Algorithmic methods for ordinary differential equations

Dissertation

zur Erlangung des Doktorgrades
der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

vorgelegt von

Kai Frederik Gehrs

Paderborn, den 31. Oktober 2006

Gutachter der Dissertation:

Prof. Dr. Benno Fuchssteiner (Betreuer der Dissertation, Universität Paderborn)

Prof. Dr. Athanassios S. Fokas (University of Cambridge)
Dr. habil. Cornelia Schiebold (Mid Sveden University)
Prof. Dr. Walter Strampp (Universität Kassel)

Tag der mündlichen Prüfung:

26.01.2007

"Sie sagen zu mir
schließ auf diese Tür
die Neugier wird zum Schrei
was wohl dahinter sei."

— T. Lindemann

Abstract

In this thesis, new algorithmic methods for ordinary differential equations (ODEs) are developed as well as the foundation of the algebraic basis for these methods. The aim of this thesis is of a rather practical nature, i.e. the presentation of practical and efficient algorithms and heuristics for the solution of ODEs, which can directly be implemented in the framework of a general purpose computer algebra system. The emphasis is clearly put on "practical and efficient". However, in order to achieve this goal, a rather extensive algebraic setup is developed and new and customized notions are designed. Extensions of the methods developed by E. S. Cheb-Terrab et. al. in the 1990s are presented. These extensions can be used to compute integrating factors of third and higher order non–linear ODEs. Furthermore, a new approach for the computation of integrating factors of ODEs arising from the application of special skew symmetric operators is developed. A more general approach and yet unpublished symmetry result by B. Fuchssteiner is presented. New applications of this result are discussed and it is shown, how the symmetry results by Cheb-Terrab et. al. can be used in this more general theoretical setting. Finally, by introduction of a new type of symmetries (non–local symmetries) a link between two theories (differential Galois theory and symmetry analysis), which were up to now considered to be totally disjoint, is established. Technically seen, this amounts to a combination of symmetry methods and the theory of nilpotent flows in order to give a new algorithmic approach for computing the important symmetric powers of linear differential operators.

Zusammenfassung

In dieser Arbeit werden neue algorithmische Methoden zur Lösung gewöhnlicher Differentialgleichungen (kurz ODEs) vorgestellt. Ziel der Arbeit ist die Darstellung von praktischen und effizienten Algorithmen und Heuristiken für ODEs sowie die dafür nötigen algebraischen Grundlagen, die sich auch praktisch in einem Computeralgebra System implementieren lassen. Die Betonung liegt auf "praktisch und effizient". Zur Realisierung dieser Verfahren werden neue algebraische Strukturen definiert und die für die Algorithmen notwendigen theoretischen Grundlagen entwickelt. Es werden Verallgemeinerungen der von E. S. Cheb-Terrab et. al. in den 1990er Jahren entwickelten Verfahren für ODEs vorgestellt. Diese können zur Berechnung von integrierenden Faktoren für nicht–lineare ODEs dritter und höherer Ordnung verwendet werden. Ferner werden neue effiziente Verfahren zur Berechnung von integrierenden Faktoren vorgestellt, die durch Anwendung schiefsymmetrischer Operatoren entstehen. Anwendungen eines neuen und bisher nicht publizierten Integrabilitätsresultats von B. Fuchssteiner werden diskutiert und es wird gezeigt, wie sich die von Cheb-Terrab entwickelten Symmetriemethoden in den allgemeineren Kontext dieses neuen Resultats einbetten lassen. Schließlich wird durch Einführung eines neuen Symmetrietyps (nicht–lokale Symmetrien) eine Verbindung zwischen zwei bis dato völlig disjunkten Theorien (Differentialgaloistheorie und Symmetrieanalyse) hergestellt. Technisch gesehen wird durch die Kombination von Symmetriemethoden und der Theorie nilpotenter Flüsse ein neuer Zugang zur Berechnung der wichtigen symmetrischen Potenzen von linearen Differentialoperatoren entwickelt.

Acknowledgement

I would like to thank

- Prof. Dr. Benno Fuchssteiner for so many things among which there are: introducing me to the MuPAD family, his unbelievable patience, support and encouragement, introducing me to the topic of differential equations and for much much more
- Prof. Dr. Walter Oevel for many many interesting and enlightening discussions
- The department of mathematics at the university of Paderborn especially including the Rechnerbetreuung Mathematik
- ICMS for their support and the chance to discuss some of the results of this thesis with the participants (especially Prof. Dr. Malcolm MacCallum, Prof. Dr. Alexandre V. Mikhailov and Prof. Dr. Sergey Tsarev) at the conference "Algebraic Theory of Differential Equations" at Heriot–Watt University, Edinburgh, 2006
- Computeralgebra–Fachgruppe Deutschland for the invitation to the conference of the Computeralgebra–Fachgruppe Deutschland, Kassel, 2005
- Dr. habil. Cornelia Schiebold and Prof. Dr. Bernd Carl for their invitation to a talk about some of the ideas of this thesis at the Friedrich–Schiller–Universität in Jena, 2005
- My family Dodo, Bodo, Michael, Julia and Jenni for their support
- Dr. Bernd Engel, Mike Menne, Lars Tegeler
- Jürgen Billing, Alessandro Dell'Aere, René S. Kaup, Jörn Maas and Tatjana Schmücker for constructive criticism and their friendship
- Christopher Creutzig, Klaus Drescher, Ralf Hillebrand, Dr. Oliver Kluge, Martin Knelleken, Torsten Metzner, Dr. Andreas Sorgatz, Dr. Stefan Wehmeier and the rest of the MuPAD family for their support
- Doris and Antonius Hillebrand for more than their tolerance and the lovely roses in front of my living room
- Studienstiftung des Deutschen Volkes, especially Dr. Susanne Happ, Dr. Petra Neumann and Prof. Dr. Dr. Dr. Hubertus R. Drobner
- Last but not least: Richard Z. Kruspe, Paul Landers, Till Lindemann, Christian Lorenz, Oliver Riedel and Christoph Schneider

Contents

Introduction

In this thesis, algorithmic methods for ordinary differential equations are developed. The aim of this thesis is of a rather practical nature, i.e. the presentation of practical and efficient algorithms and heuristics for the solution of ordinary differential equations, which can directly be implemented in the framework of a general purpose computer algebra system. The emphasis is clearly put on "practical and efficient" algorithms and heuristics. However, in order to achieve this goal, a rather extensive algebraic setup is developed and new and customized notions are designed.

Two examples

To make clear, what is meant by "practical and efficient", let us roughly sketch two examples, which will be discussed in detail in the later course of this thesis: consider a third order non–linear ordinary differential equation

$$y'''(x) = \frac{y'(x)}{y(x)}\, y''(x) - y(x)^2\, y'(x)$$

and assume that an integrating factor for this equation should be found (see also [45], p. 602, ODE 7.7 for this example). The general theory proposes to compute an integrating factor by means of solving a system of linear partial differential equations (see [11]). But an integrating factor for that equation can easily be computed from the right–hand–side of the equation. Multiplying the equation by $\frac{1}{y(x)}$ provides an exact equation, i.e. the whole equation can be integrated and, hence, the order of the equation can be reduced by 1. This integrating factor can be found without solving any auxiliary ordinary or partial differential equations using only very elementary techniques. It can be deduced simply by a careful inspection of the structure of the right–hand–side of the equation (see also Example 2.14 on page 90 for details on how this integrating factor can be found).

In the framework of a computer algebra system, where in general an elaborate symbolic solver for partial differential equations is not available, an approach for the computation of integrating factors not having to compute the solutions of further auxiliary ordinary or partial differential equations is desirable. Furthermore, even if methods or special heurstics for solving partial differential equations for the computation of integrating factors are available (as e.g. in the case of the computer algebra system MAPLE), approaches in the above sense should be preferred, since in practice they work more efficiently (see also [11]).

As a second example, consider a seventh order ordinary differential equation

$$\begin{aligned}0 = {} & 35\, u^7\, u_x + 35\, u_{xxx}\, u^5 + 385\, u^4\, u_x\, u_{xx} + 420\, u^3\, u_x^3 + 14\, u_{xxxxx}\, u^3 + \\ & 154\, u_{xxxx}\, u^2\, u_x + 280\, u_{xxx}\, u^2\, u_{xx} + 476\, u_{xxx}\, u\, u_x^2 + 672\, u\, u_x\, u_{xx}^2 + \\ & 2\, u_{xxxxxxx}\, u + 420\, u_x^3\, u_{xx} + 14\, u_{xxxxxx}\, u_x + 42\, u_{xxxxx}\, u_{xx} + 70\, u_{xxx}\, u_{xxxx},\end{aligned}$$

where $u = u(x)$, $u_x = u'(x)$, $u_{xx} = u''(x)$ etc. This equation admits the integrating factor

$$G = \frac{5u^6}{2} + 10u^3 u_{xx} + 20u^2 u_x^2 + 2u_{xxxx}u + 8u_{xxx}u_x + 6x_{xx}^2.$$

Multiplication of the equation by G provides an exact equation and the order of the differential equation can be reduced via integration. Again, as in the former example, the integrating factor can be determined without solving any auxiliary ordinary or partial differential equations.

In contrast to the above example, where the stated integrating factor is found mainly by using pattern matching methods applied to the right–hand–side of the equation, the integrating factor G is obtained by new methods for so–called skew symmetric hierarchies of ordinary differential equations introduced in Chapter 3 of this thesis. The computation of the reduction by means of the integrating factor G is discussed in details in Example 3.43 on page 172. Furthermore, Chapter 3 provides the necessary algebraic setup, which allows to compute this reduction without performing any direct integration.

Although the methods for the computation of integrating factors in the two examples above are completely different, there is one common property: both methods provide an integrating factor and in the intermediate steps of computation no auxiliary ordinary or partial differential equation has to be solved. By “practical and efficient algorithms” we mean such methods.

In the 1990s, E. S. Cheb-Terrab et. al. developed computer algebra methods for the computation of symmetries and integrating factors of mainly first and second order ordinary differential equations, which — among other theoretical aspects — are based on pattern matching analysis of the ordinary differential equations under consideration (see [7], [9], [10], [11], [12], [13]).

These methods provided amazing results in practice: e.g. large classes of the ordinary differential equations listed in the well–known book by E. Kamke (see [45]) could be tackled using these new methods. And for the computation of symmetries and integrating factors of these classes of equations, the solution of any auxiliary ordinary or partial differential equations could mainly be avoided. The method used to compute the integrating factor in the first example stated above is one of the algorithms presented in the framework of this thesis, which are inspired by the ideas of Cheb-Terrab et. al.

This thesis

In this thesis, among other items, extensions of the methods of Cheb-Terrab et. al. are presented. These extensions can be used to compute integrating factors of third and higher order non–linear ordinary differential equations. Furthermore, a new approach for the computation of integrating factors of ordinary differential equations arising from the application of special skew symmetric operators is discussed. By means of these methods, the integrating factor in the second example stated above is obtained.

However, a main emphasis of this work lies on the symmetry analysis of ODEs, where new approaches and notions are presented. Incidentally, for first order ordinary differential equations the concepts of symmetry analysis and integrating factors are identical (see also Section 2.8.1 for further remarks on this statement). Hence, for those more general classes of ordinary differential equations, where these two approaches differ, new unifying viewpoints have to be designed. In particular, a more general approach[1] and yet unpublished symmetry result established by B. Fuchssteiner is presented. Applications of

[1]More general in comparison to the symmetry methods used by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13].

this result are discussed. Especially, we show, how the symmetry results by Cheb-Terrab et. al. can be used in this more general theoretical setting.

Finally, a new connection between differential Galois theory and symmetry analysis of differential equations — two seemingly disparate areas — is presented. This connection is established with the help of the theory of nilpotent and recursive flows introduced in [27] (see also [28] and [29]). It leads to a new and fast algorithm for the computation of the important symmetric powers of linear differential operators. Symmetric powers of linear differential operators play an important role for the classification of solutions of homogeneous linear ordinary differential equations in the framework of differential Galois theory (see e. g. [6], [60]).

Outline of approach

This thesis is organized in four chapters.

Chapter 1:
Symmetries

The first chapter is dedicated to symmetry methods for solving differential equations. Various notions of transformation groups (defined by how they depend on the initial solutions to which they are applied) leading to different types of symmetries like Lie point symmetries, Lie–Bäcklund symmetries, local and non–local symmetries are introduced. Basic properties of these are discussed.

The methods of Cheb-Terrab et. al. presented in [7], [9], [10], [12], [13] using Lie point symmetries for the solution of first order ordinary differential equations are briefly summarized. Afterwards, the representation of ordinary differential equations and symmetries in phase space is introduced.

A new and yet unpublished integrability result based on independent and commuting symmetry generators by B. Fuchssteiner is presented. In the last and main part of the chapter, new applications of this result are discussed. Furthermore, it is shown, how the methods by Cheb-Terrab et. al. can be embedded in the setting of the new integrability result.

Further examples for the application to higher order ordinary differential equations and systems of linear ordinary differential equations with constant coefficients are discussed. The main results of this chapter constitute the applications and examples for the new integrability result.

Finally, at the very end of the chapter, conclusions and open problems are discussed.

Chapter 2:
Integrating factors

The notions of integrating factors and exact ordinary differential equations are introduced. In the sense of the ideas of Cheb-Terrab et. al. in [11], the most general class of n-th order ordinary differential equations associated with an integrating factor is presented. The ideas of Cheb-Terrab et. al. for the computation of integrating factors of second order equations are briefly summarized.

After introducing the Euler operator as a tool for testing the exactness of ordinary differential equations, we present extensions of the methods by Cheb-Terrab et. al. for the computation of integrating factors for third order ordinary differential equations.

These results are generalized to n-th order equations, $n \in \mathbb{N}$, $n \geq 3$, in the last part of the chapter. The main results of this chapter constitute the algorithms and heuristics for the computation of integrating factors of n-th order ordinary differential equations. Finally, at the very end of the chapter, conclusions and open problems are discussed.

Chapter 3:
Skew symmetric hierarchies

Extended integrating factors and canonical forms of differential expressions are introduced. We embed these notions in a new algebraic framework, which allows the definition of a density valued scalar product. Afterwards, skew symmetric hierarchies of ordinary differential equations and their integrating factors are constructed via certain skew symmetric differential operators (skew symmetric with respect to this density valued scalar product).

The theoretical basis for the fast computation of these ordinary differential equations and their integrating factors is developed. Therefore the new notion of the fundamental form of a skew symmetric operator is introduced as an essential tool for the computation of the members of a skew symmetric hierarchy, the computation of their integrating factors and for the reduction of their order. With the help of the fundamental form, both computational tasks — the computation of integrating factors and the reduction of the order of the differential equations under consideration — are achieved without performing any integrations.

This is done in the last part of the chapter, which is dedicated to present recursive schemes for the computation of integrating factors and reductions of ordinary differential equations arising as members of skew symmetric hierarchies. The fact that integrations can be avoided via these recursive schemes makes the methods discussed in this chapter even more efficient than the algorithms presented in Chapter 2.

The main results of this chapter constitute Section 3.3, pp. 146, where the construction of skew symmetric hierarchies is developed, and Section 3.5, pp. 165, where the recursive schemes are given.

Finally, at the very end of the chapter, conclusions and open problems are stated as well as possible extensions of the methods are indicated. A link between the symmetry methods as discussed in Chapter 1 and integrating factor methods as discussed in this chapter is stated.

Chapter 4:
Non–local symmetries: A link to differential Galois theory

This chapter is dedicated to non–local symmetries as defined in the framework of Chapter 1. Mainly, non–local symmetries for ordinary differential equations are symmetries given by partial differential equations, i.e. they act on the initial solution curves in a non–local way. With the help of a special non–local symmetry for second order linear ordinary differential equations a link between the theory of nilpotent flows introduced by B. Fuchssteiner and M. Lo Schiavo and the differential Galois theory of linear differential equations is established.

This provides a new approach for the computation of the important symmetric powers of linear differential operators. Since this chapter deals with seemingly disparate areas for the treatment of ordinary differential equations, the theory of nilpotent and recursive flows in sense of the works [27], [28] and [29] is presented.

Afterwards, basic notions and results from differential Galois theory are summarized. Symmetric powers of linear differential operators are introduced and well–known applications of these for the classification of solutions appearing among homogeneous linear ordinary differential equations are summarized. A short overview on known methods for the computation of symmetric powers is given.

Then a special nilpotent flow associated with second order homogeneous linear ordinary differential equations (corresponding to the generator of a non–local symmetry group) is presented. This nilpotent flow is finally used to present an alternative approach for the computation of symmetric powers of second order homogeneous linear ordinary differential operators. This is the only case, where already a fast algorithm is available in the literature (see [6]). Our approach is more simple and even beats this extremely fast algorithm in our practical implementation.

Open problems and perspectives for further research in this area are discussed at the end of the chapter.

The notation slightly varies from chapter to chapter. E.g., in the first two chapters the dependent variable will be denoted by $y = y(x)$. In Chapter 3 we denote the dependent variable by $u = u(x)$. The reason for this is that, in the special setting of each chapter, the mostly used notation from the cited references has been adopted to make it as easy as possible to refer to the literature and to pay respect to the work already achieved by other mathematicians and by which the results of this thesis are inspired.

Furthermore, each chapter is nearly self–contained. E.g., if one is more interested in the computation of integrating factors than in symmetry methods for ordinary differential equations, Chapter 2 and Chapter 3 can be read independently of the results presented in Chapter 1 and Chapter 4 (except for the last part of Chapter 3, where a link to the symmetry methods discussed in the framework of Chapter 1 is given).

Although many algorithmical results are presented, this thesis does not contain any source code of experimental implementations, which have been done during the time within the results of this thesis have been developed. There are at least two reasons for this fact: first of all, a presentation of complete implementations would have overstepped the framework of this thesis, and, secondly, the aim is to give a formulation of the methods, which is independent of a specific computer algebra system. Hence, all algorithms in this thesis are given in pseudo–code, but the formulation is always very close to a final implementation e.g. in the framework of a general purpose computer algebra system. However, all algorithms presented in this thesis have experimentally been implemented in the computer algebra system MuPAD to get a rough impression on how these methods work in practice and to compute most of the examples given in this thesis.

A list summarizing the most important notations used in the following chapters can be found on page 221. Since the algorithms and heuristics developed in this thesis contribute a major part of the work, a glossary of algorithms can be found at the end of this thesis on page 225. Keywords and names with a central relevance for this thesis can be found using the index on page 227.

What is new in this thesis?

In Chapter 1 we state a new and yet unpublished symmetry result by B. Fuchssteiner, which allows to characterize the solutions of ordinary differential equations admitting a sufficient number of independent commuting symmetry generators. Obviously, the applications of this new result must be new. Therefore, the main new aspect in this chapter is the exploration of the applicability of this symmetry result in several examples. We show, how the results by Cheb-Terrab et. al. presented in [7], [9], [10], [12] and [13] can be used in the setting of the new symmetry result. Further applications, e.g. in the context of systems of linear differential equations with constant coefficients, are presented.

In Chapter 2, all algorithms and heuristics for the computation of integrating factors of third and higher order non–linear ordinary differential equations are new. These results are inspired by the algorithms of E. S. Cheb-Terrab et. al. for second order ordinary differential equations (see [7]).

The algorithms for the computation of integrating factors and the reduction of equations as elements of a skew symmetric hierarchy discussed in Chapter 3 are new. Especially, the algebraic setup and the notion of the fundamental form of a skew symmetric differential operator, the algorithm for the computation of this form as well as the resulting recursion formulas for symbolic integration and the reduction of ordinary differential equations presented at the end of Chapter 3 are new. These methods are generalizations of the algorithmic considerations presented in the framework of the talks "Neue Methoden zur algorithmischen Lösung nicht–linearer ODEs über Symmetrien und integrierende Faktoren" at the "Computeralgebra Tagung 2005" and "Some practical algorithms for the solution of ODEs via symmetry methods and integrating factors" at the ICMS workshop on "Algebraic Theory of Differential Equations" (see also [31] and

[35]).

Finally, Chapter 4 presents a new connection between differential Galois theory and symmetry analysis of differential equations. This new connection gives a link between two areas, which are in general considered to be disparate. With the help of this link a new, simple and fast algorithm for the computation of the important symmetric powers of second order linear differential operators is given at the end of the chapter.

Chapter 1

Symmetries

In the years 1997 and 1998, E. S. Cheb-Terrab et. al. published at least four papers introducing new algorithms and heuristics to compute solutions of ordinary differential equations (ODEs) using symmetry generators ([7], [9], [10], [12] and [13]). They implemented these algorithms and heuristics in the computer algebra system MAPLE[1].

As a benchmark for the new algorithmic ideas, Cheb-Terrab et. al. used their implementations to solve ODEs from the well–known book "Differentialgleichungen" by E. Kamke (see also [45]). The results showed that the new methods helped to improve the former ODE solver of MAPLE drastically. Detailed results of these tests and of performance tests are listed in section 4 of [7]. Usually, computing Lie point symmetries of ODEs requires solving a system of partial differential equations. The success and the efficiency of the algorithms by Cheb-Terrab et. al. is based on the fact that Lie point symmetries for wide classes of ODEs can be computed without solving any differential equations. This becomes possible by using elaborate techniques to recognize the symmetries more or less directly from the form of the ODE given, i.e. by using elaborate techniques in *pattern matching* of differential expressions. We will come back to the ideas of Cheb-Terrab et. al. in Section 1.2 of this chapter.

What we want to keep in mind is that symmetry methods in the spirit of

[1]The computer algebra system MAPLE developed by the company Maplesoft, currently available in version 10, still offers the algorithms and heuristics implemented by E. S. Cheb-Terrab et. al. in the framework of the ODETOOLS–package. The latest news concerning this package and further packages for solving partial differential equations developed and improved by Cheb-Terrab et. al. can be found on Cheb-Terrab's homepage [15].

Cheb-Terrab et. al. may be used for solving classes of ODEs in the framework of a computer algebra system without having to solve the associated systems of partial differential equations again and again. These symmetry methods for solving ODEs are not only interesting from a theoretical point of view, but also from a completely practical point of view of someone implementing algorithms in a computer algebra systems not being able (or willing due to loss of efficiency) to solve systems of partial differential equations to find solutions of ODEs. Furthermore these partial differential equations may not be solvable in general.

Nevertheless, as far as we know, Cheb-Terrab et. al. use symmetry methods mostly to solve first order ODEs. A possible reason for this is the fact that for first order ODEs, the Lie point symmetry methods as used by Cheb-Terrab et. al. and integrating factor methods as introduced in Chapter 2 are closely connected. We will come back to this statement in the framework of the conclusions for Chapter 2 in Section 2.8. They completely leave out eventually possible applications of Lie–Bäcklund symmetries instead of Lie point symmetries. These constitute for ODEs of order higher than 1 a considerable extension of the notion of symmetry. The reason for this omission may be that Lie's famous result (see [64] p. 86 or [51], Theorem 2.64, p. 155) does not apply to these symmetries. Hence, they may not be used to derive from them integrating factors in a direct way.

For higher order ODEs, Cheb-Terrab et. al. mainly propose algorithms to compute integrating factors, which can be used to reduce the order of a given higher order ODE by 1 and thereby introducing a constant of integration. We believe that one of the reasons for using integrating factors for higher order ODEs is a completely practical one: integrating factors directly help to reduce the order of a given ODE by 1, whereas in general a single Lie–Bäcklund symmetry generator does not seem to suffice to reduce the order of an ODE admitting this symmetry. Even in the case that sufficiently many symmetries admitted by a given ODE have been found, there is still a problem in characterizing the solutions of the ODE. We always have to keep in mind that we do not want to use *some arbitrary method* to compute solutions of an ODE, but a *practical method*, i.e. a method computing the results of an ODE in such a form that the user of a computer algebra system may use the result for further, practical computations. The question of the representation of solutions is a very important prerequisite for further subsequent computations.

In the following when we speak about symmetries or symmetry generators we refer to Lie–Bäcklund symmetries, which comprise Lie point symmetries as a special case. New results by B. Fuchssteiner using symmetry generators of this kind to completely characterize the solutions of ODEs (of arbitrary order and admitting sufficiently many symmetries) will help to find a new way of representing the solutions of ODEs. Since the definition of symmetries in the results of B. Fuchssteiner are formulated in a more general setting and differ from the definition of symmetries used by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13], we first present the different notions of symmetries and then show how the results already established by Cheb-Terrab et. al. can be interpreted in a more general setting and in the framework of a more general notion of integrability.

We will see that the corresponding results may not only be used for first or second order ODEs, but for general n-th order ODEs, $n \geq 1$, or even for certain systems of ODEs as long as sufficiently many symmetry generators for the ODE or the system of ODEs under consideration are known. This means that a unified theory for solving ODEs and systems of ODEs with the help of sufficiently many symmetry generators is established leading to practical methods to express solutions in terms of conserved quantities.

1.1 Generalized symmetries and their infinitesimal generators

Since in this section we want to give an intuitive understanding of different notions of symmetries, we assume for all functions, curves, and transformations considered here that they are sufficiently often differentiable. This section is based on the lectures by B. Fuchssteiner [32].

The notion of symmetry we shall build up in such generality that all notions known to us (Lie point symmetries, Lie–Bäcklund symmetries, time–dependent symmetries, local symmetries, non–local symmetries) will turn out to be special cases. In order to give a first orientational survey we present here the general setting without proving every detail. Those details explicitly needed in the sequel will be proved when needed. In case those details have to be applied in a more special situations, we will give a suitable reference to the literature.

1.1.1 Curve functions

Solutions $y(x)$ of ordinary differential equations (ODEs) are curves in x–y–space. So let us consider some arbitrary curve parametrized by some parameter t (in the following called *time*)

$$\Gamma = \{(x(t), y(t)) \mid t \text{ in some parameter space}\},$$

where the parameter space is in general a 1–dimensional manifold, e.g. the real numbers $\mathbb{R}$. If we parametrize the curve given by a function $y(x)$ by the parameter x, then we simply write

$$\Gamma = \{(x, y(x)) \mid x \in \mathbb{R}\}$$

and call that the *functional parametrization*, i.e. the parametrization of the second component by the first. In the case, functional parametrization is chosen, we also simply write y instead of $y(x)$. Whenever x and y are considered to depend in t, we will explicitly state this dependence by writing $x(t)$ and $y(t)$, such that no ambiguity in the notation will arise.

