrals including Maple's symbolic presentation *RootOf* for the roots of expressions. On the other hand, Eq. (44) admits the linearization (cf. Ibragimov (2009), Eqs. (6.6.57)–(6.6.59))

$$u''' - \frac{2}{t^3}u = 0, \quad t = \exp(x), \quad u = y^2,$$

which is easily obtained by our algorithm **LinearizationTest II** and provides the explicit form of the solution to (44).

Similarly, if one applies *dsolve* to Eq. 33, then it outputs definite integrals with $y(x)$ in the upper limit and with integrands containing *RootOf* of cumbersome combinations of algebraic

and transcendental expressions including three arbitrary constants. It is hardly possible to obtain Eq. 36 from such an output.

Our algorithms admit a generalization to contact and higher symmetries for scalar ordinary differential equations as these cases are also well studied in symmetry analysis. It would be interesting to have a similar theory for second-order ordinary differential systems, which are of high interest in natural sciences and engineering. The main problem here is to obtain an analogue of Thm. 3 and Cor. 4. Some particular results were recently obtained for the case of two second-order ordinary differential equations (Ayub (2012); Bagderina (2010); Soh (2001); Sookmee (2011)), but algorithmic and computational aspects of this problem are still open questions.

7. Acknowledgments

The authors are grateful to Daniel Robertz and Greg Reid for helpful discussions and suggestions. This work has been partially supported by KAUST (D. A. Lyakhov and D. L. Michels), by the RUDN University Program (5-100; V. P. Gerdt), and by Stanford University (D. L. Michels).

References

- Vladimir Arnold, *Ordinary Differential Equations*, Springer, Berlin, 1992.
- Muhammad Ayub, Masood Khan, and Fazal Mahomed, Algebraic linearization criteria for systems of ordinary differential equations, *Nonlinear Dynamics* 67, no. 3 (2012): 2053–2062.
- Thomas Bächler, Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz, Algorithmic Thomas decomposition of algebraic and differential systems, *Journal of Symbolic Computation* 47, no. 10 (2012): 1233–1266.
- Yulia Bagderina, Linearization Criteria for a System of Two Second-Order Ordinary Differential Equations, *Journal of Physics A: Mathematical and Theoretical* 43, no. 46 (2010): 465201.
- George Bluman and Stephen Anco, *Symmetry and Integration Methods for Differential Equations*, Springer, New York, 2001.
- John Carminati and Khai Vu, Symbolic Computation and Differential Equations: Lie Symmetries, *Journal of Symbolic Computation* 29, no. 1 (2000): 95–116.
- Manuel Ceballos, Juan Núñez, and Angel Tenorio, Algorithmic method to obtain abelian subalgebras and ideals in Lie algebras, *International Journal of Computer Mathematics* 89, no. 10 (2012): 1388–1411.
- Norbert Euler, Thomas Wolf, Peter Leach, and Marianna Euler, Linearizable Third-Order Ordinary Differential Equations and The Generalised Sundman Transformations: The Case $X''' = 0$, *Acta Applicandae Mathematicae* 76, no. 1 (2003): 89–115.
- Tarcisio Filho and Annibal Figueiredo, [SADE] a Maple package for the symmetry analysis of differential equations, *Computer Physics Communications* 182, no. 2 (2011): 467–476.
- Vladimir Gerdt, On decomposition of algebraic PDE systems into simple subsystems, *Acta Applicandae Mathematicae* 101, no. 1 (2008): 39–51.
- Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz, The MAPLE package TDDS for computing Thomas decompositions of systems of nonlinear PDEs, *Computer Physics Communication* 234, (2019): 202–215.
- Dima Grigoriev and Alexander Chistov, Complexity of a standard basis of a D-module, *St. Petersburg Mathematical Journal* 20, no. 5 (2009): 709–736.
- Giorgio Gubbiotti and Maria Clara Nucci, Are all classical superintegrable systems in two-dimensional space linearizable?, *Journal of Mathematical Physics* 58 (2017): 012902.
- Richard Gustavson, Alexey Ovchinnikov, and Gleb Pogudin, New order bounds in differential elimination algorithms, *Journal of Symbolic Computation* 85 (2018): 128–147.
- Evelyne Hubert, Notes on triangular sets and triangulation-decomposition algorithms. II Differential systems, In: Franz Winkler, and Ulrich Langer (Editors), *Symbolic and Numerical Scientific Computation*, Hagenberg, 2001, Lecture Notes in Computer Science, Vol. 2630, Springer, Berlin, 2003, pp. 40–87.
- Nail Ibragimov and Sergey Meleshko, Linearization of third-order ordinary differential equations by point and contact transformations, *Journal of Mathematical Analysis and Applications* 308, no. 1 (2005): 266–289.
- Nail Ibragimov, Sergey Meleshko, and Supaporn Suksern, Linearization of fourth-order ordinary differential equations by point transformations, *Journal of Physics A: Mathematical and Theoretical* 41, no. 23 (2008): 235206.