Definition 1.1. A *curve function* Q is a map

$$Q : (\Gamma, x(t), y(t)) \to \mathbb{R},$$

which does not depend on the parametrization chosen for the curve.

Example 1.2. Let us give some examples:

$$Q(\Gamma, x(t), y(t)) := x(t), \tag{1.1.1}$$

$$Q(\Gamma, x(t), y(t)) := y(t), \tag{1.1.2}$$

$$Q(\Gamma, x(t), y(t)) := \left(\frac{d}{dt}y(t)\right)\left(\frac{d}{dt}x(t)\right)^{-1}. \tag{1.1.3}$$

This last quantity is the same as the derivative

$$\frac{d}{dx}y(x) = y'(x),$$

if the functional parametrization were chosen. So, to no surprise, all higher derivatives

$$y^{(k)}(x) := \frac{d^k}{dx^k}y(x),$$

$k \in \mathbb{N}$, are fulfilling that condition. However, there are other quantities like

$$Q(\Gamma, x, y(x)) := y(x - \varphi(x)), \tag{1.1.4}$$

if functional parametrization is taken and φ is a suitable function. ◇

Definition 1.3. We call a curve function a *local curve function*, if when written in functional parametrization it depends only on x and the derivatives $y^{(k)}(x)$, $k = 0, 1, 2, \ldots$, but not on other properties of the curve.

Example 1.4. The curve functions (1.1.1) (1.1.2) and (1.1.3) are local curve functions, but (1.1.4) is not a local curve function. $\diamond$

1.1.2 Groups of transformations and symmetry groups

We are interested in transformations $\Gamma \to \tilde{\Gamma}$ from one curve Γ to another curve $\tilde{\Gamma}$ given by

$$(x(t), y(t)) \to (Q_1(\Gamma, x(t), y(t)), Q_2(\Gamma, x(t), y(t))),$$

where Q_1, Q_2 are curve functions. Even more we are interested in families of such transformations $\Gamma \to R(\Gamma, \epsilon)$ depending on some parameter ϵ given by

$$(x(t), y(t)) \to (Q_1(\Gamma, x(t), y(t), \epsilon), Q_2(\Gamma, x(t), y(t), \epsilon)). \tag{1.1.5}$$

Definition 1.5. A family of transformations (1.1.5) is said to be a *one–parameter group of transformations*, if for all Γ and ϵ_1, ϵ_2 we have

$$\begin{aligned} R(R(\Gamma, \epsilon_1), \epsilon_2) &= R(\Gamma, \epsilon_1 + \epsilon_2), \\ R(\Gamma, 0) &= \Gamma. \end{aligned}$$

Definition 1.6. A transformation (or a group of transformations, respectively) is said to be a *symmetry* (or a *symmetry group*, respectively) for a given ODE, if solution curves are mapped onto solution curves.

Symmetry groups are difficult to describe, therefore attention is turned onto their infinitesimal aspects. We consider the derivative of the right–hand–side of (1.1.5) with respect to ϵ at $\epsilon = 0$:

$$\begin{aligned} (\xi(\Gamma, x(t), y(t)), \eta(\Gamma, x(t), y(t))) := \Big(& \frac{\partial}{\partial \epsilon}_{|\epsilon=0} Q_1(\Gamma, x(t), y(t), \epsilon), \\ & \frac{\partial}{\partial \epsilon}_{|\epsilon=0} Q_2(\Gamma, x(t), y(t), \epsilon) \Big). \end{aligned}$$

Definition 1.7. The map

$$(\Gamma, x(t), y(t)) \to (\xi(\Gamma, x(t), y(t)), \eta(\Gamma, x(t), y(t)))$$

is said to be the *infinitesimal generator* of the transformation family (1.1.5).

For reasons of abbreviation, the pair of functions $\xi(\Gamma, x(t), y(t))$ and $\eta(\Gamma, x(t), y(t))$ may also be referred to as the infinitesimal generator of (1.1.5).

If the family is a group, then this generator characterizes the group uniquely, i.e. the group can be recovered by some formal type of exponentiation (which however can only be carried out by solving some differential equations). The functions coming up in the infinitesimal curve–generator obviously are curve functions as well. The name curve–generator is chosen in order to avoid mixing this up with another type of infinitesimal generator for this transformation.

Definition 1.8. A group of transformations is said to be:

(i) A *Lie point group*, if the functions ξ and η do not depend on Γ, but only on the point $(x(t), y(t))$.

(ii) An *autonomous group*, if $\xi = 0$, i.e. if when written in functional parametrization, the independent variable does not change.

(iii) A *time–independent group*, if when the curve is given in functional parametrization, the functions ξ and η only depend on y, but not explicitly on x. If not time–independent, then the group is called *time–dependent*.

(iv) A *local group* when the functions ξ and η are local curve functions, i.e. when the curve is given in functional parametrization, they only depend on y, eventually on x, and derivatives $y^{(k)}$, $k \in \mathbb{N}$, of the dependent variable y with respect to the independent variable x.

(v) A *Lie–Bäcklund group*, if it is autonomous and local.

Later on we shall give examples for all these notions, also for non–local groups (we refer to Chapter 4 for such kinds of groups).

In the sequel we shall show that non–autonomous groups always can be replaced by autonomous groups, however then for example Lie point groups become Lie–Bäcklund groups (see Section 1.4). Furthermore we shall see that time–dependent groups can be replaced by time–independent groups, for that however dimension of phase space has to be increased (see Section 1.3).

In order to keep closer to the notation used in the literature when dealing with Lie point symmetries we perform a slight change in notation. We assume (1.1.5) constitutes a group and we write

$$(x(t), y(t)) \to (\tilde{x}(\Gamma, x(t), y(t), \epsilon), \tilde{y}(\Gamma, x(t), y(t), \epsilon))$$

instead. Then using generators and choosing functional parametrization we find:

$$x \to \tilde{x} = \tilde{x}(\Gamma, x, y, \epsilon) = x + \epsilon\, \xi(\Gamma, x, y) + \text{terms of higher order in } \epsilon,$$
$$y \to \tilde{y} = \tilde{y}(\Gamma, x, y, \epsilon) = y + \epsilon\, \eta(\Gamma, x, y) + \text{terms of higher order in } \epsilon.$$

Here the precise meaning of "terms of higher order in ϵ" is that the first two terms constitute the Taylor polynomial of first order in ϵ.

The functions $\xi(\Gamma, x, y)$ and $\eta(\Gamma, x, y)$ are then in formal analogy as in the usual theory of Lie point symmetries given by

$$\xi(\Gamma, x, y) = \frac{\partial}{\partial \epsilon}_{|\epsilon=0} \tilde{x}(\Gamma, x, y, \epsilon), \tag{1.1.6}$$

$$\eta(\Gamma, x, y) = \frac{\partial}{\partial \epsilon}_{|\epsilon=0} \tilde{y}(\Gamma, x, y, \epsilon). \tag{1.1.7}$$

See e.g. [64], Section 2 of Chapter I, [4], Chapter 2, or [51], Chapter 2.

Now we consider an n-th order ODE of the form

$$H(x, y(x), y'(x), \ldots, y^{(n)}(x)) = 0, \tag{1.1.8}$$

where

$$H = y^{(n)}(x) - \Phi(x, y(x), y'(x), y''(x), \ldots, y^{(n-1)}(x)). \tag{1.1.9}$$

As before, $y^{(k)}(x)$, $k \in \mathbb{N}$, denotes the k-th derivative of the dependent variable y with respect to the independent variable x. In order to find infinitesimal criteria, whether or not a given group of transformations is a symmetry group, we have to consider (as it is customary in the Lie point transformation case), how the derivatives $y^{(k)}(x)$ change under such transformations. This leads to *prolongations*.

1.1.3 Prolongations

We consider an arbitrary transformation with respect to an arbitrary parametrization

$$(x(t), y(t)) \to (\tilde{x}(\Gamma, x(t), y(t), \epsilon), \tilde{y}(\Gamma, x(t), y(t), \epsilon))$$

and compute first $\tilde{y}^{(k)}$, which shall denote the k-th derivative of the curve in functional parametrization, i.e. the k-th derivative of $\tilde{y}$ with respect to $\tilde{x}$. By the chain rule we obtain along the curve

$$\frac{d}{d\tilde{x}} = \left(\frac{d\tilde{x}}{dt}\right)^{-1} \frac{d}{dt}.$$

From this we find

$$\tilde{y}^{(k)} = \left(\left(\left(\frac{d\tilde{x}}{dt}\right)^{-1} \frac{d}{dt}\right)^{k} \tilde{y}(\Gamma, x(t), y(t), \epsilon)\right).$$

The derivative of that with respect to ϵ at $\epsilon = 0$ we call the *k-th prolongation of η* denoted by $\eta^{[k]}(\Gamma, x(t), y(t))$, i.e.

$$\eta^{[k]}(\Gamma, x(t), y(t)) = \frac{\partial}{\partial\epsilon}_{|\epsilon=0} \left(\left(\left(\frac{d\tilde{x}}{dt}\right)^{-1} \frac{d}{dt}\right)^{k} \tilde{y}(\Gamma, x(t), y(t), \epsilon)\right). \tag{1.1.10}$$

Contrary to the literature (see e.g. Chapter I of [64]), we here put the index k into square brackets, this in order to avoid misinterpreting $\eta^{[k]}(\Gamma, x(t), y(t))$ as a k-th derivative.

Now we have a look at $\eta^{[k+1]}$ given by

$$\eta^{[k+1]}(\Gamma, x(t), y(t)) = \frac{\partial}{\partial\epsilon}_{|\epsilon=0} \left(\frac{d\tilde{x}}{dt}\right)^{-1} \left(\frac{d}{dt}\left(\left(\frac{d\tilde{x}}{dt}\right)^{-1} \frac{d}{dt}\right)^{k} \tilde{y}(\Gamma, x(t), y(t), \epsilon)\right).$$

Now using the product rule for the ϵ-derivative for this as a product of two factors, choosing for Γ functional parametrization and keeping in mind that $\tilde{x}(\epsilon = 0) = x$, $\tilde{y}(\epsilon = 0) = y$, then we obtain the recursion formula

$$\eta^{[k+1]}(\Gamma, x, y(x)) = \frac{d}{dx}\eta^{[k]}(\Gamma, x, y(x)) + \left(\frac{\partial}{\partial\epsilon}_{|\epsilon=0} \left(\frac{d\tilde{x}}{dx}\right)^{-1}\right) y^{(k+1)}.$$

Now using

$$\frac{\partial}{\partial\epsilon}_{|\epsilon=0} \left(\frac{d\tilde{x}}{dx}\right)^{-1} = -\frac{d\xi}{dx},$$

we obtain the final result

$$\eta^{[k+1]}(\Gamma, x, y(x)) = \frac{d}{dx}\eta^{[k]}(\Gamma, x, y(x)) - \frac{d\xi}{dx} y^{(k+1)}. \tag{1.1.11}$$

Comparing this with the literature ([64], Section 2.3 of Chapter I, [51], Section 2.3, or [4], Section 2.3.2), where the Lie point case is treated, we see that the prolongation formulas do not differ from this more special case[2].

Prolongations are needed when infinitesimal changes of local curve functions are to be computed. Consider a local curve function

$$P := P(x, y(x), y'(x), \ldots, y^{(n)}(x)),$$

$n \in \mathbb{N}$, on a curve Γ translated by ϵ under some one–parameter group of transformations with ξ, η given by (1.1.6) and (1.1.7), i.e. on the curve given by the points

$$(\tilde{x}(\epsilon), \tilde{y}(\epsilon)) := (\tilde{x}(\Gamma, x(t), y(t), \epsilon), \tilde{y}(\Gamma, x(t), y(t), \epsilon)).$$

So, we consider P depending on ϵ

$$P(\epsilon) := P(\tilde{x}, \tilde{y}(\tilde{x}), \tilde{y}'(\tilde{x}), \ldots, \tilde{y}^{(n)}(\tilde{x}))$$

and want to compute its derivative with respect to ϵ at $\epsilon = 0$. Then obviously by the chain rule we obtain

$$\frac{\partial}{\partial \epsilon}_{|\epsilon=0} P(\epsilon) = \xi(\Gamma, x, y(x)) \frac{\partial P}{\partial x} + \sum_{k=0}^{\infty} \eta^{[k]}(\Gamma, x, y(x)) \frac{\partial P}{\partial y^{(k)}}.$$

Of course, this formally infinite series stops after the highest derivatives in P have been encountered, i.e. in the above notation the index k reaches from 0 to n.

Definition 1.9. The operator

$$X = \xi(\Gamma, x, y(x)) \frac{\partial}{\partial x} + \sum_{k=0}^{\infty} \eta^{[k]}(\Gamma, x, y(x)) \frac{\partial}{\partial y^{(k)}} \tag{1.1.12}$$

is called the *infinitesimal generator of the one–parameter group (on local quantities)*.

Since the prolongations $\eta^{[k]}$, $k \in \mathbb{N}$, are completely determined by $\xi(\Gamma, x, y(x))$ and $\eta(\Gamma, x, y(x))$ and (1.1.11), we may also write for abbreviation

$$X = \xi(\Gamma, x, y(x)) \frac{\partial}{\partial x} + \eta(\Gamma, x, y(x)) \frac{\partial}{\partial y}$$

[2] In Section 3.7 of [3] an implementation of the prolongation formula for generators of Lie point symmetries in the computer algebra system MATHEMATICA is discussed.

or

$$X = (\xi(\Gamma, x, y(x)), \eta(\Gamma, x, y(x))).$$

For an intuitive understanding of further considerations it is helpful to know the following basic principle:

Remark 1.10. Consider some manifold M (manifold variable denoted by u) and a submanifold M_0 (all sufficiently often differentiable) and some one–parameter group of transformations $R(\epsilon)$ on M. Then if and only if the vector field

$$\frac{\partial}{\partial \epsilon}_{|\epsilon=0} R(\epsilon)(u)$$

is tangential to M_0 for all points on M_0, then the group of transformations $R(\epsilon)$ leaves M_0 invariant[3].

Now we return to the differential equations of the form (1.1.8), (1.1.9) respectively, consider H as a local curve function, take M to be the infinite dimensional manifold of all suitably often differential curves given by functions $y(x)$ and define M_0 to be the submanifold, where $H = 0$, i.e. the solutions of the ODE under consideration. Then application of the remark yields:

Theorem 1.11. *A general one–parameter group of curve transformations with infinitesimal generator X (see (1.1.12)) is a symmetry group for the ODE $H = 0$ (see (1.1.8), (1.1.9)) if and only if* [4]

$$XH(x, y, y', y'', \ldots, y^{(n)}) = 0 \mod H = 0. \tag{1.1.13}$$

Proof. For a proof of this theorem in case of Lie point transformations see [64], p. 19. □

Note that (1.1.13) is equivalent to saying

$$X\Phi(x, y, y', y'', \ldots, y^{(n)}) = \eta^{[n]} \mod y^{(n)} = \Phi \tag{1.1.14}$$

by (1.1.9).

[3]Proving this fact here is not really necessary, because whenever we need it for special cases we will give explicit proofs or references. Furthermore, since proving this to its full extent, i.e. on infinite dimensional manifolds, we would need a tedious amount of infinite–dimensional manifold theory, which would go beyond the aim of this thesis.

[4]Here, $XH = 0 \bmod H = 0$ is just the customary way of saying that the expression XH has to be zero on the submanifold given by $H = 0$.

If X is the infinitesimal generator of a Lie point symmetry group, i.e. an operator, where the functions ξ and η only depend on the points x, y, satisfying (1.1.13), then X is also referred to as an *admissible Lie point operator* for the equation $H = 0$ (see also Section 1.2.1 of the PhD thesis by A. S. Fokas [21]).

Applications on how to solve ODEs using Lie point symmetries are discussed in general in [64] and [51] (see there especially Lie's Theorem in [64] p. 86 or [51], Theorem 2.64, p. 155).

Equation (1.1.13) is called a *determining (partial differential) equation* for the functions $\xi(x, y)$, $\eta(x, y)$, $\eta^{[k]}(x, y, y', \ldots, y^{(k)})$, $1 \leq k \leq n$ (see e.g. Section 9.2.2 of [43]). By Theorem 1.11, one can compute the infinitesimal generator of a symmetry group by solving the determining partial differential equation. However, this result is rather of theoretical interest for the framework of this thesis, since in most cases solving (1.1.13) for $\xi(x, y)$, $\eta(x, y)$, $\eta^{[k]}(x, y, y', \ldots, y^{(k)})$, $1 \leq k \leq n$, is as difficult as solving the original ODE (1.1.8).

Of course, the notion of a generator also makes sense in case a corresponding group is not known and may eventually require solving complicated ODEs or partial differential equations. Let us complete the notions for this more general case.

Consider two curve functions

$$(\xi(\Gamma, x, y), \eta(\Gamma, x, y)).$$

By these we consider an ϵ–linear change of the curve by (thus considering (ξ, η) as the generator of a perturbation of the curve)

$$x \to \tilde{x} = \tilde{x}(x, y, \epsilon) = x + \epsilon\, \xi(\Gamma, x, y), \tag{1.1.15}$$

$$y \to \tilde{y} = \tilde{y}(x, y, \epsilon) = y + \epsilon\, \eta(\Gamma, x, y), \tag{1.1.16}$$

and we form the prolongations in the same way as above, i.e. by using the formula (1.1.10).

Definition 1.12. Then the operator given by (1.1.12) is said to be an *infinitesimal symmetry generator* (or for short *symmetry generator*) for the ODE (1.1.8), respectively (1.1.9), if (1.1.13) holds.

In this situation any solution of the ODE (1.1.8), respectively (1.1.9), is mapped by (1.1.15), (1.1.16) onto a curve, which fulfills the equation up to first order in ϵ.

Now the following is obvious:

Remark 1.13. For a given ODE, the symmetry generators form a vector space, i.e. sums, differences and scalar multiples are again symmetry generators. This observation is a direct consequence of (1.1.13).

Lemma 1.14. *There is one universal symmetry generator for all ODEs (1.1.8), respectively (1.1.9), namely*

$$(\xi(\Gamma, x, y), y'\,\xi(\Gamma, x, y)), \tag{1.1.17}$$

where $\xi(\Gamma, x, y)$ *is some curve function and* $y = y(x)$, *i.e. functional parametrization is chosen.*

Proof. The k-th prolongation of

$$\eta = y'\,\xi(\Gamma, x, y)$$

is given by

$$\eta^{[k]} = \xi y^{(k+1)}.$$

This assertion follows using induction by k. For $k = 1$ and by using (1.1.11), we obtain

$$\begin{aligned}\eta^{[1]} &= \frac{d}{dx}\eta - \frac{d\xi}{dx}y' \\ &= \frac{d\xi}{dx}y' + \xi y'' - \frac{d\xi}{dx}y' \\ &= \xi y''.\end{aligned}$$

The same arguments provide the desired result for $\eta^{[k]}$, $k \geq 1$. Choosing X as in (1.1.12), application of X to the ODE $H = 0$ yields

$$\begin{aligned}XH &= \xi\,\frac{\partial H}{\partial x} + \sum_{k=1}^{\infty} \xi\, y^{(k+1)}\,\frac{\partial H}{\partial y^{(k)}} \\ &= \xi\frac{d}{dx}H,\end{aligned}$$

which clearly is zero. □

Finding for such a generator the corresponding group amounts in the local case to finding general solutions of systems of ODEs (phase space). For the non–local case often the group can be determined by solving partial differential equations (see also Chapter 4).

1.1.4 Lie point symmetries and first order ODEs

We close this section with a remark on how to use Lie point symmetries to tackle first order ODEs.

Remark 1.15. In the case of a first order ODE

$$y'(x) = \Phi(x, y(x)) \tag{1.1.18}$$

the determining equation for the infinitesimals $\xi(x, y)$ and $\eta(x, y)$ of a Lie point symmetry admitted by (1.1.18) reads

$$\eta_x + (\eta_y - \xi_x)\Phi - \xi_y\Phi^2 - \xi\Phi_x - \eta\Phi_y = 0. \tag{1.1.19}$$

This equation is obtained by applying the symmetry generator $X = \xi(x, y)\frac{\partial}{\partial x} + \eta(x, y)\frac{\partial}{\partial y}$ to the right–hand–side of (1.1.18) and afterwards setting the result equal to zero (modulo (1.1.18)). Hence, given the ODE (1.1.18), computing a Lie point symmetry means solving the partial differential equation (1.1.19) for $\xi(x, y)$ and $\eta(x, y)$. In terms of $\xi(x, y)$, $\eta(x, y)$ and $\Phi(x, y)$ the formal implicit solution of (1.1.18) can be given in terms of the line integral

$$\int \frac{dy - \Phi\, dx}{\eta - \xi\, \Phi} = c, \tag{1.1.20}$$

c a constant (see [10] or [4], Section 3.2 concerning solution formulas for first order ODEs of type (1.1.18) using the generator of a Lie point symmetry).

If we denote the solution of (1.1.18) written in implicit form (1.1.20) by $\varphi(x, y)$, then there are "constants of integration" $c_1(y)$ and $c_2(x)$, such that

$$\begin{aligned} \varphi(x, y) &= -\int \frac{\Phi(x, y)}{\eta(x, y) - \xi(x, y)\Phi(x, y)} dx + c_1(y), \\ \varphi(x, y) &= \int \frac{1}{\eta(x, y) - \xi(x, y)\Phi(x, y)} dy + c_2(x). \end{aligned}$$

The "constants of integration" $c_1(y)$ and $c_2(x)$ may be computed by comparing the two representations of $\varphi(x, y)$ and, hence, a more concrete representation for $\varphi(x, y)$ may be found.

Despite of the compact formula for characterizing the solutions of (1.1.18), the authors of [10] remark that there are no general rules helping to solve (1.1.19) for any given Φ.

1.2 The methods of Cheb-Terrab et. al. for first order ODEs

In this section we give a summary of the ideas of Cheb-Terrab et. al. to compute symmetries of first order ODEs. We mainly refer to the work presented in [10]. The basic algorithmic idea of Cheb-Terrab et. al. to solve first order ODEs (1.1.18) using Lie point symmetries can be summarized as follows: *instead of trying to find a symmetry of a given ODE from the determining equation (1.1.19), try to find out if the given ODE belongs to a class of ODEs admitting a certain form of symmetries.* This informal statement and the approach suggested by Cheb-Terrab et. al. is best explained by considering a concrete example.

Assume, we want to compute the infinitesimal generator $X = \xi(x,y)\frac{\partial}{\partial x} + \eta(x,y)\frac{\partial}{\partial y}$ of a Lie point symmetry of a first order ODE (1.1.18). Furthermore, assume, the ODE admits a symmetry of the form

$$X = (F(x) + G(y))\frac{\partial}{\partial x}, \tag{1.2.1}$$

i.e. $\xi(x,y) = F(x) + G(y)$ and $\eta(x,y) = 0$. From this specialized form of the symmetries and the determining equation (1.1.19), Cheb-Terrab et. al. once compute the most general class of ODEs admitting a symmetry of this form by hand (as symbolic pre–processing), i.e. they insert the special form of ξ and η into the determining equation (1.1.19) for the infinitesimals and solve this equation for Φ. This provides a representation of Φ in terms of the symmetry generator:

$$y'(x) = \Phi(x,y) = \left(\left(G(y)_y \int \frac{dx}{(F(x)+G(y))^2} - I(y)\right)(F(x)+G(y))\right)^{-1}, \tag{1.2.2}$$

where $I(y)$ is as an arbitrary function in its argument[5].

In the next step, Cheb-Terrab et. al. establish a necessary and sufficient criterion to decide, whether a given concrete first order ODE of the form (1.1.18) belongs

[5] The function or expression $I(y)$ can be viewed a constant of the integration arising from the integration of $\frac{1}{(F(x)+G(y))^2}$ with respect to x.

to the class of first order ODEs given by (1.2.2). The first observation is that whenever a given ODE belongs to the class (1.2.2), one obtains

$$\Phi \frac{\partial^2}{\partial x^2}\Big(\frac{1}{\Phi}\Big) = \frac{\frac{\partial^2}{\partial x^2}F(x)}{F(x)+G(y)}. \tag{1.2.3}$$

Hence, a necessary condition for a given ODE of the form (1.1.18) to belong to the class (1.2.2) is that the differential expression

$$\begin{aligned} \frac{\partial}{\partial y}\bigg(\Big(\Phi \frac{\partial^2}{\partial x^2}\Big(\frac{1}{\Phi}\Big)\Big)^{-1}\bigg) &= \frac{\partial}{\partial y}\Big(\frac{F(x)+G(y)}{\frac{\partial^2}{\partial x^2}F(x)}\Big) \\ &= \frac{\frac{\partial}{\partial y}G(y)}{\frac{\partial^2}{\partial x^2}F(x)} \end{aligned} \tag{1.2.4}$$

is a product of a function depending only on y and a function depending only on x, namely the product of $\frac{\partial}{\partial y}G(y)$ and the reciprocal of $\frac{\partial^2}{\partial x^2}F(x)$.

Assume, for a given first order ODE of the form (1.1.18) we find that (1.2.4) separates by product in the above mentioned sense. Then the reciprocal of the factor depending only on x is a candidate for the second derivative of a function $F(x)$, which may be an essential part of a symmetry generator for the given ODE. Now, multiplication of the factor arising from (1.2.4) only depending on x with (1.2.3) and afterwards computing the reciprocal of the result obtained gives a candidate for $F(x)+G(y)$.

If (1.2.4) separates by product in x and y and a candidate for $F(x)+G(y)$ is computed in the above mentioned manner, then a sufficient condition for the ODE under consideration to admit a Lie point symmetry with infinitesimal generator $X = (F(x)+G(y))\frac{\partial}{\partial x}$ is that the expression

$$\frac{\partial}{\partial x}\Big((F(x)+G(y))\Phi\Big) + \Phi^2 \frac{\partial}{\partial y}\Big(F(x)+G(y)\Big) \tag{1.2.5}$$

is equal to zero. The correctness of this criterion can be seen from the proofs of Proposition 3 and Proposition 4 in [10].

What is remarkable is that the sufficient and necessary criterion to decide, whether a given first order ODE of the form (1.1.18) admits a Lie point symmetry of the form (1.2.1), does not require solving any auxiliary ODEs or even partial differential equations to find a symmetry generator of the prescribed form. This fact is indeed the main aspect, why the algorithms presented in [10] are so efficient. According to the authors of [10], many of the ODEs presented in [45] admit relatively simple symmetry patterns such as (1.2.1).

Of course, the authors of [10] succeeded in classifying many further ODEs due to other *symmetry patterns*. To name a few more: the patterns for Lie point symmetry generators of the form $X = (F(x) + G(y))\frac{\partial}{\partial y}$, $X = F(x)G(y)\frac{\partial}{\partial x}$, $X = F(x)G(y)\frac{\partial}{\partial y}$, $X = F(x)\frac{\partial}{\partial x} + G(x)\frac{\partial}{\partial y}$, $X = F(y)\frac{\partial}{\partial x} + G(y)\frac{\partial}{\partial y}$, etc. and the corresponding classes of ODEs admitting such generators have successfully been classified.

We do not state these results here in detail. The main point to keep in mind is that in the case of first order ODEs of the form (1.1.18), efficient methods to compute symmetries and thereby solutions of the ODEs are still available. Additionally, these methods provide algorithms to compute symmetry generators, which are relatively simple to implement in any computer algebra system.

Once, the symmetry generator of a given ODE is detected, the solutions can be represented in implicit form using (1.1.20).

Note that the results presented in this section are only applicable to first order ODEs of the form (1.1.18). Cheb-Terrab et. al. also published some results concerning the computation of Lie point symmetries of second order ODEs (see [9]). Unfortunately, as far as we know, the results concerning the solution of second order ODEs of the form $y''(x) = \Phi(x, y(x), y'(x))$ using symmetry methods and elaborate pattern matching as in the case of first order ODEs are not published in detail. The paper [9] presenting the ideas for second order ODEs is more a technical documentation of a special implementation in the computer algebra system MAPLE than a detailed description of the underlying mathematical ideas.

The reasons for not giving a detailed description of the ideas seem unclear. We suppose that the methods using Lie point symmetries to compute solutions of ODEs of order two or even higher order did not produce the desired results in practice or are at least not that efficient as desired. The reason for this fact certainly may be that because of the relevance of Lie–Bäcklund symmetries for these equations the class of equations, which can be solved by Lie point symmetries, is not as exhaustive as in the first order case. But this is speculation. As computational tasks the authors mention in general the determining of certain differential invariants, the determining of first integrals and, finally, the determining of canonical coordinates for the associated Lie group. The emphasis lies on the word *group*, since the knowledge of the complete symmetry group allows to reduce the order of the ODE under consideration by 1 (see e.g. [51], page 140 ff.).

Cheb-Terrab et. al. seem to favor integrating factor methods for treating ODEs of order at least 2 as it becomes clear by further publications, where symmetry methods do not seem to obtain further attention by the authors.

We will now consider evolution equations in phase space and symmetries of such equations. This provides the general setting for formulating the new and yet unpublished results by B. Fuchssteiner concerning the integration of higher order ODEs, when sufficiently many symmetry generators are available.

1.3 Symmetries in the context of evolution equations

Let M be some C^∞-manifold, the manifold variable we denote by u, and $u(t)$, $t \in \mathbb{R}$, denotes the variable for a C^∞-curve on M, i.e. the variable on the manifold of all C^∞-curves on M. For abbreviation we use u_t to denote the derivative of $u(t)$ with respect to t. So, in order to avoid overabundance of notation when it is obvious that a curve is meant we write u instead of $u(t)$. We consider the evolution equation

$$u_t = K(u), \tag{1.3.1}$$

where $K(u)$ is some vector field on M. In the following we deal with symmetries of equations of the form (1.3.1).

Note that the right–hand–side of (1.3.1), i.e. the vector field $K(u)$, is assumed not to depend explicitly on the independent variable t. In the literature, e.g. in the well–known book by P. J. Olver [51], one distinguishes between the notion of time–dependent symmetries and symmetries, which do not explicitly depend on the independent variable.

However it turns out that by introducing an additional component for the points on the manifold one is able to consider any time–dependent symmetry as a time–independent symmetry on this modified manifold. So for our purposes and in the whole course of this thesis it will turn out to be sufficient to consider only symmetries and evolution equations not explicitly depending on the independent variable, since we can always switch to phase space representation with one component more (see also Remark 1.19 below).

We consider one–parameter groups of C^∞–diffeomorphisms on the manifold M, which for reasons of abbreviation will simply be referred to as diffeomorphisms. A C^∞–diffeomorphism is a one–to–one infinitely often differentiable map such that its inverse is again infinitely often differentiable. A *one–parameter diffeomorphism group* is a map

$$(u,t) \to R(t)(u)$$

infinitely often differentiable on the product $M \times \mathbb{R} = \{(u,t) \mid u \in M,\, t \in \mathbb{R}\}$ assigning to each $t \in \mathbb{R}$ a diffeomorphism $R(t)$ on M such that

$$\begin{aligned} R(t_1 + t_2) &= R(t_1) \circ R(t_2), \\ R(0) &= \mathrm{id}, \end{aligned}$$

for all $t_1, t_2 \in \mathbb{R}$, where $\circ$ denotes the composition of maps and id the identity map[6].

Remark 1.16. Transformation groups on M and equations of the form (1.3.1) are closely connected in the following way: If (1.3.1) has for all t and all initial conditions $u(t=0) = u_0$ a unique solution[7] denoted by $u(t, u_0)$, then the map

$$(u_0, t) \to R(t)(u_0) := u(t, u_0)$$

defines a transformation group, whose infinitesimal generator is $K(u)$.

The proof is a simple argument coming from the fact that the right–hand–side of (1.3.1) does not explicitly depend on t, i.e. the solution for the initial condition $u(t = t_0) := u(t_0, u_0)$ is the same as $u(t, u_0)$.

If $R(t)$, $t \in \mathbb{R}$, is a one–parameter diffeomorphism group, then

$$\Gamma(u) := \frac{d}{dt}_{|t=0} R(t)$$

[6] This implies that all the maps $R(t)$ commute and that $R(-t)$ is the inverse of $R(t)$, $t \in \mathbb{R}$.

[7] We do not discuss theoretical uniqueness conditions in the framework of this thesis. There are generalizations, e.g. of the well–known *Picard Theorem for ODEs*, which can be applied in the general setting of evolution equations (see e.g. [1], Chapter 2).

is again said to be its *infinitesimal generator*. The generator Γ may be computed using the directional derivative

$$\Gamma(u) = \frac{\partial}{\partial \epsilon}_{|\epsilon=0} \left(R(t+\epsilon)(u)\right)_{|t=0} . \tag{1.3.2}$$

By $[\![\Gamma, K]\!]$ we denote the *commutator of the vector fields* K *and* Γ defined as

$$\begin{aligned} [\![\Gamma, K]\!](u) &:= K'(u)[\Gamma(u)] - \Gamma'(u)[K(u)] \\ &:= \frac{\partial}{\partial \epsilon}_{|\epsilon=0} \big(K(u + \epsilon\, \Gamma(u)) - \Gamma(u + \epsilon\, K(u))\big). \end{aligned} \tag{1.3.3}$$

The derivatives $K'(u)[\Gamma(u)]$ and $\Gamma'(u)[K(u)]$ are also referred to as directional derivatives in the literature (see e.g. [51], Section 4.1 on variational calculus).

A one–parameter diffeomorphism group $R(t)$, $t \in \mathbb{R}$, is said to be a *symmetry group* for (1.3.1), if it maps solutions of (1.3.1) to solutions of (1.3.1).

For details on the above stated definitions we refer to [51].

The following theorem gives exactly that kind of characterization of symmetries of evolution equations, which will be used for the rest of this chapter.

Theorem 1.17. *The generator* $\Gamma(u)$ *of a one–parameter diffeomorphism group is the generator of a symmetry group for (1.3.1) if and only if* $[\![\Gamma, K]\!] = 0$.

Proof. The proof follows from [51], Proposition 5.19, p. 310. □

Methods on how to use symmetries in the context of evolution equations to compute solutions of such equations are discussed in detail in [51].

Again, especially in view of Remark 1.16, the notion of a generator also makes sense in case a corresponding group is not known and may require solving further ODEs or partial differential equations:

Definition 1.18. A C^∞–vector field $\Gamma(u)$ is said to be an *infinitesimal symmetry generator* (or for short *symmetry generator*) for (1.3.1), if $[\![\Gamma, K]\!] = 0$ holds.

Remark 1.19. Any n-th order ODE of the form (1.1.8), respectively (1.1.9), can be written in phase space notation as

$$\underbrace{\begin{pmatrix} y \\ y' \\ \vdots \\ y^{(n-1)} \\ x \end{pmatrix}_t}_{=:\vec{u}_t} = \underbrace{\begin{pmatrix} y' \\ y'' \\ \vdots \\ \Phi(x, y, y', \ldots, y^{(n-1)}) \\ 1 \end{pmatrix}}_{=:\vec{K}(\vec{u})} \tag{1.3.4}$$

with initial condition $x(t = 0) = 0$. Hence, the solutions of (1.3.4) and (1.1.9) differ only insofar as (1.3.4) contains in addition to those solutions from (1.1.9) all those solutions obtained by a translation of the independent variable x.

How Lie point symmetries of n-th order ODEs can be transformed to symmetries of the corresponding evolution equations in phase space, is summarized in the next section.

1.4 Transformation of Lie point symmetries to phase space

Since we want to view any n-th order ODE as an evolution equation in phase space, we need a way to transform Lie point symmetries to symmetries of the type presented in the preceding section. The problem is that in the definition of Lie point symmetries we considered point transformations affecting the dependent variable $y = y(x)$ and the independent variable x. In the above presented symmetry definition 1.18 for equations of type (1.3.1), the symmetry generator does not explicitly depend on the independent variable: the symmetries in the context of evolution equations do not transform the independent variable (time–independent symmetry).

Hence, we first have to present a way how to transform a Lie point symmetry generator arising from a point transformation affecting the independent variable to a Lie–Bäcklund symmetry defined by another transformation not affecting the independent variable. This transformation is given in the following lemma[8]:

[8] See also Theorem 5.2.3-1. in Section 5 on Noether's Theorem and Lie–Bäcklund Symmetries of [4]. Lemma 1.20 is special case of this result. We give a more elementary proof of this result than the one presented in [4].

Lemma 1.20. *The generator*

$$(\xi(x,y),\eta(x,y))$$

is a symmetry generator for (1.1.8), respectively (1.1.9), if and only if

$$(0,\eta(x,y)-\xi(x,y)\,y') \tag{1.4.1}$$

is a symmetry generator for (1.1.8), respectively (1.1.9).

Proof. By Lemma 1.14, $(\xi(x,y),y'\,\xi(x,y))$ is a symmetry generator (universal symmetry generator) for (1.1.8), respectively (1.1.9). Hence, using Remark 1.13, $(\xi(x,y),\eta(x,y))$ is a symmetry generator for (1.1.8), respectively (1.1.9), if and only if $(0,\eta(x,y)-\xi(x,y)\,y')$ is a symmetry generator for the ODE. □

Lemma 1.20 implies that whenever $X=(\xi(x,y),\eta(x,y))$ is the generator of a Lie point symmetry group in the sense of Definition 1.8 (i), then $\tilde{X}=(0,\eta(x,y)-\xi(x,y)\,y')$ is a generator of the Lie–Bäcklund symmetry group in the sense of Definition 1.8 (v), i.e. the generator of an autonomous and local symmetry group (which might of course not be known explicitly).

Now each of the symmetry generators of the type (1.4.1) does not affect the independent variable x. Hence, we can interpret each of these as a symmetry of the general type presented in the preceding section in the context of evolution equations.

Proposition 1.21. *Let*

$$X=(\eta(x,y)-\xi(x,y)\,y')\frac{\partial}{\partial y}$$

be a symmetry generator of (1.1.8), respectively (1.1.9), and set

$$\vec{u}=\begin{pmatrix} y\\ y'\\ \vdots\\ y^{(n-1)}\\ x\end{pmatrix}$$

as in Remark 1.19. Then

$$\vec{\Gamma}(\vec{u})=\begin{pmatrix} \eta-\xi y'\\ \eta^{[1]}-\xi y''\\ \vdots\\ \eta^{[n-1]}-\xi\Phi\\ 0\end{pmatrix} \tag{1.4.2}$$

is a symmetry generator for the evolution equation (1.3.4) in the sense of Definition 1.18, where $\eta^{[1]}, \eta^{[2]}, \ldots, \eta^{[n-1]}$ *are to be computed using recursion formula (1.1.11) for the prolongations.*

Proof. We have to show that $[\![\vec{\Gamma}, \vec{K}]\!]$ is the zero vector. We have

$$[\![\vec{\Gamma}, \vec{K}]\!] = \left[\!\!\left[\begin{pmatrix} \eta - \xi y' \\ \eta^{[1]} - \xi y'' \\ \vdots \\ \eta^{[n-1]} - \xi \Phi \\ 0 \end{pmatrix}, \begin{pmatrix} y' \\ y'' \\ \vdots \\ \Phi(x, y, y', \ldots, y^{(n-1)}) \\ 1 \end{pmatrix} \right]\!\!\right].$$

The first component of the above vector is given by

$$\begin{aligned}
&1[\eta^{[1]} - \xi y''] - (\eta - \xi y')_y[y'] - (\eta - \xi y')_{y'}[y''] - (\eta - \xi y')_x[1] \\
&= \eta^{[1]} - \xi y'' - (\eta_y - \xi_y y')y' + \xi y'' - \eta_x + \xi_x y' \\
&= \eta^{[1]} + y' \frac{d\xi}{dx} - \frac{d\eta}{dx} \\
&= 0
\end{aligned}$$

by (1.1.11). For the second component we obtain

$$\begin{aligned}
&1[\eta^{[2]} - \xi y'''] - (\eta^{[1]} - \xi y'')_y[y'] - (\eta^{[1]} - \xi y'')_{y'}[y''] - (\eta^{[1]} - \xi y'')_{y''}[y''']- \\
&(\eta^{[1]} - \xi y'')_x[1] \\
&= \eta^{[2]} - \xi y''' - (\eta^{[1]}_y - \xi_y y'')y' - \eta^{[1]}_{y'} y'' + \xi y''' - \eta^{[1]}_x + \xi_x y'' \\
&= \eta^{[2]} + y'' \frac{d\xi}{dx} - \frac{d\eta^{[1]}}{dx} \\
&= 0
\end{aligned}$$

again by (1.1.11). By the same simple computations, the first $n-2$ components of the commutator of $\vec{\Gamma}$ and $\vec{K}$ vanish. The last component vanishes, since the n-th components of $\vec{\Gamma}$ and $\vec{K}$ are constant. The $(n-1)$-st component of $[\![\vec{\Gamma}, \vec{K}]\!]$ is given by

$$\begin{aligned}
&\Phi_y[\eta - \xi y'] + \Phi_{y'}[\eta^{[1]} - \xi y''] + \ldots + \Phi_{y^{(n-1)}}[\eta^{[n-1]} - \xi\Phi] - (\eta^{[n-1]} - \xi\Phi)_y[y']- \\
&(\eta^{[n-1]} - \xi\Phi)_{y'}[y''] - \ldots - (\eta^{[n-1]} - \xi\Phi)_{y^{(n-1)}}[\Phi] - (\eta^{[n-1]} - \xi\Phi)_x[1] \\
&= \xi\Phi_x + \eta\Phi_y + \eta^{[1]}\Phi_{y'} + \ldots + \eta^{[n-1]}\Phi_{y^{(n-1)}} - \frac{d\eta^{[n-1]}}{dx} + \Phi\frac{d\xi}{dx} \\
&= \eta^{[n]} - \frac{d\eta^{[n-1]}}{dx} + \Phi\frac{d\xi}{dx} \\
&= 0,
\end{aligned}$$

where we have used Theorem 1.11, to conclude that $\xi\Phi_x + \eta\Phi_y + \eta^{[1]}\Phi_{y'} + \ldots + \eta^{[n-1]}\Phi_{y^{(n-1)}} = \eta^{[n]}$, and again (1.1.11). □

Lemma 1.20 and Proposition 1.21 directly imply

Theorem 1.22. *Any symmetry generator of the form $(\xi(x,y), \eta(x,y))$ of an ODE of the form (1.1.8), respectively (1.1.9), can be transformed to a symmetry generator of the corresponding evolution equation (1.3.4).*

Remark 1.23. If a symmetry generator of (1.1.8), respectively (1.1.9), is of the form

$$(0, \eta(x,y)),$$

one simply has to set $\xi(x,y) = 0$ in Proposition 1.21. Then the formulas are still correct, i.e. the symmetry generator (1.4.2) of the corresponding evolution equation (1.3.4) is given by

$$\vec{\Gamma}(\vec{u}) = \begin{pmatrix} \eta \\ \eta^{[1]} \\ \vdots \\ \eta^{[n-1]} \\ 0 \end{pmatrix}.$$

1.5 Complete integration via commuting symmetry generators

In this section we discuss a new and yet unpublished result established by B. Fuchssteiner and present some new applications of this result to certain classes of differential equations. The result can be formulated in the general setting of evolution equations. So we first have to fix some notations and to summarize some well–known results to be used later on.

1.5.1 Tensors, Lie derivatives and exterior derivatives

The following definitions and results from the fields of differential forms and tensor analysis can be found e.g. in [57]. We only give a very rough and short summary to fix the notation concerning those aspects, which we will need later on.

Let now the manifold M be some $(n+1)$-dimensional vector space[9] V. By $\mathcal{L}(V)$ we denote the *set of all vector fields* on V, by $\widehat{\mathcal{L}}(V)$ the *set of all covector fields* on V and by $\mathcal{F}(V)$ the *set of all scalar fields* on V. Vector fields will be written as column vectors, whereas covector fields will be represented by row vectors. The application of a covector field $\Gamma \in \widehat{\mathcal{L}}(V)$ to some vector field $K \in \mathcal{L}(V)$ will be denoted by $\langle \Gamma, K\rangle$. The *set of all n-times-covariant and m-times-contravariant tensors* on V is denoted by $\mathcal{T}_{(n,m)}(V)$. For a fixed $\Gamma(u) \in \widehat{\mathcal{L}}(V)$ the map defined by

$$T_1 : \mathcal{L}(V) \to \mathcal{F}(V), \quad K \mapsto \langle \Gamma, K\rangle$$

is a 1-times-covariant tensor, i.e. we identify $\widehat{\mathcal{L}}(V)$ as a subset of $\mathcal{T}_{(1,0)}(V)$. For $K(u) \in \mathcal{L}(V)$ the map

$$T_2 : \widehat{\mathcal{L}}(V) \to \mathcal{F}(V), \quad \Gamma \mapsto \langle \Gamma, K\rangle$$

is a 1-times-contravariant tensor, i.e. we identify $\mathcal{L}(V)$ as a subset of $\mathcal{T}_{(0,1)}(V)$.

We summarize some basic definitions and properties of *Lie derivatives* L_K with respect to a vector field $K \in \mathcal{L}(V)$:

- If $f(u) \in \mathcal{F}(V)$, then the Lie derivative of $f(u)$ with respect to $K(u)$ is defined as
$$L_K f(u) := \langle \operatorname{grad} f(u), K(u)\rangle,$$
where by $\operatorname{grad} f(u)$ we denote the *gradient* of f with respect to u, i.e. the covector field
$$\begin{pmatrix} \frac{\partial}{\partial u_0} f(u) & \frac{\partial}{\partial u_1} f(u) & \cdots & \frac{\partial}{\partial u_1} f(u) \end{pmatrix},$$
if $u_0, u_1, \ldots, u_n$ are the $n+1$ components of u.

- If $G(u) \in \mathcal{L}(V)$, then the Lie derivative of $G(u)$ with respect to $K(u)$ is defined as
$$L_K G(u) := [\![K(u), G(u)]\!],$$
where $[\![K(u), G(u)]\!]$ again denotes the commutator of $K(u)$ and $G(u)$ as defined in (1.3.3).

- For $\Gamma(u) \in \widehat{\mathcal{L}}(V)$ and $G(u) \in \mathcal{L}(V)$, the Lie derivative of $\Gamma(u)$ with respect to $K(u)$ is defined by
$$\langle L_K \Gamma(u), G(u)\rangle := L_K(\langle \Gamma(u), G(u)\rangle) - \langle \Gamma(u), L_K G(u)\rangle.$$

[9] The definitions are nearly the same, when considering a general manifold M, which is in general not a vector space. For our purposes it is sufficient, to consider the more special situation of vector spaces.

We assume that in the above stated results and the following course of this section all necessary differentiations can be performed without explicitly pointing this out.

For arbitrary n-times-covariant-m-times-contravariant tensors $T \in \mathcal{T}_{(n,m)}(V)$, $K_1, \ldots, K_n \in \mathcal{L}(V)$ and $\Gamma_1, \ldots, \Gamma_m \in \widehat{\mathcal{L}}(V)$, the Lie derivative $L_K T$ of T with respect to $K \in \mathcal{L}(V)$ is the linear map $L_K : \mathcal{T}_{(n,m)}(V) \to \mathcal{T}_{(n,m)}(V)$ defined by

$$\begin{aligned}(L_K T)(K_1, \ldots, K_n, \Gamma_1, \ldots, \Gamma_m) = {} & L_K(T(K_1, \ldots, K_n, \Gamma_1, \ldots, \Gamma_m)) - \\ & \sum_{i=1}^{n} T(K_1, \ldots, K_{i-1}, L_K K_i, K_{i+1}, \ldots, K_n, \Gamma_1, \ldots, \Gamma_m) - \\ & \sum_{j=1}^{m} T(K_1, \ldots, K_n, \Gamma_1, \ldots, \Gamma_{j-1}, L_K \Gamma_j, \Gamma_{j+1}, \ldots, \Gamma_m).\end{aligned}$$

In the following we will not consider Lie derivatives on arbitrary tensors. We simply need the above definition to present the general definition of the exterior derivative.

Note that the product rule holds for Lie derivatives, i.e.

$$L_K(\langle \Gamma(u), G(u) \rangle = \langle L_K \Gamma(u), G(u) \rangle + \langle \Gamma(u), L_K G(u) \rangle$$

for all $K(u), G(u) \in \mathcal{L}(V)$ and $\Gamma(u) \in \widehat{\mathcal{L}}(V)$. This is a direct consequence of the definition of the Lie derivative of a covector field.

In the following the application of arbitrary tensors to vector fields and covector fields, respectively, is denoted as multiplication. The *exterior derivative* d is the linear map

$$d : \mathcal{T}_{(n,0)}(V) \to \mathcal{T}_{(n+1,0)}(V),$$

recursively defined by

$$\begin{aligned} d(0) &:= 0, \\ \langle df(u), K(u) \rangle &:= L_K f(u), \\ (dT) \cdot K(u) &:= L_K(T) - d(T \cdot K(u)) \end{aligned}$$

for all $f(u) \in \mathcal{F}(V)$, $K(u) \in \mathcal{L}(V)$ and $T \in \mathcal{T}_{(n,0)}(V)$, $n \in \mathbb{N}$.

We will only consider exterior derivatives of scalar fields and covector fields. Note that for $f(u) \in \mathcal{F}(V)$ and $K(u) \in \mathcal{L}(V)$ we have

$$\langle df(u), K(u)\rangle = \langle \operatorname{grad} f(u), K(u)\rangle.$$

For a covector field $\Gamma(u) \in \widehat{\mathcal{L}}(V)$ and $K(u) \in \mathcal{L}(V)$ we obtain

$$(d\Gamma) \cdot K(u) = L_K\Gamma(u) - d(\langle \Gamma(u), K(u)\rangle).$$

Remark 1.24. (Existence of potentials) Let $\Gamma(u) \in \widehat{\mathcal{L}}(V)$ be a closed covector field, i.e. $d\Gamma(u) = 0$. Then at least locally there is a scalar field $s(u) \in \mathcal{F}(V)$, such that

$$ds(u) = \Gamma(u).$$

Such a scalar field $s(u)$ will also be referred to as a *potential* of the covector field $\Gamma(u)$. This result is known as the famous Poincaré Lemma (see also [51], Section 1.5 on differential forms)[10].

1.5.2 An integrability result

We use the same notation as in the preceding part of this section, i.e. the manifold is an $(n+1)$–dimensional vector space V.

Consider the evolution equation

$$u_t = K_0(u), \tag{1.5.1}$$

$u = u(t)$, $t \in \mathbb{R}$, and K_0 a vector field in u. Let $K_1(u), \dots, K_n(u)$ be symmetry generators for (1.5.1), i.e.

$$[\![K_0(u), K_i(u)]\!] = 0$$

for all $1 \le i \le n$. Furthermore we assume that

- $\{K_0(u), K_1(u), \dots, K_n(u)\} \subseteq V$ is set of linear independent vector fields for each $u = u(t)$, $t \in \mathbb{R}$

and

- $[\![K_i(u), K_j(u)]\!] = 0$ for all $1 \le i, j \le n$.

[10] The underlying setting in Section 1.5 on differential forms of [51] is much more general than we will need here. For our purposes the closed covector fields under consideration will simply be gradients of scalar functions.

We fix a point $u(0) \in V$ and parameterize the line between $u(0)$ and some arbitrary point $u \in V$ via

$$U(\lambda, u) = u(0) + \lambda(u - u(0)) \tag{1.5.2}$$

for $0 \leq \lambda \leq 1$. Since $\{K_0(u), K_1(u), \ldots, K_n(u)\} \subseteq V$ is linear independent (i.e. a basis of V), we find scalar coefficients $\beta_i(\lambda, u)$, $0 \leq i \leq n$, such that

$$u - u(0) = \sum_{i=0}^{n} \beta_i(\lambda, u)\, K_i(U(\lambda, u)). \tag{1.5.3}$$

Define

$$s_j(u) = \int_0^1 \beta_j(\lambda, u)\, d\lambda \tag{1.5.4}$$

for all $0 \leq j \leq n$.

Theorem 1.25. (B. Fuchssteiner, 2005) *Under the above assumptions we define covector fields $\Gamma_0(u), \ldots, \Gamma_n(u)$, such that*

$$\langle \Gamma_i(u), K_j(u) \rangle = \delta_{ij}$$

for all $0 \leq i, j \leq n$, where by δ_{ij} we denote the usual Kronecker symbol, i.e. $\delta_{ij} = 1$ when $i = j$ and $\delta_{ij} = 0$ otherwise. Then:

(i) $L_{K_j}(\Gamma_i) = 0$ *for all $0 \leq i, j \leq n$.*

(ii) Γ_i *is a closed covector field for all $0 \leq i \leq n$.*

Proof. Note that since $\{K_0(u), \ldots, K_n(u)\}$ forms a basis of V, it is sufficient to characterize the covector fields $\Gamma_0(u), \ldots, \Gamma_n(u)$ by their application to $K_0(u), \ldots, K_n(u)$. The rest follows by taking linear combinations.