- Nail Ibragimov, *A Practical Course in Differential Equations and Mathematical Modelling. Classical and New Methods. Nonlinear Mathematical Models. Symmetry and Invariance Principles*, Higher Education Press, World Scientific, New Jersey, 2009.
- Markus Lange-Hegermann, The Differential Dimension Polynomial for Characterizable Differential Ideals, arXiv:math-AC/1401.5959.
- Ian Lisle and Tracy Huang, Algorithmic calculus for Lie determining systems, *Journal of Symbolic Computation* 79 (2017): 482–498.
- Markus Lange-Hegermann, *DifferentialThomas: Thomas decomposition of differential systems*, <http://wwwb.math.rwth-aachen.de/thomasdecomposition/>.
- Ziming Li and Dongming Wang, Coherent, Regular and Simple Systems in Zero Decompositions of Partial Differential Systems, *Systems Science and Mathematical Sciences* 12, no. 5 (1999): 43–60.
- Sophus Lie, Klassifikation und Integration von gewöhnlichen Differentialgleichungen zwischen x, y , die eine Gruppe von Transformationen gestatten, III, *Archiv for Matematik og Naturvidenskab*, 8(4), 1883, 371–458. Reprinted in Lie's *Gesammelte Abhandlungen*, 5, paper XIY, 1924, 362–427.
- Sophus Lie, *Vorlesungen über kontinuierliche Gruppen mit geometrischen und anderen Anwendungen*. Bearbeitet und herausgegeben von Dr. G. Schefferes. Teubner, Leipzig, 1883.
- Dmitry A. Lyakhov, Vladimir P. Gerdt and Dominik L. Michels, Algorithmic Verification of Linearizability for Ordinary Differential Equations. Proceedings of the 42nd International Symposium on Symbolic and Algebraic Computation (ISSAC'17), Michael Burr (Ed.), ACM New York (2017): 285–292.
- Fazal Mahomed and Peter Leach, Symmetry Lie Algebra of n th Order Ordinary Differential Equations. *Journal of Mathematical Analysis and Applications* 151, no. 1 (1990): 80–107.
- Peter Olver, *Applications of Lie Groups to Differential Equations*, 2nd Edition. Graduate Texts in Mathematics, Vol. 107, Springer, New York, 1993.
- Peter Olver, *Equivalence, Invariance and Symmetry*. Cambridge University Press, 1995.
- Lev Ovsyannikov, *Group Analysis of Differential Equations*, Academic Press, New York, 1992.
- Greg Reid, Algorithms for reducing a system of PDEs to standard form, determining the dimension of its solution space and calculating its Taylor series solution, *European Journal of Applied Mathematics* 2, no. 4 (1991): 293–318.
- Greg Reid, Finding abstract Lie symmetry algebras of differential equations without integrating determining equations, *European Journal of Applied Mathematics* 2, no. 04 (1991): 319–340.
- Daniel Robertz, *Formal Algorithmic Elimination for PDEs*, Lecture Notes in Mathematics, Vol. 2121, Springer, 2014.
- Fritz Schwarz, *Algorithmic Lie Theory for solving ordinary Differential Equations*, Chapman & Hall/CRS, Boca Raton, 2008.
- Werner Seiler, *Involution: The Formal Theory of Differential Equations and its Applications in Computer Algebra*, Algorithms and Computation in Mathematics 24, Springer, Heidelberg, 2010.
- Celestin Soh and Fazal Mahomed, Canonical forms for systems of two second-order ordinary differential equations, *Journal of Physics A: Mathematical and General* 34 (2001): 2883–2991.
- Sakka Sookmee and Sergey Meleshko, Linearization of Two Second-Order Ordinary Differential Equations via Fiber Preserving Point Transformations, *ISRN Mathematical Analysis*, Article ID: 452689, 2011.
- Joseph Thomas, *Differential Systems*, AMS Colloquium Publications, Vol. XXI, AMS, New York, 1937.
- Joseph Thomas, *Systems and Roots*, William Byrd Press, Richmond, 1962.
- Khair Vu, Grace Jefersson, and John Carminati, Finding higher symmetries of differential equations using the MAPLE package DESOLVII, *Computer Physics Communications* 183, no. 4 (2012): 1044–1054.
- Dongming Wang, *Elimination Methods*, Springer, Wien, 2001.