(i) We have $L_{K_j}(\langle \Gamma_i, K_s \rangle) = L_{K_j}(\delta_{is}) = 0$ for all $0 \leq i, j, s \leq n$. Furthermore $L_{K_j}(K_s) = [\![K_j, K_s]\!] = 0$. By the product rule for Lie derivatives, we find

$$0 = L_{K_j}(\langle \Gamma_i, K_s \rangle) = \langle L_{K_j}(\Gamma_i), K_s \rangle + \underbrace{\langle \Gamma_i, L_{K_j}(K_s) \rangle}_{=0},$$

i.e. $\langle L_{K_j}(\Gamma_i), K_s \rangle = 0$. Since $\{K_0(u), K_1(u), \ldots, K_n(u)\} \subseteq V$ is a basis of V and $\langle L_{K_j}(\Gamma_i), K_s \rangle = 0$ holds for all $0 \leq s \leq n$, it follows $L_{K_j}(\Gamma_i) = 0$. This proves the first assertion of the theorem.

(ii) Since

$$(d\Gamma_i) \cdot K_s = \underbrace{L_{K_s}(\Gamma_i)}_{=0} - d(\underbrace{\langle \Gamma_i, K_s \rangle}_{\delta_{is}}),$$

for all $0 \leq s \leq n$, it follows $d(\Gamma_i) = 0$, i.e. Γ_i is a closed vector field for all $0 \leq i \leq n$. □

The central result of this section is the following theorem, which shows how to characterize the solutions of any evolution equation of the form (1.5.1) under the assumption that we have n commuting and linear independent symmetry generators.

Theorem 1.26. (B. Fuchssteiner, 2005) *The solutions of the evolution equation (1.5.1) are then characterized in implicit form by*

$$\begin{pmatrix} s_0(u(t)) \\ s_1(u(t)) \\ \vdots \\ s_n(u(t)) \end{pmatrix} = \begin{pmatrix} s_0(u(0)) + t \\ s_1(u(0)) \\ \vdots \\ s_n(u(0)) \end{pmatrix}, \tag{1.5.5}$$

where the $s_i(u(t))$, $0 \leq i \leq n$, are defined as in (1.5.4).

Proof. Let $\Gamma_0(u), \ldots, \Gamma_n(u)$ be the covector fields as defined in Theorem 1.25. For $j \in \{0, \ldots, n\}$ it follows from (1.5.3) and the definition of the covector field Γ_j that

$$\begin{aligned} \langle \Gamma_j(U(\lambda, u)), u - u(0) \rangle &= \left\langle \Gamma_j(U(\lambda, u)), \sum_{i=0}^{n} \beta_i(\lambda, u)\, K_i(U(\lambda, u)) \right\rangle \\ &= \sum_{i=1}^{n} \beta_i(\lambda, u) \cdot \underbrace{\langle \Gamma_j(U(\lambda, u)), K_i(U(\lambda, u)) \rangle}_{=\delta_{ij}} \\ &= \beta_j(\lambda, u). \end{aligned}$$

Hence, by (1.5.4) we obtain

$$\begin{aligned} s_j(u) &= \int_0^1 \beta_j(\lambda, u)\, d\lambda \\ &= \int_0^1 \langle \Gamma_j(U(\lambda, u)), u - u(0) \rangle\, d\lambda \\ &= \int_0^1 \langle \Gamma_j(u(0) + \lambda \cdot (u - u(0))), u - u(0) \rangle\, d\lambda, \end{aligned}$$

i.e. $s_j(u)$ is a potential of $\Gamma_j(u)$ and $ds_j(u) = \Gamma_j(u)$. It follows :

$$\frac{d}{dt}s_j(u) = \langle ds_j(u), u_t\rangle = \langle \Gamma_j(u), K_0(u)\rangle = \delta_{0j},$$

i.e.

$$\frac{d}{dt}s_0(u(t)) = 1 \quad \text{and} \quad \frac{d}{dt}s_i(u(t)) = 0,$$

for $1 \leq i \leq n$. Here we have used (1.5.1) to replace u_t by $K_0(u)$. This completes the proof of the theorem. □

Remark 1.27. This theorem should not be mixed up with Lie's celebrated result (see e.g. [64] p. 86 or [51], Theorem 2.64, p. 155). Lie's result deals with the case of Lie point symmetries and the present result with flows in n-dimensional space, which corresponds — if ODEs of higher order are considered — to the case of Lie–Bäcklund symmetries. The difference to Lie's theorem becomes clear by the fact that generators for Lie point are two dimensional flows[11].

In Remark 1.19 on page 33 we stated that we can always write any ODE in the scalar case in phase space notation, i.e. we can view any such ODE as an evolution equation of the form (1.5.1).

Hence, Theorem 1.26 is applicable to those types of differential equations of the form (1.1.9), which we discussed in Section 1.1 and Section 1.2 and, especially, in the context of Lie point symmetries. The methods of Cheb-Terrab et. al. can be transferred to the more general setting of Theorems 1.25 and 1.26 as we will see in the following part of this chapter. Further examples and applications making use of Theorem 1.26 are discussed in the next subsection.

Remark 1.28. In the proof of Theorem 1.26 we saw that the functions $s_i(u)$, $0 \leq i \leq n$, as defined in (1.5.4) can be obtained as the potentials of the covector fields $\Gamma_i(u)$, $0 \leq i \leq n$, as defined in Theorem 1.25. When applying the results of Theorem 1.25 and Theorem 1.26 to ODEs, where sufficiently many independent commuting symmetry generators are known, we practically proceed as follows: in a first step we compute the phase space representation of the ODE and the symmetry generators providing the vector fields $K_i(u)$, $0 \leq i \leq n$,

[11] A generalization of the results stated in Theorem 1.25 and Theorem 1.26 combining symmetries and conserved quantities of an evolution equation is sketched in the framework of Section 3.6.4 on page 181 at the end of Chapter 3, where we discuss open problems and perspectives.

where $K_0(u) = K(u)$ corresponds to the right–hand–side of the phase space representation of the ODE to be solved. We then create the matrix

$$A_{[K_0,\ldots,K_n]} = \begin{pmatrix} K_0(u) & \cdots & K_n(u) \end{pmatrix}, \tag{1.5.6}$$

whose i-th column consists of the vector field $K_i(u)$ for $0 \leq i \leq n$. Then we compute the inverse $A^{-1}_{[K_0,\ldots,K_n]}$ of $A_{[K_0,\ldots,K_n]}$ and obtain

$$A^{-1}_{[K_0,\ldots,K_n]} = \begin{pmatrix} \Gamma_0(u) \\ \vdots \\ \Gamma_n(u) \end{pmatrix},$$

i.e. the i-th row of the matrix $A^{-1}_{[K_0,\ldots,K_n]}$ gives the covector field $\Gamma_i(u)$ for $0 \leq i \leq n$. Since $\Gamma_i(u)$ is closed, we can compute the potential $s_i(u)$ of $\Gamma_i(u)$ for $0 \leq i \leq n$.

The solutions of the ODE under consideration are then characterized by the conserved quantities

$$\begin{aligned} s_0(u) &= c_0 + t, \\ s_1(u) &= c_1, \\ &\vdots \\ s_n(u) &= c_n, \end{aligned}$$

where $c_0, c_1, \ldots, c_n$ are constants of integration, which are determined by suitable initial values for the ODE under consideration. By conserved quantities we mean that $s_i(u)$, $1 \leq i \leq n$, does not change along the orbits of (1.5.1), which is the case if and only if $L_K s_i(u) = 0$ for $1 \leq i \leq n$ (i.e. the application of the gradient of $s_i(u)$, $1 \leq i \leq n$, to the vector field $K(u)$ provides 0). In the case of $s_0(u)$, we obtain $L_K s_0(u) = 1$.

The scalar quantities $s_i(u)$, $0 \leq i \leq n$, can be viewed as new coordinates, in which the original flow given by (1.5.1) becomes linear in t with respect to the first coordinate[12] $s_0(u)$ and constant with respect to all other coordinates $s_i(u)$, $1 \leq i \leq n$. From a more general viewpoint, the set of all u satisfying $s_0(u) = c_0 + t, s_1(u) = c_1, \ldots, s_n(u) = c_n$ contribute an invariant manifold associated with (1.5.1). See also Chapter I of [21] for details.

[12] A coordinate with this property is also known as an action angle variable.

Note that to invert $A_{[K_0,\ldots,K_n]}$ means to compute the inverse of a matrix with symbolic coefficients. Using standard Gaussian elimination to compute $A^{-1}_{[K_0,\ldots,K_n]}$ may not be efficient enough for larger values of n, since the performance of standard Gaussian elimination on symbolic matrices suffers from an increasing expression swell. This expression swell arises from the division by pivot elements during the elimination process. Costly normalization strategies have to be applied to the arising matrix components on the one hand to keep the components as small and simple as possible and on the other hand to be able to decide, which elements serve as pivot elements in the elimination process (i.e. one has to decide, whether a complex symbolic expression simplifies to zero or not).

Hence, in a practical implementation of the above proposed strategy it is essential to use specialized algorithms for the inversion of symbolic matrices to keep up efficiency. Such an algorithm, which works fine in practice and in the examples we considered, is proposed by Sasaki and Murao in [55][13].

We will consider some examples at the end of the next subsection, where we perform such computations for concrete ODEs.

1.5.3 Examples and applications

We now discuss some applications of Theorem 1.25 and Theorem 1.26. The idea of this section is on the one hand to show, how well–known methods for solving ODEs, such as separation of variables for first order ODEs and Lie symmetry methods, can be obtained and interpreted in the theoretical framework of Theorem 1.25 and Theorem 1.26. On the other hand we will discuss the application of Theorem 1.26 to systems of ODEs with constant coefficients to see, which kind of solution formulas can be deduced using the new symmetry methods. Furthermore we consider a combination of the methods by Cheb-Terrab et. al. used in [7], [9], [10], [13] and the results of Theorem 1.26 for the case of second order linear ODEs.

In the first example we show that Theorem 1.25 and Theorem 1.26 applied in the case of first order ODEs of the form $y'(x) = f(y(x))$ provide the well–known formula for separation of variables.

[13] An implementation of these methods is already available in the standard distribution of the MuPAD library "linalg" for linear algebra computations.

Example 1.29. (Separation of variables) Consider the ODE

$$y'(x) = f(y(x)) \tag{1.5.7}$$

for some given function f in its argument. When looking at standard text books about ODEs (see e.g. [66]), formally the solution of such an ODE is computed via the method of separation of variables, i.e. we write

$$\frac{dy}{dx} = y'(x) = f(y)$$

and then obtain the solution in implicit form as

$$\int_{y(0)}^{y} \frac{1}{f(z)}\, dz = x + c,$$

where c denotes some constant of integration and $y(0)$ some suitable initial value to be considered.

Now we consider the problem of solving (1.5.7) with the help of Theorem 1.26. Since the phase space for the ODE (or vector space as considered in Theorem 1.26) is 1-dimensional, we are in the case $n = 0$, i.e. in the notation of the preceding subsection we do not need any further symmetry generator to characterize the solutions of $y'(x) = f(y(x))$ with the help of Theorem 1.26.

It follows from (1.5.3) that

$$y - y(0) = \beta(\lambda, y) f(y(0) + \lambda(y - y(0))),$$

and, hence,

$$\beta(\lambda, y) = \frac{y - y(0)}{f(y(0) + \lambda(y - y(0)))}.$$

Thus, by (1.5.4) we obtain

$$s(y) = \int_0^1 \frac{y - y(0)}{f(y(0) + \lambda(y - y(0)))}\, d\lambda = \int_{y(0)}^{y} \frac{1}{f(z)}\, dz$$

by using the apparent substitution $z := y(0) + \lambda(y - y(0))$. By Theorem 1.26, the solutions of $y'(x) = f(y(x))$ are implicitly given by $s(y) = s(y(0)) + x$, which directly corresponds to the representation of the solutions obtained from the method of separation of variables. $\diamond$

In Section 1.4 on page 34 we discussed, how to transform the generators of Lie point symmetries to symmetry generators in phase space notation. In the following example we see that in the case of a first order ODE and a given Lie point symmetry, application of Theorem 1.26 provides the same solution formula as stated in (1.1.20) on page 27.

Especially, this implies that the results to solve ODEs via symmetry methods already established by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13] can directly be transferred to the more general setting of Theorem 1.26.

Example 1.30. (Lie point symmetries of first order ODEs) We consider a general first order ODE of the form

$$y'(x) = \Phi(x, y(x)) \tag{1.5.8}$$

and assume that (1.5.8) admits a Lie point symmetry with symmetry generator

$$X = \xi(x,y)\,\frac{\partial}{\partial x} + \eta(x,y)\,\frac{\partial}{\partial y}. \tag{1.5.9}$$

In phase space notation as introduced in Remark 1.19 on page 33 we have to consider

$$u = \begin{pmatrix} y \\ x \end{pmatrix}$$

i.e. the ODE (1.5.8) corresponds to the evolution equation

$$u_t = \underbrace{\begin{pmatrix} \Phi(u) \\ 1 \end{pmatrix}}_{=K_0(u)}.$$

In Section 1.4 on page 34 we stated how to transform the generator of a Lie point symmetry to phase space. In our situation, this provides the vector field

$$K_1(u) = \begin{pmatrix} \eta(u) - \xi(u)\cdot\Phi(u) \\ 0 \end{pmatrix} \tag{1.5.10}$$

corresponding to the symmetry generator (1.5.9). To compute the potentials of the coefficients $\beta_1(\lambda, u)$ and $\beta_2(\lambda, u)$ in the linear combination of $K_0(U(\lambda, u))$ and $K_1(U(\lambda, u))$ as stated in (1.5.3) on page 41, we choose a straight line as parameterization between $u(0)$ and u, i.e.

$$U(\lambda, u) = u(0) + \lambda(u - u(0)) = \begin{pmatrix} y(0) + \lambda(y - y(0)) \\ \lambda x \end{pmatrix}.$$

Then we obtain

$$K_0(U(\lambda,u)) = \begin{pmatrix} \Phi(U(\lambda,u)) \\ 1 \end{pmatrix},$$
$$K_1(U(\lambda,u)) = \begin{pmatrix} \eta(U(\lambda,u)) - \xi(U(\lambda,u))\Phi(U(\lambda,u)) \\ 0 \end{pmatrix}.$$

We have to consider the system of linear equations

$$u - u(0) = \beta_0(U(\lambda,u))\, K_0(U(\lambda,u)) + \beta_1(U(\lambda,u))\, K_1(U(\lambda,u)),$$

whose solution is given by

$$\beta_0 = x,$$
$$\beta_1 = \frac{y - y(0) - x\cdot\Phi(U(\lambda,u))}{\eta(U(\lambda,u)) - \xi(U(\lambda,u))\Phi(U(\lambda,u))}.$$

Hence, it follows

$$s_0(u) = \int_0^1 x\, d\lambda = x + c_0,$$
$$s_1(u) = \int_0^1 \frac{y - y(0) - x\cdot\Phi(U(\lambda,u))}{\eta(U(\lambda,u)) - \xi(U(\lambda,u))\Phi(U(\lambda,u))}\, d\lambda,$$

where c_0 denotes a constant of integration.

We may write the integral defining $s_1(u)$ in the form

$$s_1(u) = \int_0^1 \left\langle \begin{pmatrix} \frac{1}{\eta(U(\lambda,u))-\xi(U(\lambda,u))\Phi(U(\lambda,u))} \\ -\frac{\Phi(U(\lambda,u))}{\eta(U(\lambda,u))-\xi(U(\lambda,u))\Phi(U(\lambda,u))} \end{pmatrix}, \begin{pmatrix} y - y(0) \\ x \end{pmatrix} \right\rangle d\lambda$$

where by $\left\langle \begin{pmatrix} \frac{1}{\eta(U(\lambda,u))-\xi(U(\lambda,u))\Phi(U(\lambda,u))} \\ -\frac{\Phi(U(\lambda,u))}{\eta(U(\lambda,u))-\xi(U(\lambda,u))\Phi(U(\lambda,u))} \end{pmatrix}, \begin{pmatrix} y - y(0) \\ x \end{pmatrix} \right\rangle$ we mean the usual standard scalar product of the two vectors. Form the last representation of $s_1(u)$ we already see that this integral corresponds to the line integral appearing in the solution formula for first order ODEs (1.1.20) on page 27.

Due to Theorem 1.26, the solutions of (1.5.8) are characterized in implicit form via

$$x + c_0 = s_0(u(0)) + t,$$
$$\int_0^1 \frac{y - y(0) - x\cdot\Phi(U(\lambda,u))}{\eta(U(\lambda,u)) - \xi(U(\lambda,u))\Phi(U(\lambda,u))}\, d\lambda = s_1(u(0)).$$

The first equation is a consequence of converting the scalar ODE (1.5.8) to phase space and introducing a new independent variable t. It simply states that x and t differ up to a constant, which is clear, since from the representation of (1.5.8) in phase space we read–off that the derivative of x with respect to t is 1. This fact is simply mirrored by the first equation including s_0.

The second equation involving s_1 is exactly the solution formula for a first order ODE of the form (1.5.8) admitting the Lie point symmetry (1.5.9) as stated in (1.1.20) on page 27. $\diamond$

In the next two examples we apply the ideas provided by Theorem 1.26 to systems of differential equations to find implicit characterizations for their solutions with the help of commuting symmetry generators.

Example 1.31. (Systems of ODEs with constant coefficients) Let $M_1, \ldots, M_n \in \mathbb{R}^{n\times n}$ be $(n \times n)$-matrices with real components such that

$$[\![M_i, M_j]\!] = M_j\, M_i - M_i\, M_j = 0$$

for all $1 \leq i, j \leq n$. Consider as manifold M the n-dimensional vector space, where these matrices act, and here in particular the vector fields

$$\vec{u} \to K_k(\vec{u}) := M_k\, \vec{u}$$

for $1 \leq k \leq n$, where

$$\vec{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$$

denotes the manifold variable. Clearly these vector fields commute, since the matrices do commute. Now we form a matrix as in (1.5.6) and call an element $\vec{u} \in M$ *generic*, if this matrix

$$\begin{aligned} A_{\vec{u}} &= \begin{pmatrix} K_1(\vec{u}) & \cdots & K_n(\vec{u}) \end{pmatrix} \\ &= \begin{pmatrix} M_1\, \vec{u} & \cdots & M_n\, \vec{u} \end{pmatrix} \end{aligned}$$

has rank n, i.e. is an invertible matrix. At generic $\vec{u}$ the vector fields $K_1(\vec{u}), \ldots, K_n(\vec{u})$ are linear independent.

For some arbitrary but fixed $i \in \{1, \ldots, n\}$ we consider a system of differential equations

$$\vec{u}(t)_t = M_i\, \vec{u}(t), \tag{1.5.11}$$

with n-dimensional phase space. This system we want to solve for a generic initial condition[14] $\vec{u}(0)$. One should observe that when $\vec{u}(0)$ is generic, then all $\vec{u}(t)$ are generic. This is true, because the matrix

$$A_{\vec{u}(t)} = \begin{pmatrix} M_1\,\vec{u} & \cdots & M_n\,\vec{u} \end{pmatrix}$$

is obtained from

$$A_{\vec{u}(0)} = \begin{pmatrix} M_1\,\vec{u}(0) & \cdots & M_n\,\vec{u}(0) \end{pmatrix}$$

by application of the exponential of M_i, clearly an invertible matrix.

Now we consider the quantities introduced in (1.5.3) and (1.5.4) (see page 41). By this we obtain for $\lambda \in [0,1]$ with regard to the fixed $\vec{u}(0)$

$$\vec{\beta}(\lambda,\vec{u}) = (A_{\vec{u}(0)} + \lambda(A_{\vec{u}-\vec{u}(0)}))^{-1}(\vec{u} - \vec{u}(0)), \tag{1.5.12}$$

where we put the different β_j as entries into a vector $\vec{\beta}$. In the same way we can put the s_j as entries into a vector[15]

$$\vec{s}(\vec{u}) = \int_0^1 \vec{\beta}(\lambda,\vec{u})\,d\lambda. \tag{1.5.13}$$

Here one should observe that there is a connected open set Ω in the vector space under consideration, which contains $\vec{u}(0)$ and where for $\vec{u}$ in the set Ω the $\vec{\beta}(\lambda,\vec{u})$ are never singular[16]. We then consider the components of $\vec{s}(\vec{u})$ as new coordinates on the connected open set Ω and find that the orbit of (1.5.11) in these new coordinates is linear in the i-th component and zero in the others:

$$\vec{s}(\vec{u}(t)) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ t \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{1.5.14}$$

[14] In a while we will see what to do in important cases for non–generic initial conditions.

[15] This means that we take the line integral from $\vec{u}(0)$ to $\vec{u}$ along the straight line connecting them. Since for this quantity line integrals are path independent, we could take any other path connecting these two points, thus avoiding eventual singularities on that line. However, for getting simple and compact formulas and for demonstrating the method, we have chosen, this is the most simple path.

[16] This is a consequence of the closedness of the spectrum of a continuous linear operator.

If instead of (1.5.11) the equation with another M_j is chosen, then only the component linear in t is the j-th one instead of the i-th one.

If we take other points in Ω as initial condition for our equations, say with $\vec{s}(\vec{u}(0)) = \vec{s}(0)$, then the solution for (1.5.11) reads

$$\vec{s}(\vec{u}(t)) = \begin{pmatrix} s_1(0) \\ \vdots \\ s_{i-1}(0) \\ t + s_i(0) \\ s_{i+1}(0) \\ \vdots \\ s_n(0) \end{pmatrix}. \tag{1.5.15}$$

The inversion of the symbolic matrix $A_{\vec{u}(0)} + \lambda\,(A_{\vec{u}-\vec{u}(0)})$ as stated in (1.5.12) can be done using special computational methods as discussed in [34] and [2]. For a matrix $A(t)$, whose components depend on the variable t, it is known that $A(t)^{-1}$ can be expanded as a Laurent series at the origin. The main results stated in [2] provide efficient algorithms for the computation of the coefficients of such series.

The assertions of the foregoing results follow from Theorem 1.26, however, we additionally present a direct proof here (see also [33]).

First we state that a formal integration for obtaining the $\vec{s}(\vec{u})$ in different form is given by

$$\vec{s}(\vec{u}) = \ln(A^{-1}_{\vec{u}(0)} A_{\vec{u}})(A_{\vec{u}} - A_{\vec{u}(0)})^{-1}(\vec{u} - \vec{u}(0)). \tag{1.5.16}$$

With the help of this we then show that (1.5.15) gives the solutions of $\vec{u}(t)_t = M_i \vec{u}(t)$ in implicit form. Without loss of generality we may assume $i = 1$. We evaluate on the orbits of $\vec{u}(t)_t = M_1 \vec{u}(t)$ and obtain

$$\begin{aligned}
\vec{\beta}(\lambda, \vec{u(t)}) &= (A_{\vec{u}(0)} + \lambda\,(A_{\vec{u}(t)-\vec{u}(0)}))^{-1}\,(\vec{u}(t) - \vec{u}(0)) \\
&= (\mathbb{1} + \lambda\,A^{-1}_{\vec{u}(0)} A_{\vec{u}(t)-\vec{u}(0)})^{-1}\,A^{-1}_{\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)) \\
&= (\mathbb{1} - \lambda\,(A^{-1}_{\vec{u}(0)} A_{\vec{u}(t)-\vec{u}(0)}) + \lambda^2\,(A^{-1}_{\vec{u}(0)} A_{\vec{u}(t)-\vec{u}(0)})^2 - \ldots) \\
&\quad\; A^{-1}_{\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)).
\end{aligned}$$

Hence, integration with respect to λ and the usual series expansions of the function ln provide:

$$\begin{aligned}
\vec{s}(\vec{u}(t)) &= \int_0^1 \vec{\beta}(\lambda, u(\vec{t}))\, d\lambda \\
&= \Big(1\!\!1 - \frac{1}{2}\,(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)}) + \frac{1}{3}\,(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)})^2 - \dots\Big) \\
&\quad A^{-1}_{\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)) \\
&= \Big((A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)}) - \frac{1}{2}\,(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)})^2 + \frac{1}{3}\,(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)})^3 - \dots\Big) \\
&\quad (A^{-1}_{\vec{u}(t)-\vec{u}(0)}\, A_{\vec{u}(0)})\, A^{-1}_{\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)) \\
&= \ln(1\!\!1 + A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)-\vec{u}(0)})\, A^{-1}_{\vec{u}(t)-\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)) \\
&= \ln(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)})\, A^{-1}_{\vec{u}(t)-\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)),
\end{aligned}$$

which is the representation of $\vec{s}(\vec{u}(t))$ as claimed in (1.5.16).

Next we have to see that (1.5.15) gives the solutions of $\vec{u}(t)_t = M_1\vec{u}(t)$ in implicit form. Since $\vec{u}(t)_t = M_1\vec{u}(t)$ is a system with constant coefficients, the explicit solution can always be formally written as $\vec{u}(t) = e^{tM_1}\vec{u}(0)$, where e^{tM_1} denotes the exponential of the matrix tM_1 in the usual way (see e.g. [66]).

This provides

$$\begin{aligned}
\vec{s}(\vec{u}(t)) &= \ln(A^{-1}_{\vec{u}(0)}\, A_{\vec{u}(t)})\, A^{-1}_{\vec{u}(t)-\vec{u}(0)}\,(\vec{u}(t) - \vec{u}(0)) \\
&= \ln\Big(A^{-1}_{\vec{u}(0)}\,\big(\; M_1\, e^{tM1}\,\vec{u}(0) \quad \cdots \quad M_n\, e^{tM1}\,\vec{u}(0)\;\big)\Big) \\
&\quad \Big(\big(\; M_1\, e^{tM1}\,\vec{u}(0) \quad \cdots \quad M_n\, e^{tM1}\,\vec{u}(0)\;\big) - A_{\vec{u}(0)}\Big)^{-1}\,(e^{tM_1}\vec{u}(0) - \vec{u}(0)) \\
&= \ln(A^{-1}_{\vec{u}(0)}\, e^{tM_1}\, A_{\vec{u}(0)})\,(e^{tM_1}\, A_{\vec{u}(0)} - A_{\vec{u}(0)})^{-1}(e^{tM_1}\,\vec{u}(0) - \vec{u}(0)) \\
&= A^{-1}_{\vec{u}(0)}\, \ln(e^{tM_1})\, A_{\vec{u}(0)}\, A^{-1}_{\vec{u}(0)}\,(e^{tM_1} - 1\!\!1)^{-1}\,(e^{tM_1} - 1\!\!1))\,\vec{u}(0) \\
&= A^{-1}_{\vec{u}(0)}\, t\, M_1\,\vec{u}(0) \\
&= t\,\vec{e}_1,
\end{aligned}$$

where $\vec{e}_1$ denotes the transposed of the vector $(1, 0, \dots, 0)$ and e^{tM_1} and M_i commute, since M_1 and M_i commute, $1 \le i \le n$. This proves the assertion. $\diamond$

Example 1.32. (The obvious applications of the last example) One should observe that the usual solution function for a system with constant coefficients is not suitable for determining explicit symbolic solutions algorithmically. The reason is that for the explicit representation of the exponential of a matrix the zeroes of its characteristic polynomial have to be determined, which may not be possible for degrees higher than four[17].

However, Example 1.31 shows that explicit formulas may be obtained algorithmically if changes in coordinates and quadratures are taken into consideration. Equations having such solution formulas are said to be integrable, a notion which is well–known in the context of Hamiltonian systems.

The only task, which remains open for showing that a system with constant coefficients

$$\vec{u}(t)_t = M\vec{u}(t)$$

is integrable, is the construction of the matrices, which were needed in Example 1.31. That however is obvious: Take $i = 2$ and

$$M_1 := \mathbb{1},\, M_2 := M^1,\, \ldots\,,\, M_n := M^{n-1},$$

then for generic initial conditions all the foregoing results can be applied. However, what do we do when the solution is looked for an initial value $\vec{u}(0)$, which is non–generic, i.e. for which the matrix

$$\begin{pmatrix} \vec{u}(0) & M\,\vec{u}(0) & M^2\,\vec{u}(0) & \cdots & M^{n-1}\,\vec{u}(0) \end{pmatrix}$$

does not have rank n, i.e. where the columns are not linear dependent?

Well, that is simple: If we have rank $n-k$, then we take the vector space spanned by the basis given by the $n-k$ linear independent columns of the matrix. This vector space then is invariant under M. Thus, when represented with respect to that basis, M becomes an $(n-k)\times(n-k)$ matrix, and the initial value is generic. Hence also the non–generic situation is integrable.

Another but similar class of equations integrable by this method is that of homogeneous linear differential equations with constant coefficients. Consider

$$u^{(n)}(t) + m_{n-1}\,u^{(n-1)}(t) + \ldots + m_1\,u'(t) + m_0\,u(t) = 0, \tag{1.5.17}$$

[17]Even if this is possible in special cases or for polynomial equations of order 3 and 4, the representations of the roots of the characteristic polynomials in exact form may in general lead to awfully complicated expressions.

$m_i \in \mathbb{R}$, $0 \leq i \leq n-1$, $u(t)$ some real valued function. In phase space notation we obtain

$$\underbrace{\begin{pmatrix} u(t) \\ u'(t) \\ \vdots \\ u^{(n-1)}(t) \end{pmatrix}_t}_{\vec{u}(t)_t} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & & 1 \\ -m_0 & -m_1 & -m_2 & \cdots & -m_{n-1} \end{pmatrix}}_{=M} \underbrace{\begin{pmatrix} u(t) \\ u'(t) \\ \vdots \\ u^{(n-1)} \end{pmatrix}}_{=\vec{u}(t)} \tag{1.5.18}$$

and we are back at what we considered a few lines ago. Here we always have a generic situation. $\diamond$

The next example is in a way a negative example. We discuss the application of Theorem 1.26 to find symmetries of second order homogeneous linear differential equations. It turns out that Theorem 1.26 combined with an ansatz similar to the strategies used by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13] to find symmetries of first order ODEs in the situation of second order linear ODEs does not lead to new ways of finding solutions.

Nevertheless, the example is interesting in so far that it demonstrates the well–known link between homogeneous linear ODEs and their associated Riccati equations.

Example 1.33. (Second order homogeneous linear ODEs) We consider second order homogeneous linear ODEs, i.e. ODEs of the form

$$z''(x) = f_1(x)\, z'(x) + f_0(x)\, z(x) \tag{1.5.19}$$

for given functions $f_0(x), f_1(x)$. Note that any second order homogeneous linear ODE can be transformed to the more special form

$$y''(x) = a(x)\, y(x) \tag{1.5.20}$$

using the transformation

$$z(x) = y(x)\, \exp\Big(-\frac{\int f_1(x)dx}{2}\Big).$$

Hence, we restrict our attention to ODEs of the form (1.5.20). To be able to apply Theorem 1.26 to the situation considered here, we work with the phase space representation of (1.5.20), which is

$$\underbrace{\begin{pmatrix} u_0 \\ u_1 \\ x \end{pmatrix}_t}_{=u_t} = \underbrace{\begin{pmatrix} u_1 \\ au_0 \\ 1 \end{pmatrix}}_{=K_0(u)}, \tag{1.5.21}$$

where $u_0 = u_0(x) = y(x)$ and $u_1 = u_1(x) = y'(x)$.

Every second order homogeneous linear ODE written in phase space notation admits the symmetry generator

$$K_1 = \begin{pmatrix} u_0 \\ u_1 \\ 0 \end{pmatrix}, \tag{1.5.22}$$

i.e. $[\![K_0(u), K_1(u)]\!] = 0$ (scaling symmetry). To be able to use Theorem 1.26, we have to find another symmetry generator $K_2(u)$ admitted by (1.5.21), such that $K_0(u)$, $K_1(u)$ and $K_2(u)$ are linear independent and, additionally, the commutator of $K_1(u)$ and $K_2(u)$ vanishes.

In general, if we search for a symmetry generator $K_2(u)$ of the form

$$K_2(u) = \begin{pmatrix} k_1(u_0, u_1, x) \\ k_2(u_0, u_1, x) \\ 0 \end{pmatrix}, \tag{1.5.23}$$

where $k_1(u_0, u_1, x)$ and $k_2(u_0, u_1, x)$ are scalar functions in their arguments, and

$$\left[\!\!\left[\begin{pmatrix} u_1 \\ au_0 \\ 1 \end{pmatrix}, \begin{pmatrix} k_1(u_0, u_1, x) \\ k_2(u_0, u_1, x) \\ 0 \end{pmatrix} \right]\!\!\right] = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\left[\!\!\left[\begin{pmatrix} u_0 \\ u_1 \\ 1 \end{pmatrix}, \begin{pmatrix} k_1(u_0, u_1, x) \\ k_1(u_0, u_1, x) \\ 0 \end{pmatrix} \right]\!\!\right] = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The vanishing of the commutators provides the determining equations

$$k_{1u_0} u_1 + k_{1u_1} au_0 + k_{1x} - k_2 = 0, \tag{1.5.24}$$

$$k_{2u_0} u_1 + k_{2u_1} au_0 + k_{2x} - ak_1 = 0, \tag{1.5.25}$$

$$k_{1u_0} u_0 + k_{1u_1} u_1 - k_1 = 0, \tag{1.5.26}$$

$$k_{2u_0} u_0 + k_{2u_1} u_1 - k_2 = 0, \tag{1.5.27}$$

for the components $k_1(u_0, u_1, x)$ and $k_2(u_0, u_1, x)$ of the symmetry generator $K_2(u)$ (we use indices for the corresponding partial derivatives).

From the last two equations we choose the ansatz $k_1 = f_1(x)u_0 + f_2(x)u_1$ as well as $k_2 = g_1(x)u_0 + g_2(x)u_1$ for some unknown functions $f_1(x)$, $f_2(x)$, $g_1(x)$ and $g_2(x)$. Insertion of these representations for k_1 and k_2 into the first two equations provides

$$f_1u_1 + f_2au_0 + f_{1x}u_0 + f_{2x}u_1 - g_1u_0 - g_2u_1 = 0, \tag{1.5.28}$$

$$g_1u_1 + g_2au_0 + g_{1x}u_0 + g_{2x}u_1 - af_1u_0 - af_2u_1 = 0. \tag{1.5.29}$$

It follows for f_1, f_2, g_1 and g_2 that

$$f_1 + f_{2x} = g_2, \tag{1.5.30}$$

$$af_2 + f_{1x} = g_1, \tag{1.5.31}$$

$$g_1 + g_{2x} = af_2, \tag{1.5.32}$$

$$ag_2 + g_{1x} = af_1. \tag{1.5.33}$$

Inserting equation (1.5.32) for g_1 into equation (1.5.31), we find $g_{2x} = -f_{1x}$, i.e. $g_2 = -f_1 + c_1$, where c_1 is a constant.

Hence, equations (1.5.30), (1.5.31), (1.5.32) and (1.5.33) are equivalent to

$$2f_1 + f_{2x} - c_1 = 0, \tag{1.5.34}$$

$$g_1 - f_{1x} = af_2, \tag{1.5.35}$$

$$g_2 = -f_1 + c_1, \tag{1.5.36}$$

$$a(c_1 - 2f_1) + g_{1x} = 0. \tag{1.5.37}$$

Solving equation (1.5.34) for f_1 and inserting this representation for f_1 into (1.5.35), (1.5.36) and (1.5.37), we obtain

$$f_1 = \frac{c_1}{2} - \frac{f_{2x}}{2}, \tag{1.5.38}$$

$$g_1 = af_2 - \frac{f_{2xx}}{2}, \tag{1.5.39}$$

$$g_2 = \frac{c_1}{2} + \frac{f_{2x}}{2}, \tag{1.5.40}$$

$$af_{2x} + g_{1x} = 0. \tag{1.5.41}$$

Insertion of the total derivative of the right–hand–side of (1.5.39) into (1.5.41) provides $2af_{2x} + a_x f_2 - \frac{1}{2}f_{2xxx} = 0$. This equation we write in the form $(af_2^2 - \frac{1}{2}f_2 f_{2xx} + \frac{1}{4}f_{2x}^2)_x = 0$, from which we finally obtain

$$f_1 = \frac{c_1}{2} - \frac{f_{2x}}{2}, \tag{1.5.42}$$

$$g_1 = \frac{\frac{1}{2}f_2 f_{2xx} - \frac{1}{4}f_{2x}^2 - \frac{1}{2}f_2 f_{2xx} + c_2}{f_2} = \frac{-\frac{1}{4}f_{2x}^2 + c_2}{f_2}, \tag{1.5.43}$$

$$g_2 = \frac{c_1}{2} + \frac{f_{2x}}{2}, \tag{1.5.44}$$

$$a = \frac{\frac{1}{2}f_2 f_{2xx} - \frac{1}{4}f_{2x}^2 + c_2}{f_2^2}. \tag{1.5.45}$$

After Multiplication of (1.5.45) by 4 and setting $c_2 = 0$, it follows

$$4a = \frac{2f_2 f_{2xx} - f_{2x}^2}{f_2^2} \Longleftrightarrow 4a = 2\Big(\frac{f_{2x}}{f_2}\Big)_x + \Big(\frac{f_{2x}}{f_2}\Big)^2. \tag{1.5.46}$$

Setting $z(x) := \frac{f_{2x}}{f_2}$, we can write the right–hand–side of the equivalence (1.5.46) as

$$4a = 2z'(x) + z(x)^2. \tag{1.5.47}$$

It follows that f_2 can be obtained as the solution of the Riccati ODE (1.5.47) as well as afterwards solution of the first order homogeneous linear ODE $f_2'(x) = z(x)f_2(x)$. Since there is no general method to compute closed symbolic solutions of Riccati ODEs of the form (1.5.47), the ansatz under consideration is not successful in general[18].

A possible ansatz to simplify the problem in determining an additional symmetry generator $K_2(u)$ in the sense of the strategies presented in [7], [9], [10], [12] and [13] by Cheb-Terrab et. al. would be to take into account simplifying assumptions concerning the dependencies of the components $k_1(u_0, u_1, x)$ and $k_2(u_0, u_1, x)$ of $K_2(u)$ on u_0, u_1, x. In [10], Cheb-Terrab et. al. inspect first order non–linear ODEs admitting such simpler forms of symmetry generators. We started considering the ansatz $k_1 = f_1(x)u_0 + f_2(x)u_1$ and $k_2 = g_1(x)u_0 + g_2(x)u_1$ for the components of the desired third symmetry generator $K_2(u)$. To avoid

[18] See for example the book [66] for further results on Riccati ODEs and methods to find solutions of such ODEs. In Section 4.1 of [53] the authors discuss special algorithmic approaches to determine at least special solutions — so–called rational solutions — of Riccati ODEs.

having to solve a Riccati ODE of type (1.5.47) to determine f_1, f_2, g_1 and g_2, we take the simplifying assumptions $g_1(x) = 0$ and $f_2(x) = 0$ into account, i.e. $k_1 = k_1(u_0, x)$ and $k_2 = k_2(u_1, x)$.

Under this assumptions, the original determining equations (1.5.30), (1.5.31), (1.5.32) and (1.5.33) for the components of $K_2(u)$ read

$$f_1 = g_2, \tag{1.5.48}$$
$$f_{1x} = 0, \tag{1.5.49}$$
$$g_{2x} = 0, \tag{1.5.50}$$
$$ag_2 = af_1. \tag{1.5.51}$$

But from these equations it follows $f_1 = c_1$, $g_2 = c_2$ for constants c_1, c_2 and from (1.5.51) again $c_1 = c_2$. Finally from $f_1 = c_1$, $g_2 = c_1$, $f_2 = 0$ and $g_1 = 0$ it follows that $K_2(u)$ is a scalar multiple of $K_1(u)$ and, hence, of no use for applying Theorem 1.26 (the vector fields need to be linear independent).

An alternative ansatz could be to assume $f_2(x) = 0$ and $g_2(x) = 0$, i.e. $k_1 = k_1(u_0, x)$ and $k_2 = k_2(u_1, x)$. Under these assumptions we obtain $f_1(x) = 0$ and $g_1(x) = 0$, i.e. $K_2(u)$ is zero and again useless.

Taking into account that $f_1(x) = 0$ and $g_2(x) = 0$, i.e. $k_1 = k_1(u_1, x)$ and $k_2 = k_2(u_0, x)$, the determining equations (1.5.30), (1.5.31), (1.5.32) and (1.5.33) provide

$$f_{2x} = 0, \tag{1.5.52}$$
$$af_2 = g_1, \tag{1.5.53}$$
$$g_1 = af_2, \tag{1.5.54}$$
$$g_{1x} = 0. \tag{1.5.55}$$

Now from equation (1.5.52) it follows $f_2 = c_1$, c_1 a constant, and from equation (1.5.55) we find $g_1 = c_2$ for a constant c_2. Using (1.5.53) or (1.5.54) it follows $a = \frac{c_2}{c_1}$, i.e. the homogeneous linear ODE under consideration has constant coefficients and, hence, belongs to the class of ODEs already treated in Example 1.31.

The restrictions we considered concerning the dependencies of the components of the desired third symmetry generator are too strong. Even in the case, where we assume that only one of the functions $f_1(x)$, $f_2(x)$, $g_1(x)$ and $g_2(x)$ vanishes, we do not get a satisfying result: e.g. if we assume that $f_2(x) = 0$, we find from (1.5.30), (1.5.31), (1.5.32) and (1.5.33) the determining system

$$f_1 = g_2, \tag{1.5.56}$$

$$f_{1x} = g_1, \tag{1.5.57}$$

$$g_1 + g_{2x} = 0, \tag{1.5.58}$$

$$ag_2 + g_{1x} = af_1. \tag{1.5.59}$$

From (1.5.59) we conclude $g_1 = c_1$, c_1 a constant, since $af_1 = ag_2$ by (1.5.56), and, thus, from (1.5.58) that $g_2 = -\int g_1\,dx + c_2 = -c_1x + c_2$ for a constant c_2. From (1.5.57) it follows $f_1 = \int c_1 dx = c_1x + c_3$ for a constant c_3 and, because of (1.5.56), we must have $c_1 = -c_1$, i.e. $c_1 = 0$ and $c_2 = c_3$ and $K_2(u)$ is again a scalar multiple of $K_1(u)$ and, hence, of no use for us.

Drawing a conclusion from these computations, we state that for second order linear ODEs neither the methods by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13] nor a combination of these methods with Theorem 1.26 provide a satisfying result. Apart from the trivial symmetry generators obtained when taking simplifying assumptions on the dependencies of the desired symmetry generator into account, we saw that in general to determine the desired third symmetry can lead to the question of being able to find solutions of a Riccati ODE of type (1.5.47). Although any second order homogeneous linear ODE can be associated with a Riccati ODE of type (1.5.47)[19], this example shows that in the situation of second order homogeneous linear ODEs Theorem 1.26 does not seem to provide new insights into ways of finding solutions. $\diamond$

Now we come back to the case of non–linear ODEs. The following two examples demonstrate the use of Theorem 1.26 in the case of concrete third order non–linear ODEs, where two commuting symmetry generators are known.

Example 1.34. Consider the ODE

$$y'''(x) = \frac{y''(x)^2}{2y'(x)}.$$

This ODE can easily be reduced to a second order ODE by introducing a new function $z(x) = y'(x)$. We do not reduce this ODE, since we directly want to

[19] See for example the book [51], Section 2.6, Theorem 2.66 and Example 2.69.

apply Theorem 1.26 to it[20].

Since the right–hand–side of the ODE does not depend on x explicitly, the phase space for the ODE is of dimension 3 and we do not have to introduce a new independent variable t for x. The phase space representation is given by

$$\underbrace{\begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix}_x}_{=u_x} = \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ \frac{u_2^2}{2u_1} \end{pmatrix}}_{K_0(u)}, \tag{1.5.60}$$

where $u_0 = u_0(x) = y(x)$, $u_1 = u_1(x) = y'(x)$ and $u_2 = u_2(x) = y''(x)$. Equation (1.5.60) admits the two commuting symmetry generators

$$K_1(u) = \begin{pmatrix} \frac{8u_1^3}{u_2^3} \\ \frac{12u_1^2}{u_2^2} \\ \frac{12u_1}{u_2} \end{pmatrix}, \qquad K_2(u) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

i.e. $[\![K_0(u), K_1(u)]\!] = [\![K_0(u), K_2(u)]\!] = [\![K_1(u), K_2(u)]\!] = 0$. We now follow the steps of computations described in Remark 1.28. First we build the matrix

$$A_{[K_0,K_1,K_2]} = \begin{pmatrix} u_1 & \frac{8u_1^3}{u_2^3} & 1 \\ u_2 & \frac{12u_1^2}{u_2^2} & 0 \\ \frac{u_2^2}{2u_1} & \frac{12u_1}{u_2} & 0 \end{pmatrix},$$

whose columns consist of the three commuting vector fields $K_0(u), K_1(u), K_2(u)$, and compute its inverse given by

$$A^{-1}_{[K_0,K_1,K_2]} = \begin{pmatrix} 0 & \frac{2}{u_2} & -\frac{2u_1}{u_2^2} \\ 0 & -\frac{u_2^2}{12u_1^2} & \frac{u_2}{6u_1} \\ 1 & -\frac{4u_1}{3u_2} & \frac{2u_1^2}{3u_2^2} \end{pmatrix}.$$

[20] Although at first glance it seems to be skilful to reduce the order of the ODE, the ODE in the above stated and non–reduced form is a good example to illustrate the results of Theorem 1.26 simply because the expressions arising from the computation of the covector fields $\Gamma_i(u)$ introduced in Theorem 1.25 are not too large to be displayed within the framework of this example. A more complex ODE is treated in the framework of the next example below.

Now the rows of $A^{-1}_{[K_0,K_1,K_2]}$ correspond to the closed covector fields $\Gamma_0(u), \Gamma_1(u), \Gamma_2(u)$ introduced in Theorem 1.25, i.e. by writing these covector fields as row vectors we obtain

$$\begin{aligned}
\Gamma_0(u) &= \begin{pmatrix} 0 & \frac{2}{u_2} & -\frac{2u_1}{u_2^2} \end{pmatrix},\\
\Gamma_1(u) &= \begin{pmatrix} 0 & -\frac{u_2^2}{12u_1^2} & \frac{u_2}{6u_1} \end{pmatrix},\\
\Gamma_2(u) &= \begin{pmatrix} 1 & -\frac{4u_1}{3u_2} & \frac{2u_1^2}{3u_2^2} \end{pmatrix}.
\end{aligned}$$

Finally, computing the potentials $s_0(u), s_1(u), s_2(u)$ of $\Gamma_0(u), \Gamma_1(u), \Gamma_2(u)$, respectively, provides

$$s_0(u) = \frac{2u_1}{u_2}, \qquad s_1(u) = \frac{u_2^2}{12u_1}, \qquad s_2(u) = u_0 - \frac{2u_1^2}{3u_2}.$$

Hence, due to Theorem 1.26, the solutions of the ODE (still in phase space notation) are implicitly characterized by

$$\frac{2u_1}{u_2} = c_0 + x, \qquad \frac{u_2^2}{12u_1} = c_1, \quad u_0 - \frac{2u_1^2}{3u_2} = c_2,$$

for constants of integration c_0, c_1, c_2. Reintroducing the original dependent variable $y(x)$, the equations read

$$\frac{2y'(x)}{y''(x)} = c_0 + x, \qquad \frac{y''(x)^2}{12y'(x)} = c_1, \quad y(x) - \frac{2y'(x)^2}{3y''(x)} = c_2. \tag{1.5.61}$$

One easily verifies

$$\frac{d}{dx}\Big(\frac{2y'(x)}{y''(x)}\Big) = 1, \quad \frac{d}{dx}\Big(\frac{y''(x)^2}{12y'(x)}\Big) = 0, \quad \frac{d}{dx}\Big(y(x) - \frac{2y'(x)^2}{3y''(x)}\Big) = 0$$

by performing the total differentiation with respect to x and afterwards substituting the appearing third derivatives of $y(x)$ by the right–hand–side of the ODE under consideration.

Equations (1.5.61) can be viewed as an algebraic system of equations in the indeterminates $y(x)$, $y'(x)$ and $y''(x)$. Hence, the explicit form of the solution can be obtained via a suitable elimination method for systems of algebraic equations. In this concrete example we obtain the solution

$$y(x) = c_1x^3 + c_0^3c_1 + 3c_0^2c_1x + 3c_0c_1x^2 + c_2,$$

where we used MuPAD to obtain this representation. $\diamond$

To see that Theorem 1.26 indeed provides a powerful theoretical basis for a practical characterization of the solutions of relatively complex ODEs, where sufficiently many commuting symmetry generators are known, let us consider as a last example a more complex third order ODE.

Example 1.35. We consider the third order ODE

$$y'''(x) = \frac{y''(x)(y'(x) + \sqrt{y'(x)^2 - y(x)y''(x)})}{y(x)}. \tag{1.5.62}$$

Since the right–hand–side of the ODE does not explicitly depend on x, we do not have to introduce a new independent variable when considering it in phase space. The phase space representation of the ODE is given by

$$\underbrace{\begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix}_x}_{=u_x} = \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ \frac{u_2(u_1+\sqrt{u_1^2-u_0u_2})}{u_0} \end{pmatrix}}_{K_0(u)}, \tag{1.5.63}$$

where $u_0 = u_0(x) = y(x)$, $u_1 = u_1(x) = y'(x)$ and $u_2 = u_2(x) = y''(x)$. Equation (1.5.63) admits the two commuting symmetry generators

$$K_1(u) = \begin{pmatrix} \frac{16(\sqrt{u_1^2-u_0u_2}-u_1)^4}{u_2^4} \\ -\frac{32(\sqrt{u_1^2-u_0u_2}-u_1)^3}{u_2^3} \\ \frac{48(\sqrt{u_1^2-u_0u_2}-u_1)^2}{u_2^2} \end{pmatrix}, \qquad K_2(u) = \begin{pmatrix} -\frac{2(\sqrt{u_1^2-u_0u_2}-u_1)}{u_2} \\ 1 \\ 0 \end{pmatrix},$$

i.e. $[\![K_0(u), K_1(u)]\!] = [\![K_0(u), K_2(u)]\!] = [\![K_1(u), K_2(u)]\!] = 0$. Again we proceed as described in Remark 1.28. We construct the matrix $A_{[K_0,K_1,K_2]}$, whose columns consist of the three commuting symmetry generators:

$$A_{[K_0,K_1,K_2]} = \begin{pmatrix} u_1 & \frac{16(\sqrt{u_1^2-u_0u_2}-u_1)^4}{u_2^4} & -\frac{2(\sqrt{u_1^2-u_0u_2}-u_1)}{u_2} \\ u_2 & -\frac{32(\sqrt{u_1^2-u_0u_2}-u_1)^3}{u_2^3} & 1 \\ \frac{u_2(u_1+\sqrt{u_1^2-u_0u_2})}{u_0} & \frac{48(\sqrt{u_1^2-u_0u_2}-u_1)^2}{u_2^2} & 0 \end{pmatrix}.$$

Computing the inverse is again no problem at all. It can be done very efficiently even using standard methods like standard Gaussian elimination and no specialized algorithms for dealing with symbolic matrix components. Although the inverse is easy to compute, its components are very large expressions due to the expression swell[21]. For this reason, they are not presented here explicitly.

Proceeding as described in Remark 1.28, the rows of the matrix $A^{-1}_{[K_0,K_1,K_2]}$ correspond to the closed covector fields $\Gamma_0, \Gamma_1, \Gamma_2$ as introduced in Theorem 1.25. Computing the potentials $s_0(u), s_1(u), s_2(u)$ of the closed covector fields introduced in (1.5.4), we obtain an implicit characterization of the solutions of (1.5.62) in form of the system $s_0(u) = c_0 + x$, $s_1(u) = c_1$, $s_2(u) = c_2$, where c_0, c_1, c_2 are constants of integration. Since the expressions for $s_0(u)$, $s_1(u)$ and $s_2(u)$ are very large, we only present the result for $s_1(u)$ as an example. The representation computed for $s_1(u)$ is a fraction $\frac{f(u)}{g(u)}$, whose numerator $f(u)$ is given by

$$
\begin{aligned}
f(u) = \quad & 2\,u_0^3\,u_1\,u_2^5 - 2\,\sqrt{u_1^2 - u_0\,u_2}\,u_0^3\,u_2^5 - 2\,u_0^2\,u_1^3\,u_2^4 + \\
& 2\,\sqrt{u_1^2 - u_0\,u_2}\,u_0^2\,u_1^2\,u_2^4 + 3\,\sqrt{u_1^2 - u_0\,u_2}\,u_0\,u_1^4\,u_2^3 - \\
& 4\,\sqrt{\left(u_1^2 - u_0\,u_2\right)^3}\,u_0\,u_1^2\,u_2^3 + \sqrt{\left(u_1^2 - u_0\,u_2\right)^5}\,u_0\,u_2^3 - \\
& 2\,\sqrt{u_1^2 - u_0\,u_2}\,u_1^6\,u_2^2 + 4\,\sqrt{\left(u_1^2 - u_0\,u_2\right)^3}\,u_1^4\,u_2^2 - \\
& 2\,\sqrt{\left(u_1^2 - u_0\,u_2\right)^5}\,u_1^2\,u_2^2,
\end{aligned}
$$

whereas the denominator $g(u)$ is

$$
\begin{aligned}
g(u) = \quad & 16\,\Big(-12\,u_0^4\,u_1\,u_2^3 + 2\,\sqrt{u_1^2 - u_0\,u_2}\,u_0^4\,u_2^3 + 36\,u_0^3\,u_1^3\,u_2^2 - \\
& 14\,\sqrt{u_1^2 - u_0\,u_2}\,u_0^3\,u_1^2\,u_2^2 - 24\,u_0^2\,u_1^5\,u_2 + \\
& 13\,\sqrt{u_1^2 - u_0\,u_2}\,u_0^2\,u_1^4\,u_2 + 6\,\sqrt{\left(u_1^2 - u_0\,u_2\right)^3}\,u_0^2\,u_1^2\,u_2 + \\
& \sqrt{\left(u_1^2 - u_0\,u_2\right)^5}\,u_0^2\,u_2 + 2\,\sqrt{u_1^2 - u_0\,u_2}\,u_0\,u_1^6 \\
& - 2\,\sqrt{\left(u_1^2 - u_0\,u_2\right)^5}\,u_0\,u_1^2\Big).
\end{aligned}
$$

[21] This is the case even after applying elaborate simplification algorithms to the components of the matrix to decrease the size of the expressions.

For our computations we used MuPAD on the one hand to obtain the above expressions and on the other hand to verify, whether $\frac{g(u)}{f(u)}$ is indeed a conserved quantity of the ODE under consideration. Substituting u_0 by $y(x)$, u_1 by $y'(x)$ and $u_2 = y''(x)$ in $\frac{g(u)}{f(u)}$, computing the total derivative of the arising expression with respect to x and finally reducing this expression modulo the original ODE, i.e. replacing $y'''(x)$ by the right–hand–side of the original ODE, the expression simplifies to 0 as desired. $\diamond$

Of course, in complex examples like the one discussed above, it is not always possible to compute explicit representations for the solutions of the ODE under consideration. In the case, where the solutions are given by a set of polynomial equations in the dependent variable $y(x)$ and its derivatives, even simple elimination strategies or Groebner basis techniques may be used to find explicit formulas for the solutions.

In the case, where the equations of the system characterizing the solutions of an ODE involve roots of polynomial expressions or even more complex functions in $y(x)$ and its derivatives, the chances are not that good to find explicit solutions. Nevertheless, the characterization of solutions is a completely algebraic one and may for some classes of such systems of equations be solved by special algorithms for the computation of symbolic solutions or even by heuristic ansatzes. Imagine, we are only interested in finding certain classes of solutions such as exponential functions or even polynomials. Then the set of equations, which implicitly characterize the solutions, may be helpful to decide, whether such solutions exist, and if so, to compute them.

1.6 Conclusions

1.6.1 Resumé

The main results discussed in this chapter are the applications of Theorem 1.26 by B. Fuchssteiner, which allows the complete integration of ODEs admitting sufficiently many commuting and independent symmetry generators.

We showed how the successful Lie point symmetry methods established by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13] can be embedded into the framework of Theorem 1.26. Furthermore, we discussed applications of Theorem 1.26 for the cases of second order homogeneous linear ODEs and n-th order homogeneous linear ODEs with constant coefficients.

The Examples 1.34 and 1.35 illustrated that the application of Theorem 1.26 provides an implicit characterization of the solutions of the ODEs under consideration in terms of conserved quantities. This is an alternative way of representing the solutions of an ODE in implicit form, which allows to draw further conclusions concerning the structure of solutions, for which in most cases even no explicit representation can be computed without introducing new classes of special functions[22].

1.6.2 Open problems and perspectives

1. Using the result provided by Theorem 1.26 in the framework of a computer algebra system to solve ODEs requires methods for determining sufficiently many independent and commuting symmetry generators. A possible way to determine such symmetry generators can be seen in generalizations of the pattern matching methods introduced by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13]. Algorithms for computing the desired symmetries without solving auxiliary ODEs or even partial differential equations for the case of n-th order ODEs, $n \in \mathbb{N}$, $n \geq 2$, have to be developed.

2. Furthermore, if not sufficiently many symmetry generators can be found, other techniques for solving the ODE under consideration or at least reducing its order have to be found. An alternative approach helping to reduce the order of an ODE is the computation of integrating factors. Some algorithms and heuristics for finding such integrating factors are discussed in the framework of the next two chapters.

[22] An example for the introduction of new functions as solutions of ODEs are the Bessel functions, which can be defined as the solutions of a second order homogeneous linear ODE (see e.g. [38], p. 125 and pp. 622).

Chapter 2

Integrating factors and associated ODE families

2.1 Introduction

In this chapter we present algorithms for computing integrating factors of certain classes of n-th order non–linear ODEs, $n \geq 3$. The results are inspired by the former work of E. S. Cheb-Terrab and A. D. Roche published in [11].

In [11], the authors treat the case of second order ODEs. There exist certain extensions of this work to higher order ODEs also done by E. S. Cheb-Terrab. To our knowledge these results have never been published in a journal or as a preprint on the web. The only hint we could find is a description of certain functions of the "detools"-package included in the computer algebra system MAPLE[1]. This description can be found on the website [14] and currently gives no precise information on the improvements for the case of ODEs of order at least 3. In the description of the function "intfactor"[2] the author speaks about "certain classes" of third and higher order ODEs for which the order can be reduced if the integrating factor is a polynomial in a certain derivative of the dependent variable of the ODE considered.

Our extensions of the ansatzes of E. S. Cheb-Terrab and A. D. Roche for finding integrating factors for second order ODEs directly base on the ideas

[1] The computer algebra system MAPLE is currently available in version 10. Since a few years E. S. Cheb-Terrab is a member of the symbolic computation group associated with MAPLE. He has implemented his algorithms in MAPLE.

[2] This function is an essential part of the MAPLE "detools"-package.

presented in [11]. In fact we mostly do not make further assumptions on the special form of the integrating factor to be computed except for the fact that it does not depend on all "possible variables"[3]. We rather take assumptions into account concerning the algebraic form of the ODEs considered and find heuristics for computing candidates for integrating factors of these ODEs.

We give a more detailed outline of these ideas in Section 2.6. First of all we present a short overview on the ideas of E. S. Cheb-Terrab and A. D. Roche described in [11].

2.2 Basic terminology

We consider n-th order ODEs in the independent variable x and the dependent variable $y(x)$ of the form

$$y^{(n)}(x) = \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x)). \tag{2.2.1}$$

Definition 2.1. We call (2.2.1) an *exact ODE* if the expression

$$y^{(n)}(x) - \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) \tag{2.2.2}$$

is a *total derivative*, i.e. if there is a function $R = R(x, y(x), y'(x), \ldots, y^{(n-1)}(x))$ such that

$$\frac{d}{dx} R(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) = y^{(n)} - \Phi(x, y(x), y'(x), \ldots, y^{(n)}(x)). \tag{2.2.3}$$

The function $R(x, y(x), y'(x), \ldots, y^{(n-1)}(x))$ is called a *first integral* and it provides the *conserved quantity*

$$R(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) = c \tag{2.2.4}$$

of (2.2.1), c some constant of integration.

The notion of a general n-th order exact ODE does not seem to be a traditional terminology in the classical literature on differential equations. The notion of an exact ODE is more commonly known from the context of first order ODEs of the form

$$A(x, y(x)) + y'(x)\, B(x, y(x)) = 0, \tag{2.2.5}$$

[3]This rather unprecise formulation will become clear in the next part of this chapter.

where $A(x, y(x))$ and $B(x, y(x))$ are arbitrary expressions in x and $y(x)$. A first order ODE of the form (2.2.5) is said to be an *exact ODE* if the vector field

$$\vec{V}(x, y(x)) = \begin{pmatrix} A(x, y(x)) \\ B(x, y(x)) \end{pmatrix}$$

has a potential, i.e. if the condition

$$\frac{\partial A(x,y)}{\partial y} = \frac{\partial B(x,y)}{\partial x}$$

holds. If $R(x, y(x))$ is a potential of the vector field $\vec{V}(x, y(x))$, then the solutions of (2.2.5) are given in implicit form by $R(x, y(x)) = c$, c a constant. Furthermore we have

$$\begin{aligned} \frac{d}{dx} R(x, y(x)) &= \frac{\partial R(x,y)}{\partial x} + y'(x) \frac{\partial R(x,y)}{\partial y} \\ &= A(x, y(x)) + y'(x) B(x, y(x)). \end{aligned}$$

Hence, the right–hand–side of (2.2.5) is a total derivative. Definition 2.1 is a sensible generalization of the commonly known notion of first order exact ODEs also used by Cheb-Terrab et. al. in [11].

Remark 2.2. The exactness of an ODE of the form (2.2.1) can be proved without performing any integrations using the Euler operator, which is introduced in Theorem 2.6 in Section 2.5.

For a general ODE of the form (2.2.5), an expression $\mu(x, y(x))$ is called an integrating factor if

$$\mu(x, y(x)) \left(A(x, y(x)) + y'(x)\, B(x, y(x)) \right) = 0 \tag{2.2.6}$$

is an exact ODE. The following definition is the generalization of the notion of integrating factors for general n-th order ODEs of the form (2.2.1).

Definition 2.3. An expression $\mu(x, y(x), \ldots, y^{(n-1)}(x))$ is called an *integrating factor* for (2.2.1) if

$$\mu(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) \left(y^{(n)}(x) - \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) \right) = 0 \tag{2.2.7}$$

is an exact n-th order ODE.

Assume, we have found an integrating factor $\mu(x, y(x), \ldots, y^{(n-1)}(x))$ for an n-th order ODE of the form (2.2.1). Then we know that there is a first integral, i.e. an expression of the form $R(x, y(x), y'(x), \ldots, y^{(n-1)}(x))$, such that the solutions of (2.2.1) are given in implicit form by

$$R(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) = c_n,$$

where c_n is a constant of integration. Hence, finding an integrating factor, multiplying it by $y^{(n)}(x) - \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x)$ and afterwards integrating the resulting expression provides a reduction of the order of (2.2.1) by 1. If $R(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) = c_n$ can be solved for $y^{(n-1)}$, we obtain an $(n-1)$-st order ODE

$$y^{(n-1)}(x) = \widetilde{\Phi}(c_n, x, y(x), y'(x), \ldots, y^{(n-2)}),$$

where the right–hand–side only contains x, $y(x)$ and derivatives of $y(x)$ with respect to x up to order $n-1$ as well as the constant of integration c_n. For this ODE we search for another integrating factor $\widetilde{\mu}(x, y(x), y'(x), \ldots, y^{(n-2)}(x))$ and proceed again as described above. Finally, if it is always possible to solve the resulting first integrals for the highest derivatives and if we succeed to find enough integrating factors, we obtain an implicit expression of the form

$$\Psi(c_1, c_2, \ldots, c_n, x, y(x)) = 0,$$

i.e. a reduction of (2.2.1) giving solutions of (2.2.1) in implicit form depending on n constants of integration (whose choice depends on suitable initial values $y(x_0) = y_0$, $y'(x_0) = y_1$, $\ldots$ $y^{(n-1)}(x_0) = y_{n-1}$ for certain values $x_0, y_0, y_1, \ldots, y_{n-1}$ of the coefficient domain under consideration) and x and $y(x)$, only.

The main problem in trying to solve n-th order ODEs of the form (2.2.1) by searching for integrating factors is that an integrating factor is in general determined by a partial differential equation. In the situation of (2.2.1), the determining equation[4] for an integrating factor of the form $\mu(x, y(x), \ldots, y^{(n-1)}(x))$ is given by

$$\frac{\partial H}{\partial y} - \frac{d}{dx}\Big(\frac{\partial H}{\partial y'}\Big) + \frac{d^2}{dx^2}\Big(\frac{\partial H}{\partial y''}\Big) + \ldots + (-1)^n \frac{d^n}{dx^n}\Big(\frac{\partial H}{\partial y^{(n)}}\Big) = 0, \qquad (2.2.8)$$

[4]Equation (2.2.8) is obtained by applying the Euler operator to the differential expression H (see Theorem 2.6 in Section 2.5 for the introduction of the Euler operator).

where $H = H(x, y(x), y'(x), \ldots, y^{(n)}(x))$ is given by

$$H := \mu(x, y(x), y'(x), \ldots, y^{(n-1)}(x))(y^{(n)}(x) - \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x))).$$

Equation (2.2.8) can be written in the form

$$A(x, y(x), y'(x), \ldots, y^{(2n-3)}(x)) + y^{(2n-2)}(x)B(x, y(x), y'(x), \ldots, y^{(n-1)}(x)) = 0,$$

where $A(x, y(x), y'(x), \ldots, y^{(2n-3)}(x))$ is a polynomial expression of degree $n-1$ in $y^{(n)}(x)$ and a linear polynomial expression in $y^{(k)}(x)$, $n < k \leq 2n-3$.

Even in the case $n = 2$, i.e. when the ODE under consideration is a second order ODE

$$y''(x) = \Phi(x, y(x), y'(x))$$

for which an integrating factor $\mu(x, y(x), y'(x))$ has to be found, equation (2.2.8) is of the form

$$A(x, y(x), y'(x)) + y''(x)B(x, y(x), y'(x)) = 0.$$

By setting $A(x, y(x), y'(x)) = 0$ and $B(x, y(x), y'(x)) = 0$, one obtains the system of partial differential equations

$$\begin{aligned} 0 =& (y'\mu_{y'y} - \mu_y + \mu_{y'x})\Phi + (\Phi_{y'x} + y'\Phi_{y'y} - \Phi_y)\mu + (y')^2\mu_{yy} + \\ & (\mu_y\Phi_{y'} + \mu_{y'}\Phi_y + 2\mu_{xy})y' + \mu_{y'}\Phi_x + \mu_x\Phi_{y'} + \mu_{xx}, \\ 0 =& y'\mu_{y'y} + \Phi\mu_{y'y'} + \mu\Phi_{y'y'} + 2\mu_y + 2\mu_{y'}\Phi_{y'} + \mu_{y'x}, \end{aligned}$$

where indices denote partial derivatives. For these determining equations and further details we refer to [11], Section 2. Cheb-Terrab et. al. remark that solving this system of partial differential equations is in principle as hard to solve as the original ODE.

Recall that our focus lies on methods for solving ODEs in the programming environment of a computer algebra system, where we do not have the possibility (or are not willing due to a loss of efficiency) to use an elaborate symbolic solver for partial differential equations. Therefore we are interested in ways of finding integrating factors without solving any partial differential equations or even other auxiliary ODEs.

The main idea to reach this goal for second order ODEs of the form $y''(x) = \Phi(x, y(x), y'(x))$ presented in [11] is: do not search for integrating factors of the most general form $\mu = \mu(x, y(x), y'(x))$, but try to compute integrating factors of the form such as $\mu = \mu(x, y(x))$, $\mu = \mu(x, y'(x))$ and $\mu = \mu(y(x), y'(x))$, i.e. assume a more special form for the integrating factors to be computed.

Under this special assumptions on the form of the integrating factors to be computed, Cheb-Terrab et. al. succeeded in giving necessary and sufficient conditions for a second order ODE of the form $y''(x) = \Phi(x, y(x), y'(x))$ to admit an integrating factor depending only on two of the "variables" x, $y(x)$ and $y'(x)$.

The basic ideas are:

- Compute the most general class of second order ODEs of the from

  $$y''(x) = \Phi(x, y(x), y'(x))$$

  admitting an integrating factor of the prescribed form.

- Find a necessary condition for a given second order ODE to belong to such a general class of ODEs.

- Read–off the integrating factor from the ODE.

Of course, "reading–off integrating factors" has to be specified in detail. In general, we mean by "reading–off integrating factors" the fact that we do not need to solve any system of partial differential equations nor auxiliary ODEs to compute an integrating factor. We will describe the ideas of Cheb-Terrab et. al. in Section 2.4. Before we come to discuss these ideas, we use the following section to introduce the above mentioned most general class of ODEs associated with an integrating factor.

2.3 The ODE family associated with a given integrating factor

The basic idea used by E. S. Cheb-Terrab and A. D. Roche in [11] to compute integrating factors of ODEs is outlined in the following. For reasons of abbreviation we write y for $y(x)$, y' for $y'(x)$, $y^{(i)}$ for $y^{(i)}(x)$, $1 \leq i \leq n$ from now on. Partial derivatives will be denoted by corresponding indices.

Assume that $\mu(x, y, y', \ldots, y^{(n-1)})$ is an integrating factor for (2.2.1) and $R(x, y, y', \ldots, y^{(n-1)})$ is a first integral. Then we have

$$\begin{aligned} 0 &= \mu(x, y, y', \ldots, y^{(n-1)}) \left(y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})\right) \\ &= \frac{d}{dx} R(x, y, y', \ldots, y^{(n-1)}) \\ &= R_x + y'\, R_y + y''\, R_{y'} + \ldots + y^{(n)}\, R_{y^{(n-1)}}. \end{aligned}$$

Since Φ does not depend on $y^{(n)}$, we can conclude from the above equation that

$$\mu(x, y, y', \ldots, y^{(n-1)})\, y^{(n)} = R_{y^{(n-1)}}(x, y(x), y'(x), \ldots, y^{(n-1)})\, y^{(n)}$$

and, hence,

$$\mu(x, y, y', \ldots, y^{(n-1)}) = R_{y^{(n-1)}}(x, y(x), y'(x), \ldots, y^{(n-1)}).$$

Formal integration with respect to $y^{(n-1)}$ (we simply view $y^{(n-1)}$ as a symbolic variable as if it did not depend on the independent variable x) provides the following representation of the first integral $R = R(x, y, y', \ldots, y^{(n-1)})$, where $G(x, y', y'', \ldots, y^{(n-2)})$ is an arbitrary function in its arguments and plays the role of a "constant of integration":

$$\begin{aligned} R(x, y, y', \ldots, y^{(n-1)}) = G(x, y, y', \ldots, y^{(n-2)}) + \\ \int \mu(x, y, y', \ldots, y^{(n-1)})\, dy^{(n-1)}. \end{aligned}$$

Remark 2.4. Note, that the integral

$$\int \mu(x, y, y', \ldots, y^{(n-1)})\, dy^{(n-1)}$$

has to be understood as a "partial integral", i.e. we view $x, y, y', \ldots, y^{(n-1)}$ as independent variables and treat the integrand $\mu(x, y, y', \ldots, y^{(n-1)})$ as a function of $y^{(n-1)}$ and — with respect to the integration — symbolic "constants" $x, y, y', \ldots, y^{(n-2)}$.

Inserting the above representation for R into the equation

$$R_x + y'\, R_y + y''\, R_{y'} + \ldots + y^{(n)}\, R_{y^{(n-1)}} = 0,$$

we obtain

$$\begin{aligned}0 = \quad & \Big(G(x,y',y'',\ldots,y^{(n-2)}) + \int \mu(x,y,y',\ldots,y^{(n-1)})\,dy^{(n-1)}\Big)_x + \\ & y'\Big(G(x,y',y'',\ldots,y^{(n-2)}) + \int \mu(x,y,y',\ldots,y^{(n-1)})\,dy^{(n-1)}\Big)_y + \\ & y''\Big(G(x,y',y'',\ldots,y^{(n-2)}) + \int \mu(x,y,y',\ldots,y^{(n-1)})\,dy^{(n-1)}\Big)_{y'} + \ldots + \\ & y^{(n)}\Big(G(x,y',y'',\ldots,y^{(n-2)}) + \int \mu(x,y,y',\ldots,y^{(n-1)})\,dy^{(n-1)}\Big)_{y^{(n-1)}}.\end{aligned}$$

Since

$$\Big(\int \mu(x,y,y',\ldots,y^{(n-1)})\,dy^{(n-1)}\Big)_{y^{(n-1)}} = \mu(x,y,y',\ldots,y^{(n-1)})$$

and G does not depend on $y^{(n-1)}$, we obtain from the above equation by solving for $y^{(n)}$ the representation

$$\begin{aligned}y^{(n)} = -\frac{1}{\mu}\Big[\quad & \Big(G + \int \mu\,dy^{(n-1)}\Big)_x + y'\Big(G + \int \mu\,dy^{(n-1)}\Big)_y + \\ & y''\Big(G + \int \mu\,dy^{(n-1)}\Big)_{y'} + \ldots + y^{(n-1)}\Big(G + \int \mu\,dy^{(n-1)}\Big)_{y^{(n-2)}}\Big].\end{aligned}$$

This equation containing the general function $G = G(x,y,y',\ldots,y^{(n-2)})$ can be viewed as the most general class of n-th order ODEs of the form (2.2.1) admitting an integrating factor $\mu = \mu(x,y,y',\ldots,y^{(n-1)})$. We have proved:

Theorem 2.5. *If an n-th order ODE (2.2.1) can be turned into an exact n-th order ODE by multiplication with an integrating factor $\mu = \mu(x,y,y',\ldots,y^{(n-1)})$ it must be of the form*

$$\begin{aligned}y^{(n)} = -\frac{1}{\mu}\Big[\quad & \Big(G + \int \mu\,dy^{(n-1)}\Big)_x + y'\Big(G + \int \mu\,dy^{(n-1)}\Big)_y + \\ & y''\Big(G + \int \mu\,dy^{(n-1)}\Big)_{y'} + \ldots + y^{(n-1)}\Big(G + \int \mu\,dy^{(n-1)}\Big)_{y^{(n-2)}}\Big]\end{aligned}$$

with some function $G = G(x,y,y',\ldots,y^{(n-1)})$ in its argument. The integrations with respect to $y^{(n-1)}$ have to be understood in the sense of Remark 2.4.

In other words: Being an element of the above class of n-th order ODEs is a necessary condition for the existence of an integrating factor of the from $\mu = \mu(x,y,y',\ldots,y^{(n-1)})$.

In the later course of this chapter we will consider numerous applications of Theorem 2.5. Hence, we refer to the following sections for concrete examples.

2.4 Integrating factors for second order ODEs

The application of what we here called Theorem 2.5 to the case of second order ODEs has been done by E. S. Cheb-Terrab and A. D. Roche in [11]. The basic ideas can be illustrated in the most simple way by considering an example from [11].

Due to Theorem 2.5 the most general class of second order ODEs admitting an integrating factor of the form $\mu = \mu(x, y, y')$ is given by

$$y'' = -\frac{1}{\mu}\left[\left(G + \int \mu\, dy'\right)_x + y'\left(G + \int \mu\, dy'\right)_y\right], \tag{2.4.1}$$

where $G = G(x, y)$.

In general it seems much too complicated to decide whether a given concrete second order ODE belongs to the class (2.4.1).

What Cheb-Terrab and Roche did is assuming that μ does not depend on x, y *and* y', but only on x and y or x and y' or y and y'. Let us consider the case $\mu = \mu(x, y)$, i.e. μ does not contain y'.

By taking this restriction into account, the most general class of second order ODEs admitting an integrating factor of the form $\mu = \mu(x, y)$ simplifies (2.4.1) to

$$\begin{aligned} y'' &= -\frac{1}{\mu(x,y)}\left[\left(G(x,y) + \int \mu(x,y)\, dy'\right)_x + y'\left(G(x,y) + \int \mu(x,y)\, dy'\right)_y\right] \\ &= -\frac{1}{\mu(x,y)}\left[G_x(x,y) + y'\,\mu_x(x,y) + y'\,G_y(x,y) + (y')^2\,\mu_y(x,y)\right] \\ &= -\frac{\mu_y(x,y)}{\mu(x,y)}\,(y')^2 - \frac{G_y(x,y) + \mu_x(x,y)}{\mu(x,y)}\,y' - \frac{G_x(x,y)}{\mu(x,y)}, \end{aligned}$$

i.e. the ODEs under consideration are of the form

$$y'' = a(x,y)\,(y')^2 + b(x,y)\,y' + c(x,y), \tag{2.4.2}$$

where $a(x, y)$, $b(x, y)$ and $c(x, y)$ can be determined in terms of $\mu(x, y)$ and

$G(x, y)$ as

$$a(x,y) = -\frac{\mu_y(x,y)}{\mu(x,y)}, \tag{2.4.3}$$

$$b(x,y) = -\frac{G_y(x,y) + \mu_x(x,y)}{\mu(x,y)}, \tag{2.4.4}$$

$$c(x,y) = -\frac{G_x(x,y)}{\mu(x,y)}. \tag{2.4.5}$$

Hence, a given concrete second order ODE can only admit an integrating factor of the form $\mu = \mu(x, y)$, if it is a quadratic polynomial in y' with coefficients in x and y. This is an easy to verify criterion and in the case that a given ODE is of the form (2.4.2), the functions $a(x, y)$, $b(x, y)$ and $c(x, y)$ can be read–off easily[5].

Cheb-Terrab and Roche succeeded in solving the equations (2.4.3), (2.4.4) and (2.4.5). Depending on the fact, whether $2a_x(x, y) - b_y(x, y)$ equals zero or not, and some easy to verify integrability conditions, they could establish closed form solutions for the searched for integrating factor $\mu(x, y)$ or at least a reduction of the problem of solving the system of partial differential equations (2.4.3), (2.4.4) and (2.4.5) to the computation of a solution of a second order homogeneous linear ODE. In the case, where a closed form solution for $\mu(x, y)$ could be obtained, this solution involves integrations, but it is given only in terms of the coefficients $a(x, y)$, $b(x, y)$ and $c(x, y)$ and partial derivatives of these with respect to x and y. Especially, no further solutions of additional auxiliary ODEs or partial differential equations need to be computed.

More precisely, in the case $2a_x(x, y) - b_y(x, y) \neq 0$, Cheb-Terrab and Roche succeeded to express the total derivative

$$\frac{\mu_x(x,y)}{\mu(x,y)} + y' \, \frac{\mu_y(x,y)}{\mu(x,y)}$$

of $\ln(\mu(x, y))$, $y = y(x)$, with respect to x only in terms of $a(x, y)$, $b(x, y)$ and $c(x, y)$ and partial derivatives of these with respect to x and y. Once they got this, the result for $\mu(x, y)$ is obtained by computing the integral of this total derivative and afterwards applying the exponential function to the result. We skip the technical details here, since we will not make direct use of the results

[5] This is not only a theoretical statement. The determination of $a(x, y)$, $b(x, y)$ and $c(x, y)$ is easy to implement in any general purpose computer algebra system since all these software packages offer fast and efficient routines for handling polynomial expressions.

by Cheb-Terrab and Roche for the case of second order ODEs (we only make use of the general underlying ideas used to find integrating factors).

In the case $2a_x(x,y) - b_y(x,y) = 0$ the situation is a bit more complicated. Then $\mu(x,y)$ cannot be directly determined similar to the method described above. It turns out that $\mu(x,y)$ is mainly obtained by means of an integral of a function $\nu(x)$, which arises as a solution of a second order homogeneous linear ODE. Indeed, this fact can be viewed as a simplification of the original problem of determining $\mu(x,y)$ as a solution of a system of partial differential equations. Of course, the computation of the solution $\nu(x)$ of a second order homogeneous linear ODE is a problem in itself.

Although there is no generic formula giving the solution of such an ODE in general, differential Galois theory provides various specialized algorithms for the computation of such solutions (see e.g. [53] or [20]). And it has to be emphasized that having to compute $\nu(x)$ as a solution of a linear ODE is the only case in [11], where an auxiliary ODE has to be solved to find an integrating factor.

As a conclusion: Cheb-Terrab and Roche established an "algebraic" criterion, which is easy to verify, to decide, whether a given second order ODE admits an integrating factor $\mu = \mu(x,y)$. In nearly all cases discussed in [11] they succeeded in solving the symbolic system of partial differential equations (2.4.3), (2.4.4) and (2.4.5) for general functions $a(x,y)$, $b(x,y)$ and $c(x,y)$ *once by hand*, i.e. in a step of pre–processing not using a computer algebra system. Once, the solution is found, the integrating factor is given by this solution.

The main point is that the solving of the system of partial differential equations (2.4.3), (2.4.4) and (2.4.5) is done "once by hand", such that the resulting formulas for the solutions can be directly implemented in a computer algebra system and the system does not have to solve such a system of partial differential equations for every input ODE again[6].

Nevertheless, it has be to mentioned that the procedure sketched above to treat integrating factors of second order ODEs of the form $\mu = \mu(x,y)$ is the easiest case among the three different cases $\mu = \mu(x,y)$, $\mu = \mu(x,y')$ and $\mu = \mu(y,y')$

[6]Proceeding this way it is even possible to implement the above strategy in a computer algebra system, which does not offer any routines for the symbolic solution of linear partial differential equations associated with the integrating factor.

treated in [11]. Especially the case $\mu = \mu(x, y')$ turns out to be far more complicated than the above discussed case.

The aim of this part of the thesis is to present possible extensions of the ideas used for second order ODEs for ODEs of order three and higher. For our cases we first of all need to have a way to check whether a given n-th order ODE is already exact or not.

2.5 The Euler Operator: Proving the exactness of an ODE

To check, whether a given n-th order ODE of the form (2.2.1) is exact, one can use the so–called *Euler Operator*. The following theorem is a special case of Theorem 2.1 in [36], where the authors also propose to use the Euler operator as an algorithmical tool for a test for exactness (see also [51], Theorem 4.4 and Example 4.5, pp. 250).

Theorem 2.6. *Let $f(x, y, y', \ldots, y^{(n)})$ be a function in x, y and the i-th derivatives $y^{(i)}$ of y, $1 \le i \le n$. Then $f(x, y, y', \ldots, y^{(n)})$ is a total derivative if and only if*

$$\sum_{i=0}^{n} (-1)^i \, \frac{d^i}{dx^i} \Big(\frac{\partial f(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}} \Big) = 0. \tag{2.5.1}$$

Here we use the convention $\frac{d^0}{dx^0} \big(\frac{\partial f(x,y,y',\ldots,y^{(n)})}{\partial y^{(0)}} \big) = \frac{\partial f(x,y,y',\ldots,y^{(n)})}{\partial y}$.

Proof. A proof of a more general version of this theorem implying the above assertion can be found in [51], Theorem 4.4 and Example 4.5, pp. 250. □

Theorem 2.6 states that a given n-th order ODE can be checked for exactness without performing any integration.

The application of the Euler operator can be turned directly into an easy to implement and efficient algorithm to check for the exactness of a differential equation[7]. We will make extensive use of Theorem 2.6 in the next sections.

[7] An implementation of the Euler operator in the computer algebra system MATHEMATICA is discussed in Section 3.6.3 and Section 3.6.4 of [3].

Note that the determining equation for an integrating factor of an n-th order ODE presented in (2.2.8) can also be obtained by application of the Euler operator to the differential expression

$$\mu(x, y(x), y'(x), \ldots, y^{(n-1)}(x))(y^{(n)}(x) - \Phi(x, y(x), y'(x), \ldots, y^{(n-1)}(x))$$

and afterwards setting the result equal to zero.

2.6 Integrating factors for third order ODEs

As stated in the introduction to this chapter there are extensions to n-th order ODEs, $n \geq 3$, of the methods of Cheb-Terrab and Roche described in [11]. As far as we know, these extensions and the algorithmic details used to treat ODEs of order $n \geq 3$ are not explained in detail nor published within the framework of an article or preprint.

The only information we could find is available in the internet on the website [8] by E. S. Cheb-Terrab providing the HTML-documentation of the "detools"-package in the computer algebra system MAPLE. There the authors of the package mention that "certain classes" of third and higher order ODEs can be treated in the sense of the above described techniques if the integrating factor is a polynomial in a certain derivative of the dependent variable of the ODE considered. We think that this restriction concerning the algebraic structure of the integrating factors considered seemed necessary to the authors to proceed in the same or a similar way as described in Section 2.4.

As far as we can see, further restrictions of such kind must be taken into account to be able to obtain results leading to implementable and efficient algorithms[8]. In the following we consider third order ODEs

$$y''' = \Phi(x, y, y', y''). \tag{2.6.1}$$

As in the preceding sections, we write y''', y'', y' and y in the following instead of $y'''(x)$, $y''(x)$, $y'(x)$ and $y(x)$. Due to Theorem 2.5, if $\mu = \mu(x, y, y', y'')$ is an

[8]By "implementable and efficient algorithms" we mean that the algorithms and heuristics established in the following part of this chapter serve to compute integrating factors not using any auxiliary ODEs or even partial differential equations to be solved.

integrating factor of ODE (2.6.1), the ODE must be of the form

$$y''' = -\frac{1}{\mu}\left[\left(G + \int \mu\, dy''\right)_x + y'\left(G + \int \mu\, dy''\right)_y + y''\left(G + \int \mu\, dy''\right)_{y'}\right], \tag{2.6.2}$$

where $G = G(x, y, y')$, the "constant" of integration, is an arbitrary function in its arguments. In the following we investigate the cases

- $\mu = \mu(x, y)$,
- $\mu = \mu(x, y')$,
- $\mu = \mu(y, y')$,
- $\mu = \mu(y'')$,
- $\mu = f(x, y, y')\,(y'')^m$, $m \in \mathbb{Z} \setminus \{-2, 0\}$.

In contrast to the results for second order ODEs published in [11], where the authors give necessary and sufficient conditions for the existence of integrating factors of special classes of ODEs, we mainly give necessary conditions[9]. An advantage of our algorithms for third order ODEs introduced in the next subsections is the fact that they can easily be generalized to n-th order ODEs. These generalizations are discussed at the end of this chapter.

2.6.1 Integrating factors $\mu(x, y)$.

Under the assumption $\mu = \mu(x, y)$, the class of ODEs (2.6.2) can be written as

$$y''' = -\frac{\mu_x + y'\,\mu_y + G_{y'}}{\mu}\, y'' - \frac{G_x + y'\,G_y}{\mu}. \tag{2.6.3}$$

Due to Theorem 2.5, this is the most general class of third order ODEs admitting an integrating factor of the form $\mu = \mu(x, y)$. For the class (2.6.3) of ODEs we did not find a satisfying algorithmic approach to check, whether a given ODE of order three belongs to the class or not. So we took into account another simplifying assumption. We assumed that the "constant" of integration $G = G(x, y, y')$ does not depend on y'.

[9]See also Remark 2.17 of Section 2.6.4 for more precise information on what is meant by giving only necessary conditions.

This assumption leads to the general family of ODEs

$$y''' = -\frac{\mu_x + y'\,\mu_y}{\mu}\,y'' - \frac{G_x + y'\,G_y}{\mu} \tag{2.6.4}$$

to be taken into account[10]. If G does not depend on y', (2.6.4) obviously is a linear polynomial in y''. This fact is easy to verify for a given third order ODE. The leading coefficient of this polynomial in y'' is again a linear polynomial, but in y', since $\mu = \mu(x, y)$ does not depend on y' by assumption. We set:

$$a(x, y) := -\frac{\mu_x}{\mu}, \quad b(x, y) := -\frac{\mu_y}{\mu}, \tag{2.6.5}$$

i.e. the leading coefficient of (2.6.4) viewed as a polynomial in y'' can be written as $a(x, y) + y'\,b(x, y)$. For the coefficients of the polynomial $a(x, y) + y'\,b(x, y)$ in y' the symmetry of second derivatives must hold, i.e.

$$a_y(x, y) = b_x(x, y). \tag{2.6.6}$$

By

$$\int a(x, y)\,dx + b(x, y)\,dy$$

we denote the function of x and y uniquely determined up to the addition of a constant, such that its partial derivatives with respect to x and y satisfy

$$\Big(\int a(x, y)\,dx + b(x, y)\,dy\Big)_x = a(x, y)$$

and

$$\Big(\int a(x, y)\,dx + b(x, y)\,dy\Big)_y = b(x, y).$$

Then a candidate[11] for an integrating factor is obtained by

$$\mu(x, y) = \exp\Big(-\int a(x, y)\,dx + b(x, y)\,dy\Big). \tag{2.6.7}$$

Our results can be summarized in the form of the following algorithm, which tries to compute an integrating factor $\mu = \mu(x, y)$ of a given third order ODE.

---

[10]In subsection 2.6.5 on pp. 96 we discuss alternative algorithms and heuristics, for which we do not use this simplifying assumption.

[11]We use the notion of "candidates" for integrating factors, since the above stated conditions are only necessary conditions for the existence of an integrating factor of the prescribed form. See also Remark 2.17 on page 95, where we present further explanations on the fact why our conditions are not sufficient.

Algorithm 2.7. (Computing integrating factors of the form $\mu = \mu(x, y)$ of third order ODEs) Given a third order ODE $y''' = \Phi(x, y, y', y'')$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$$

   and compute

$$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', y'')$ is a linear polynomial in y''. If not, stop the algorithm.

3. Check, whether the leading coefficient of $\Phi(x, y, y', y'')$ as a polynomial in y'' is itself a linear polynomial in y'. If not, stop the algorithm.

4. Determine the coefficients $a(x, y)$ and $b(x, y)$ as in (2.6.5) and check whether (2.6.6) is fulfilled. If not, stop the algorithm.

5. Compute (2.6.7) as a candidate for μ.

6. Compute

$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}.$$

   If $A = 0$, μ is an integrating factor for the ODE. Return the result for μ. Otherwise stop the algorithm. △

Proof. The correctness of the algorithm follows from the preceding discussion. □

In step 1 and step 6 we make use of the Euler operator to check for exactness (see Theorem 2.6). The following example demonstrates the use of Algorithm (2.7) for determining an integrating factor of a third order ODE belonging to the special class of ODEs (2.6.4) under consideration.

Example 2.8. The ODE

$$y''' = -y'' - y'\,y'' - \frac{y}{\exp(x+y)} - x\,\frac{y'}{\exp(x+y)}$$

is not exact, since application of the Euler operator (Theorem 2.6) to the expression

$$y''' + y'' + y'\,y'' + \frac{y}{\exp(x+y)} + x\,\frac{y'}{\exp(x+y)}$$

provides $\frac{x-y}{\exp(x+y)}$, i.e. the expression is not a total derivative. The functions $a(x,y)$ and $b(x,y)$ in step 4 of Algorithm 2.7 are given by

$$a(x,y) = -1, \qquad b(x,y) = -1.$$

Hence, a candidate for an integrating factor of the given ODE is computed by (2.6.7) providing

$$\begin{aligned}\mu(x,y) &= \exp\Big(-\int a(x,y)\,dx + b(x,y)\,dy\Big)\\ &= \exp(x+y),\end{aligned}$$

where the constant of integration can be ignored, since whenever $\mu(x,y)$ is an integrating factor, then also $c\,\mu(x,y)$ for an arbitrary non–zero constant c. Multiplication of

$$y''' + y'' + y'\,y'' + \frac{y}{\exp(x+y)} + x\,\frac{y'}{\exp(x+y)}$$

by μ and afterwards application of the Euler operator to this expression proves that the resulting expression is indeed a total derivative and, hence,

$$\mu(x,y) = \exp(x+y)$$

is an integrating factor for the given ODE. $\diamond$

2.6.2 Integrating factors $\mu(x, y')$.

Under the assumption $\mu = \mu(x, y')$, the class of ODEs (2.6.2) can be written as

$$y''' = -\frac{\mu_{y'}}{\mu}\,(y'')^2 - \frac{\mu_x + G_{y'}}{\mu}\,y'' - \frac{G_x + y'\,G_y}{\mu}. \tag{2.6.8}$$

Due to Theorem 2.5, this is the most general class of third order ODEs admitting an integrating factor of the form $\mu = \mu(x, y')$. As in the preceding case $\mu = \mu(x, y)$, we take into account the simplifying assumption, that the "constant" of integration $G = G(x, y, y')$ does not depend on y'.

Under this assumption (2.6.8) becomes

$$y''' = -\frac{\mu_{y'}}{\mu}\,(y'')^2 - \frac{\mu_x}{\mu}\,y'' - \frac{G_x + y'\,G_y}{\mu}. \tag{2.6.9}$$

Since μ and G do not depend on y'', (2.6.9) can be viewed as a quadratic polynomial in y'' with coefficients depending on x, y and y'. We define

$$a(x, y') := -\frac{\mu_{y'}}{\mu}, \qquad b(x, y') := -\frac{\mu_x}{\mu}, \tag{2.6.10}$$

i.e. $a(x, y')$ is the coefficient of $(y'')^2$ and $b(x, y')$ is the coefficient y''. Again, for these functions the symmetry of second derivatives must hold, i.e.

$$a_x(x, y') = b_{y'}(x, y'). \tag{2.6.11}$$

If we denote by

$$\int a(x, y')dy' + b(x, y')dx$$

the function in x and y' uniquely determined up to the addition of a constant, whose partial derivatives with respect to y' and x satisfy

$$\Big(\int a(x, y')dy' + b(x, y')dx\Big)_{y'} = a(x, y')$$

and

$$\Big(\int a(x, y')dy' + b(x, y')dx\Big)_{x} = b(x, y').$$

A candidate[12] for $\mu = \mu(x, y')$ is found to be

$$\mu(x, y') = \exp\Big(-\int a(x, y')\,dy' + b(x, y')\,dx\Big). \tag{2.6.12}$$

From the above discussion we can deduce Algorithm 2.9, which tries to compute an integrating factor of the form $\mu = \mu(x, y')$.

[12] We use the notion of "candidates" for integrating factors, since the above stated conditions are only necessary conditions for the existence of an integrating factor of the prescribed form. See also Remark 2.17 on page 95, where we present further explanations on the reason why our conditions are not sufficient.

Algorithm 2.9. (Computing integrating factors of the form $\mu = \mu(x, y')$ of third order ODEs) Given a third order ODE $y''' = \Phi(x, y, y', y'')$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$$

   and compute

$$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', y'')$ is a quadratic polynomial in y''. If not, stop the algorithm.

3. Check, whether the coefficients of $(y'')^2$ and y'' in $\Phi(x, y, y', y'')$ viewed as a polynomial in y'' do not contain y. If not, stop the algorithm.

4. Determine the functions $a(x, y')$ and $b(x, y')$ as in (2.6.10) and check, whether (2.6.11) is fulfilled. If not, stop the algorithm.

5. Compute (2.6.12) as a candidate for μ.

6. Compute

$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}.$$

   If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

Steps 1 and 6 of Algorithm 2.9 make use of the Euler operator to decide, whether the corresponding expression is a total derivative. Let us consider an example, in which Algorithm 2.9 can be applied.

Example 2.10. The ODE

$$y''' = -(y'')^2 - y'' - \frac{2\,x\,y}{\exp(x+y')} - x^2\,\frac{y'}{\exp(x+y')}$$

is not exact, since application of the Euler operator (Theorem 2.6) to the expression

$$y''' + (y'')^2 + y'' + \frac{2\,x\,y}{\exp(x+y')} + x^2\,\frac{y'}{\exp(x+y')}$$

provides

$$\frac{1}{\exp(x+y')}\big(2\,y + 4\,x\,y' - x^2\,y' + 2\,x^2\,y'' - 2\,x\,y + \\ x^2 - x^2\,y'\,y'' - 2\,x\,y\,y'' + 2\,\exp(x+y')\,y''''\big),$$

i.e. the expression is not a total derivative. The functions $a(x,y')$ and $b(x,y')$ in step 4 of the Algorithm 2.9 are given by

$$a(x,y') = -1, \qquad b(x,y') = -1.$$

Hence, a candidate for an integrating factor of the given ODE is computed by (2.6.12) providing

$$\begin{aligned}\mu(x,y') &= \exp\Big(-\int a(x,y')\,dy' + b(x,y')\,dx\Big)\\ &= \exp(x+y'),\end{aligned}$$

where the constant of integration can be ignored, since whenever $\mu(x,y')$ is an integrating factor, then also $c\,\mu(x,y')$ for an arbitrary non–zero constant c. Multiplication of

$$y''' + (y'')^2 + y'' + \frac{2\,x\,y}{\exp(x+y')} + x^2\,\frac{y'}{\exp(x+y')}$$

by μ and afterwards applying the Euler operator to this expression proves that the resulting expression is indeed a total derivative and, hence,

$$\mu(x,y') = \exp(x+y')$$

is an integrating factor for the given ODE. $\diamond$

2.6.3 Integrating factors $\mu(y, y')$.

Under the assumption $\mu = \mu(y, y')$, the class of ODEs (2.6.2) can be written as

$$y''' = -\frac{\mu_{y'}}{\mu}(y'')^2 - \frac{\mu_y\, y' + G_{y'}}{\mu}\, y'' - \frac{G_x + y'\, G_y}{\mu}. \tag{2.6.13}$$

Due to Theorem 2.5, this is the most general class of third order ODEs admitting an integrating factor of the form $\mu = \mu(y, y')$. As in the treatment of the preceding cases $\mu = \mu(x, y)$ and $\mu = \mu(x, y')$, we take into account the simplifying assumption that the "constant" of integration $G = G(x, y, y')$ does not depend on y'.

Under this assumption (2.6.13) becomes

$$y''' = -\frac{\mu_{y'}}{\mu}(y'')^2 - \frac{\mu_y\, y'}{\mu}\, y'' - \frac{G_x + y'\, G_y}{\mu}. \tag{2.6.14}$$

As in the case $\mu = \mu(x, y')$, the ODE (2.6.14) can be viewed as a quadratic polynomial in y'', since G and μ do not depend on y''. We consider

$$a(y, y') := -\frac{\mu_{y'}}{\mu}, \qquad b(y, y') := -\frac{\mu_y\, y'}{\mu}. \tag{2.6.15}$$

The symmetry conditions for the second derivatives (2.6.6) and (2.6.11) are slightly different in the case $\mu = \mu(y, y')$ compared to the preceding cases $\mu = \mu(x, y)$ and $\mu(x, y')$. In the situation of (2.6.14)

$$a_y(y, y') = \Big(\frac{b(y, y')}{y'}\Big)_{y'} \tag{2.6.16}$$

must hold. Similar to the former cases, where we computed candidates for integrating factors $\mu = \mu(x, y)$ and $\mu = \mu(x, y')$, we denote by

$$\int a(y, y')\, dy' + \frac{b(y, y')}{y'}\, dy$$

the function in y and y' uniquely determined up to the addition of a constant, such that its partial derivatives with respect to y' and y satisfy

$$\Big(\int a(y, y')\, dy' + \frac{b(y, y')}{y'}\, dy\Big)_{y'} = a(y, y')$$

and

$$\Big(\int a(y,y')\,dy' + \frac{b(y,y')}{y'}\,dy\Big)_y = \frac{b(y,y')}{y'}.$$

Then a candidate[13] for $\mu = \mu(y,y')$ is obtained to be

$$\mu(y,y') = \exp\Big(-\int a(y,y')\,dy' + \frac{b(y,y')}{y'}\,dy\Big). \tag{2.6.17}$$

We summarize our results in an algorithm, which tries to compute an integrating factor of the form $\mu(y,y')$ for a given third order ODE.

Algorithm 2.11. (Computing integrating factors of the form $\mu = \mu(y,y')$ of third order ODEs) Given a third order ODE $y''' = \Phi(x,y,y',y'')$ proceed as follows:

1. Check the ODE for exactness, i.e. set

   $$\Psi(x,y,y',y'',y''') := y''' - \Phi(x,y,y',y'')$$

   and compute

   $$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x,y,y',y'')$ is a quadratic polynomial in y''. If not, stop the algorithm.

3. Check, whether the coefficients of $(y'')^2$ and y'' in $\Phi(x,y,y',y'')$ viewed as a polynomial in y'' only depend on y and y'. If not, stop the algorithm.

4. Determine the functions $a(y,y')$ and $b(y,y')$ as in (2.6.15) and check, whether (2.6.16) is fulfilled. If not, stop the algorithm.

5. Compute (2.6.17) as a candidate for μ.

[13] We use the notion of "candidates" for integrating factors, since the above stated conditions are only necessary conditions for the existence of an integrating factor of the prescribed form. See also Remark 2.17 on page 95, where we present further explanations on the reasons why our conditions are not sufficient.

6. Compute

$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}.$$

   If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

Algorithm 2.11 is used to find an integrating of a member of the special class of ODEs (2.6.14) in the next example.

Example 2.12. The ODE

$$y''' = -(y'')^2 - y'\,y'' - \frac{y}{\exp(y+y')} - x\,\frac{y'}{\exp(y+y')}$$

is not exact, since application of the Euler operator (Theorem 2.6) to the expression

$$y''' + (y'')^2 + y'\,y'' + \frac{y}{\exp(y+y')} + x\,\frac{y'}{\exp(y+y')}$$

provides

$$\frac{1}{\exp(y+y')}\big(-y - y\,y' + 2\,x\,y'' - y\,y'' - x\,(y')^2 + 2\,y' - \\ x\,y'\,y'' + 2\,\exp(y+y')\,y''''\big),$$

i.e. the expression is not a total derivative. The functions $a(y,y')$ and $b(y,y')$ in step 4 of the Algorithm 2.11 are given by

$$a(y,y') = -1, \qquad b(y,y') = -y'.$$

Hence, a candidate for an integrating factor of the given ODE is computed by (2.6.17) providing

$$\begin{aligned}\mu(y,y') &= \exp\Big(-\int a(y,y')\,dy' + \frac{b(y,y')}{y'}\,dy\Big)\\ &= \exp(y+y'),\end{aligned}$$

where the constant of integration can be ignored, since if $\mu(y, y')$ is an integrating factor, then also $c\,\mu(y, y')$ for any non–zero constant c. Multiplication of

$$y''' + (y'')^2 + y'\,y'' + \frac{y}{\exp(y+y')} + x\,\frac{y'}{\exp(y+y')}$$

by μ and afterwards applying the Euler operator to this expression proves that the resulting expression is indeed a total derivative and, hence,

$$\mu(y, y') = \exp(y + y')$$

is an integrating factor for the given ODE. ⋄

Remark 2.13. Algorithms 2.7, 2.9 and 2.11 also work in the cases, where μ does not depend on *two* elements of the set $\{x, y, y'\}$, but only on *one* of them. The special cases $\mu = \mu(x)$, $\mu = \mu(y)$ and $\mu = \mu(y')$ are automatically treated by the algorithms stated. This also implies that e.g. for computing an integrating factor of the form $\mu = \mu(y)$, Algorithm 2.7 or Algorithm 2.11 may be used. This fact is illustrated by Example 2.14.

Example 2.14. The ODE

$$y''' = \frac{y'}{y}\,y'' - y^2\,y' \tag{2.6.18}$$

is not exact, but it admits the integrating factor $\mu = \mu(y) = \frac{1}{y}$ (see [45], p. 602, ODE 7.7).

The functions $a(x, y)$ and $b(x, y)$ in step 4 of Algorithm 2.7 are given by

$$a(x, y) = 0, \qquad b(x, y) = \frac{1}{y}.$$

Hence, a candidate for an integrating factor of the given ODE is computed by (2.6.7) providing

$$\begin{aligned}\mu(x, y) &= \exp\Big(-\int a(x, y)\,dx + b(x, y)\,dy\Big)\\ &= \frac{1}{y}.\end{aligned}$$

Indeed, this is an integrating factor for (2.6.18) and constants of integration can be ignored as in the preceding examples of this section.

Alternatively, also Algorithm 2.11 can be used to determine the searched for integrating factor. The functions $a(y, y')$ and $b(y, y')$ in step 4 of the Algorithm 2.11 are given by

$$a(y, y') = 0, \qquad b(y, y') = \frac{y'}{y}.$$

Hence, a candidate for an integrating factor of the given ODE is computed by (2.6.17) providing

$$\begin{aligned}\mu(y, y') &= \exp\Big(-\int a(y, y')\, dy' + \frac{b(y, y')}{y'}\, dy\Big) \\ &= \frac{1}{y},\end{aligned}$$

i.e. application of both algorithms 2.7 and 2.11 leads to the same integrating factor. ◇

2.6.4 Integrating factors $\mu(y'')$.

Under the assumption $\mu = \mu(y'')$, the general class of ODEs (2.6.2) reads

$$y''' = -\frac{G_x + y'\, G_y + y''\, G_{y'}}{\mu}. \tag{2.6.19}$$

Due to Theorem 2.5, this is the most general class of third order ODEs admitting an integrating factor of the form $\mu = \mu(y'')$. Since $G(x, y, y')$ is an arbitrary function in its arguments, the class of ODEs (2.6.19) is not of such a simple algebraic structure as the classes discussed for the cases $\mu = \mu(x, y)$, $\mu = \mu(x, y')$ and $\mu = \mu(y, y')$, where we demanded that $G(x, y, y')$ does not depend on y' and found out that the class of ODEs is polynomial in y''. In the above class (2.6.19) it even does not really simplify the situation if we assume that $G(x, y, y')$ does not depend on y', since this assumption provides

$$y''' = -\frac{1}{\mu}\, y'' - \frac{G_x + y'\, G_y}{\mu}. \tag{2.6.20}$$

At first glimpse, it seems easy to find out μ as the negative reciprocal of the coefficient of y''. But since μ depends on y'', the right–hand–side of the equation does not need to be a polynomial in y''. Furthermore, the problem of computing μ from the right–hand–side of the ODE by simple means as used in the preceding part of this section has not been simplified by assuming that $G(x,y,y')$ does not depend on y'. If we drop this assumption, the class of ODEs can be written as

$$y''' = -\frac{G_{y'}}{\mu}\, y'' - \frac{G_x + y'\, G_y}{\mu}. \tag{2.6.21}$$

To read–off μ from the right–hand–side of this equation is not more difficult than to find a candidate for μ in equation (2.6.20), since G does not depend on y'' anyway.

Hence, instead of taking into account the simplifying assumption that the "constant" of integration $G(x,y,y')$ does not depend on y' as in the treatment of the preceding cases, we demand for the case $\mu = \mu(y'')$, that $\mu(0) \neq 0$. If $\mu(0) \neq 0$, μ cannot be of the form $\mu = y'' g(y'')$ for an arbitrary function $g(y'')$, hence, $\frac{1}{\mu}$ is a proper factor of

$$\frac{G_{y'}}{\mu}\, y''$$

and

$$\frac{G_x + y'\, G_y}{\mu}.$$

Thus one can try to factor the right–hand–side of (2.6.21) into the product of a factor depending only on y'' as a candidate for $-\frac{1}{\mu}$ and a second factor as a candidate for

$$G_{y'}\, y'' + G_x + G_y\, y'.$$

If this second factor is a linear polynomial in y'' with coefficients depending only on x, y and y' and a total derivative, there is at least a chance that the negative reciprocal of the first factor depending only on y'' is a candidate for an integrating factor[14].

Of course, we should first investigate, if there is a term of the form $g(y'')\, y''$ in (2.6.21). If this is the case, we may assume that $G(x,y,y')$ indeed does not depend on y', i.e. we are in the situation of (2.6.20) and the reciprocal of $g(y'')$ is a candidate for an integrating factor of the given ODE (in fact, the negative of the reciprocal of $g(y'')$ corresponds to $\mu(y'')$, but constant multiples can be ignored).

[14]Of course, finding a common factor depending only on y'' in an arbitrary expression in x, y, y' and y'' is a difficult problem and one cannot expect to succeed in all possible settings. Nevertheless, modern computer algebra systems offer routines for factoring not only polynomials, but also arbitrary expressions.

We sum up the ideas for the case $\mu = \mu(y'')$, $\mu(0) \neq 0$, in the following algorithm[15]:

Algorithm 2.15. (Computing integrating factors of the form $\mu = \mu(y'')$, $\mu(0) \neq 0$, of third order ODEs) Given a third order ODE $y''' = \Phi(x, y, y', y'')$ proceed as follows.

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$$

   and compute

   $$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Try to find out, if $\Phi(x, y, y', y'')$ contains a factor $g(y'')$ depending only on y''. If this is not the case, stop the algorithm.

3. Let $\Phi(x, y, y', y'') = g(y'')\, h(x, y, y', y'')$. Compute

   $$T := h_y - \frac{d}{dx}h_{y'} + \frac{d^2}{dx^2}h_{y''} - \frac{d^3}{dx^3}h_{y'''}.$$

   If $T = 0$, i.e. $h(x, y, y', y'')$ is a total derivative and a linear polynomial in y'', go to step 4. If this is not the case, stop the algorithm.

4. Choose $\mu = \frac{1}{g(y'')}$ as a candidate for an integrating factor of the given ODE.

5. Set $\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$ and compute

   $$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}.$$

   If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

[15] In principle it should better be called a heuristic than an algorithm, since it strongly depends on the power of the factoring routine available, when implemented in practice.

Proof. The correctness of the algorithm follows from the preceding discussion. □

Note that in step 1 of Algorithm 2.15 first of all the algorithms 2.7, 2.9 and 2.11 should be used, since they are much more systematic. The above algorithm may be considered as the "last hope" for finding an integrating factor depending only on y'' if algorithms 2.7, 2.9 and 2.11 fail.

We apply Algorithm 2.15 in an example of an ODE belonging to the general class of ODEs (2.6.19).

Example 2.16. Consider the ODE

$$y''' = \frac{\cos(x+y+y')}{\exp(y'')} + y' \, \frac{\cos(x+y+y')}{\exp(y'')} + y'' \, \frac{\cos(x+y+y')}{\exp(y'')}.$$

This ODE is not a member of the classes (2.6.4), (2.6.14) and (2.6.9) nor is the expression

$$y''' - \frac{\cos(x+y+y')}{\exp(y'')} - y' \, \frac{\cos(x+y+y')}{\exp(y'')} - y'' \, \frac{\cos(x+y+y')}{\exp(y'')}$$

a total derivative (the expression obtained by applying the Euler Operator from Theorem 2.6 is too large to be displayed here).

Nevertheless, it is quite easy to see that the right–hand–side of the given ODE contains a factor depending only on y'', i.e. $\frac{1}{\exp(y'')}$. The remaining factor is

$$-\cos(x+y+y') - y' \, \cos(x+y+y') - y'' \, \cos(x+y+y'),$$

which is a linear polynomial in y'' and a total derivative, since it equals $-\frac{d}{dx}\sin(x+y+y')$. Hence, Algorithm 2.15 determines $\mu = \exp(y'')$ as a candidate for an integrating factor. Multiplying

$$y''' - \frac{\cos(x+y+y')}{\exp(y'')} - y' \, \frac{\cos(x+y+y')}{\exp(y'')} - y'' \, \frac{\cos(x+y+y')}{\exp(y'')}$$

by μ provides

$$\exp(y'')\, y''' - \cos(x+y+y') - y' \, \cos(x+y+y') - y'' \, \cos(x+y+y'),$$

which is a total derivative and, hence,

$$\mu(y'') = \exp(y'')$$

an integrating factor for the given ODE. ⋄

We close this subsection with a remark on the necessity of checking for total derivatives in the last step of Algorithm 2.7, Algorithm 2.9, Algorithm 2.11 and Algorithm 2.15.

Remark 2.17. The ODE classes (2.6.4), (2.6.9), (2.6.14) and (2.6.19) and the conditions on the algebraic structure of the ODEs established in the former part of this section are necessary conditions for the existence of integrating factors of the form $\mu = \mu(x, y)$, $\mu = \mu(x, y')$, $\mu = \mu(y, y')$ and $\mu = \mu(y'')$, respectively, but they are not sufficient.

In other words: It may happen that the Algorithms 2.7, 2.9, 2.11 and 2.15 compute a candidate for μ, but μ is not an integrating factor for the ODE considered. The class (2.6.9) of ODEs considered in the case $\mu = \mu(x, y')$ was

$$y''' = -\frac{\mu_{y'}}{\mu}\,(y'')^2 - \frac{\mu_x}{\mu}\,y'' - \frac{G_x + y'\,G_y}{\mu}.$$

We found a candidate for an integrating factor of the form $\mu = \mu(x, y')$ by noticing that each member of this class of ODEs must be a quadratic polynomial in y''. Then we took the coefficients of $(y'')^2$ and y'' and derived a result for μ, but without taking into account the constant term

$$-\frac{G_x + y'\,G_y}{\mu}$$

(constant with respect to y''). By ignoring the constant term, it may of course happen that Algorithm 2.9 computes a candidate for μ, but μ does not appear in the denominator of the constant term of the given ODE or it appears in the denominator, but the numerator is not the total derivative of a function $G = G(x, y)$.

The following example illustrates a case, where Algorithm 2.9 finds a candidate for an integrating factor, which is in fact not an integrating factor.

Example 2.18. The ODE

$$y''' = -(y'')^2 - y'' - \frac{y}{\exp(x + y')} - \frac{y\,y'}{\exp(x + y')}$$

is not exact, since application of the Euler operator (Theorem 2.6) to the expression

$$y''' + (y'')^2 + y'' + \frac{y}{\exp(x + y')} + \frac{y\,y'}{\exp(x + y')}$$

provides

$$\frac{y\,y'' - y\,y' + (y')^2 + y' - y\,y'\,y'' + 2\,\exp(x+y')\,y'''' + 1}{\exp(x+y')},$$

i.e. the expression is not a total derivative. The functions $a(x, y')$ and $b(x, y')$ determined in step 4 of Algorithm 2.9 are

$$a(x, y') = -1, \qquad b(x, y') = -1.$$

Hence, a candidate for an integrating factor of the given ODE is computed in step 5 of Algorithm 2.9 to be

$$\begin{aligned}\mu &= \exp\Big(-\int a(x, y')\,dy' + b(x, y')\,dx\Big)\\ &= \exp(x+y').\end{aligned}$$

Multiplying

$$y''' + (y'')^2 + y'' + \frac{y}{\exp(x+y')} + \frac{y\,y'}{\exp(x+y')}$$

by μ and afterwards applying of the Euler Operator (see Theorem 2.6) to the resulting expression provides the result 1. Hence, multiplication of the given ODE with μ does not lead to a first integral and thus Algorithm 2.9 fails. $\diamond$

This shows that the test for exactness in the last step of each of the Algorithms 2.7, 2.9, 2.11 and 2.15 cannot be removed. The same holds for the algorithms presented in the next subsection, where we consider a more special form of integrating factors.

2.6.5 Integrating factors $f(x, y, y')(y'')^m$.

Recall that in general for a third order ODE an integrating factor admitted by the ODE can involve x, y, y' and y''. In the former part of this chapter, we investigated integrating factors only depending on at most two of the “variables” x, y, y' and y''. In this section we present algorithms to find integrating factors depending on x, y, y' and y''. Furthermore, in the previous part, we always made simplifying assumptions concerning the “constant of integration” G appearing in class of ODEs (2.6.2) under consideration. In the following, we also allow the case $G = G(x, y, y')$ instead of $G = G(x, y)$, but we have to assume that $\frac{d}{dx}G(x, y, y') \neq 0$, i.e. G must not be a constant[16].

[16]The reason for the assumption will become clear in the following.

We consider the more concrete ansatz $\mu = f(x, y, y')(y'')^m$, $m \in \mathbb{Z} \setminus \{-2, 0\}$, $f(x, y, y')$ an arbitrary function in its arguments, concerning the form of the integrating factor[17]. As in the former part of this chapter, we first compute the most general class of third order ODEs admitting an integrating factor of the form $\mu = f(x, y, y')(y'')^m$ and afterwards give an algorithm how to compute a candidate for an integrating factor from an ODE belonging to the class of ODEs under consideration.

In the following we treat the three cases

- $\mu = f(x, y, y')(y'')^m$, $m \in \mathbb{N}$,
- $\mu = f(x, y, y')\frac{1}{(y'')^m}$, $m \in \mathbb{N}$, $m \geq 3$,
- $\mu = f(x, y, y')\frac{1}{y''}$,

separately.

The case $\mu = f(x, y, y')(y'')^m$, $m \in \mathbb{N}$. If we take into account the prescribed form of integrating factors, the partial integrals appearing in (2.6.2) are given by

$$\begin{aligned}
\int \mu \, dy'' &= \frac{1}{m+1} f (y'')^{m+1}, & \Big(\int \mu \, dy''\Big)_x &= \frac{1}{m+1} f_x (y'')^{m+1}, \\
\Big(\int \mu \, dy''\Big)_y &= \frac{1}{m+1} f_y (y'')^{m+1}, & \Big(\int \mu \, dy''\Big)_{y'} &= \frac{1}{m+1} f_{y'} (y'')^{m+1}.
\end{aligned}$$

Consequently, the most general class of third order ODEs admitting an integrating factor of the prescribed form reads

$$y''' = -\frac{G_x + y' G_y}{f (y'')^m} - \frac{G_{y'}}{f (y'')^{m-1}} - \frac{1}{m+1}\Big(\frac{f_x}{f} y'' + \frac{f_y}{f} y' y'' + \frac{f_{y'}}{f}(y'')^2\Big). \quad (2.6.22)$$

Note that $f(y'')^m$ and $f(y'')^{m-1}$ denote the multiplication of f by $(y'')^m$ and $(y'')^{m-1}$, respectively, and not the application of f to $(y'')^m$ and $(y'')^{m-1}$.

The right–hand–side of (2.6.22) is a rational expression in y'', since neither $f = f(x, y, y')$ nor $G = G(x, y, y')$ do contain y''. Assume we are given an ODE

[17]The case $m = 0$ is excluded, since for $m = 0$ the integrating factors considered would not depend on y'' at all. The case $m = -2$ is excluded for a technical reason, which will become obvious soon.

of this form, i.e. neither $G_x + y'G_y$ nor $G_{y'}$ vanish. Then we can read–off the highest power of y'' appearing in a denominator of the expressions forming the right–hand–side of the given ODE. This gives a candidate for the power m of y'' appearing in the searched for integrating factor. Furthermore, (2.6.22) has a polynomial part in y'', i.e. the expression

$$-\frac{1}{m+1}\Big(\frac{f_x}{f}+\frac{f_y}{f}y'\Big)y'' - \frac{1}{m+1}\frac{f_{y'}}{f}(y'')^2.$$

Hence, we can read–off the quantities[18]

$$T_1 = -\frac{1}{m+1}\Big(\frac{f_x}{f}+\frac{f_y}{f}y'\Big), \qquad T_2 = -\frac{1}{m+1}\frac{f_{y'}}{f},$$

i.e. T_1 is the coefficient of y'' and T_2 is the coefficient of $(y'')^2$ in the polynomial part of (2.6.22)[19]. Under the assumptions concerning the form of the integrating factor, $-(m+1)(T_1 + y''T_2)$ has to be the total derivative of $\ln(f)$ with respect to x and we may apply the Euler operator (see Theorem 2.6) to this expression to check, whether this is indeed the case. If $-(m+1)(T_1 + y''T_2)$ is a total derivative, we have found a candidate for an integrating factor to be

$$\mu = \mu(x, y, y', y'') = \exp\big(-(m+1)\,D^{-1}(T_1 + y''T_2)\big)\,(y'')^m, \tag{2.6.23}$$

where by $D^{-1}(T_1 + y''T_2)$ we denote a function $F(x, y, y')$ uniquely determined up to the addition of a constant of integration, such that its total derivative with respect to x equals $T_1 + y''T_2$, i.e.

$$\frac{d}{dx}D^{-1}(T_1 + y''T_2) = \frac{d}{dx}F(x, y, y') = T_1 + y''T_2. \tag{2.6.24}$$

If either $G_x + y'G_y = 0$ or $G_{y'} = 0$, then (2.6.22) is an ODE of the form

$$y''' = \frac{H(x, y, y')}{f(y'')^l} - \frac{1}{m+1}\Big(\frac{f_x}{f}y'' + \frac{f_y}{f}y'y'' + \frac{f_{y'}}{f}(y'')^2\Big) \tag{2.6.25}$$

[18] Note that also in the case $m = 1$, these coefficients can be determined. If $m = 1$, then the expression $-\frac{G_{y'}}{f(y'')^{m-1}}$ in the right–hand–side of (2.6.22) contributes a constant term with respect to y'', but the quadratic polynomial $T_1y'' + T_2(y'')^2$ by assumption does not have a term constant with respect to y''.

[19] We do not have to make further case differentiations, whether $T_1 = 0$ or $T_2 = 0$. Even if both expressions vanish, i.e. the polynomial part of (2.6.22) is the zero polynomial, the definition of the candidate for the integrating factor (2.6.23) still makes sense. If $T_1 = 0$ and $T_2 = 0$, the integrating factor is — up to a multiplicative constant — a pure power of y'' and the factor given in terms of the exponential function in (2.6.23) also reduces to a constant (see also Example 2.25 for the application of Algorithm 2.22).

for some function $H(x, y, y')$ in its arguments and $l \in \mathbb{N}$, i.e. it is still a rational expression in y'', but we cannot read–off the candidate for the power m of y'' in the searched for integrating factor directly, since in the case $G_x + y'G_y = 0$ we have $l = m - 1$, whereas in the case $G_{y'} = 0$ we have $l = m$. Hence, if the given ODE is of the form (2.6.25), we have two candidates for m. Consequently, we obtain two candidates for an integrating factor, namely

$$\mu_1 = \mu_1(x, y, y', y'') = \exp\left(-(l+2)\, D^{-1}(T_1 + y''T_2)\right)(y'')^{l+1} \tag{2.6.26}$$

and

$$\mu_2 = \mu_2(x, y, y', y'') = \exp\left(-(l+1)\, D^{-1}(T_1 + y''T_2)\right)(y'')^{l}. \tag{2.6.27}$$

Since the integrand in the formulas for both candidates is the same, it is easy to compute both candidates in practice without significant loss of efficiency.

In the case $G_x + y'G_y = 0 = G_{y'}$, i.e. $\frac{d}{dx}G(x, y, y') = 0$, the ODE (2.6.22) is a quadratic polynomial in y''. This case is excluded in the following algorithm, since there seems to be no easy way to find a candidate for the power m without solving any auxiliary ODEs or partial differential equations[20].

Algorithm 2.19. (Computing integrating factors of the form $\mu = f(x, y, y')(y'')^m$, $m \geq 1$, of third order ODEs) Given a third order ODE $y''' = \Phi(x, y, y', y'')$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$$

and compute

$$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', y'')$ is a rational expression in y'' of the form

$$\frac{H_1(x, y, y')}{(y'')^m} + \frac{H_2(x, y, y')}{(y'')^{m-1}} + T_1(x, y, y')y'' + T_2(x, y, y')(y'')^2,$$

[20]Note that $\frac{f_x}{f}, \frac{f_y}{f}, \frac{f_{y'}}{f}$ may have a common constant factor, which in general makes it impossible to simply find a candidate for m by factoring out the constants.

where $H_1 \neq 0$, $H_2 \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments. If this is the case and

$$(T_1+T_2y'')_y - \frac{d}{dx}(T_1+T_2y'')_{y'} + \frac{d^2}{dx^2}(T_1+T_2y'')_{y''} - \frac{d^3}{dx^3}(T_1+T_2y'')_{y'''} = 0$$

i.e. $T_1(x,y,y') + T_2(x,y,y')y''$ is a total derivative, go to step 4.

3. Check, whether $\Phi(x,y,y',y'')$ is a rational expression in y'' of the form

$$\frac{H(x,y,y')}{(y'')^l} + T_1(x,y,y')y'' + T_2(x,y,y')(y'')^2,$$

where $H \neq 0$, T_1 and T_2 can be arbitrary functions of their arguments. If this is the case and

$$(T_1+T_2y'')_y - \frac{d}{dx}(T_1+T_2y'')_{y'} + \frac{d^2}{dx^2}(T_1+T_2y'')_{y''} - \frac{d^3}{dx^3}(T_1+T_2y'')_{y'''} = 0$$

i.e. $T_1(x,y,y') + T_2(x,y,y')y''$ is a total derivative, go to step 5. If this is not the case, stop the algorithm.

4. Compute (2.6.23) as a candidate for μ and go to step 6.

5. Compute (2.6.26) and (2.6.27) as candidates for μ and go to step 6.

6. Compute

$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}$$

for each candidate for μ found in the previous steps. If $A = 0$ for such μ, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

We consider an example, where we can apply Algorithm 2.19 to find an integrating factor.

Example 2.20. The third order ODE

$$y''' = -\frac{12\,y^2\,(y')^2\,y'' + 4\,y^2\,y'\,(y'')^4 + y^2\,(y'')^5 + 8\,x\,y^2\,(y'')^4 + 20\,y\,(y')^4\,y''}{16\,y^3\,(y'')^3 + 16\,x^2\,y^2\,(y'')^3 + 4\,y^2\,y'\,(y'')^3} + \frac{4\,y\,(y')^3 + 12\,x\,y\,(y')^2\,y'' - 4\,(y')^6 - 4\,x\,(y')^4}{16\,y^3\,(y'')^3 + 16\,x^2\,y^2\,(y'')^3 + 4\,y^2\,y'\,(y'')^3}$$

can be written in the form

$$y''' = -\frac{y'\left(\frac{(y')^3}{y} - \frac{(y')^3\,((y')^2+x+y)}{y^2}\right) + \frac{(y')^3}{y}}{(y'')^3\,(4\,x^2+4\,y+y')} - \frac{\frac{2\,(y')^4}{y} + \frac{3\,(y')^2\,((y')^2+x+y)}{y}}{(y'')^2\,(4\,x^2+4\,y+y')} - \frac{2\,x\,y''}{4\,x^2+4\,y+y'} - \frac{y'\,y''}{4\,x^2+4\,y+y'} - \frac{(y'')^2}{4\,(4\,x^2+4\,y+y')},$$

i.e. the right–hand–side is a rational expression in y''. The ODE is not exact, which is proved by Algorithm 2.19 in step 1 using the Euler operator. In step 2, $m = 3$ is determined as the highest power of y'' appearing in the denominator of the right–hand–side of the given ODE. The expressions

$$T_1 = -\frac{2\,x\ + y'}{4\,x^2+4\,y+y'}, \quad T_2 = -\frac{1}{4\,(4\,x^2+4\,y+y')}$$

are determined and, again using the Euler operator, it is verified that $T_1 + y''T_2$ is a total derivative. Hence, the candidate

$$\begin{aligned}\mu(x,y,y',y'') &= \exp\left(-4\,D^{-1}(T_1+y''T_2)\right)(y'')^3\\ &= (4y+y'+4x^2)(y'')^3,\end{aligned}$$

is computed in step 4 of Algorithm 2.19 (constants of integration can be ignored). Finally, in step 6 it is proved that this is indeed an integrating factor for the ODE we started with. ⋄

Again, as in the case of the algorithms of the preceding part of this chapter for finding integrating factors of third order ODEs, the test for exactness in the last step of Algorithm 2.19 is essential. This is demonstrated by the next example.

Example 2.21. The ODE

$$y''' = -\frac{y^2\,y'\,(y'')^3 + y^2\,(y'')^4 + y^2\,(y'')^3 + 3\,y\,(y')^5 + 15\,y\,(y')^4\,y''}{3\,y^3\,(y'')^2 + 3\,x\,y^2\,(y'')^2 + 3\,y^2\,y'\,(y'')^2} + \frac{3\,x\,y\,(y')^3 + 9\,x\,y\,(y')^2\,y'' - 3\,(y')^6 - 3\,x\,(y')^4}{3\,y^3\,(y'')^2 + 3\,x\,y^2\,(y'')^2 + 3\,y^2\,y'\,(y'')^2}$$

can be written as

$$y''' = -\frac{\frac{(y')^3\,((y')^2+x)}{y} - \frac{(y')^4\,((y')^2+x)}{y^2}}{(y'')^2\,(x+y+y')} - \frac{\frac{2\,(y')^4}{y} + \frac{3\,(y')^2\,((y')^2+x)}{y}}{y''\,(x+y+y')} - \frac{y''}{3\,(x+y+y')} - \frac{y'\,y''}{3\,(x+y+y')} - \frac{(y'')^2}{3\,(x+y+y')}.$$

Using Algorithm 2.19, in step 1 it finds that the ODE is not exact. In step 2 the algorithm recognizes that the right–hand–side of the given ODE is a rational expression and computes $m = 2$. Furthermore, $T_1 + y''T_2$ is a total derivative, where

$$T_1 = -\frac{1+y'}{3\,(x+y+y')}, \quad T_2 = -\frac{1}{3\,(x+y+y')}.$$

Hence, a candidate for an integrating factor is computed in step 4 providing

$$\begin{aligned}\mu(x,y,y',y'') &= \exp\left(-3\,D^{-1}(T_1 + y''T_2)\right)(y'')^2\\ &= (y+y'+x)(y'')^2,\end{aligned}$$

but in step 6 it turns out that multiplication of the given ODE with μ does not provide an exact equation (constants of integration can be ignored). Hence, Algorithm 2.19 fails to compute an integrating factor.

The reason, why the algorithm fails, is in principle the same, which makes Algorithms 2.7, 2.9, 2.11 and 2.15 fail (see also Remark 2.17 and Example 2.18): the "constant" of integration $G = G(x,y,y')$ appearing in the class of ODEs (2.6.22) treated above must contribute a total derivative. The concrete ODE considered here is a rational expression in y'', but the expressions, where y'' appears in the denominator, are not of the form

$$-\frac{G_x + y'G_y}{(y'')^2\,(x+y+y')} - \frac{G_{y'}}{y''\,(x+y+y')}$$

for some function $G = G(x,y,y')$. $\diamond$

The case $\mu = f(x,y,y')\frac{1}{(y'')^m}$, $m \in \mathbb{N}$, $m \geq 3$. The partial integrals appearing in (2.6.2) are now given by

$$\int \mu\,dy'' = \frac{1}{1-m} f \frac{1}{(y'')^{m-1}}, \qquad \left(\int \mu\,dy''\right)_x = \frac{1}{1-m} f_x \frac{1}{(y'')^{m-1}},$$
$$\left(\int \mu\,dy''\right)_y = \frac{1}{1-m} f_y \frac{1}{(y'')^{m-1}}, \qquad \left(\int \mu\,dy''\right)_{y'} = \frac{1}{1-m} f_{y'} \frac{1}{(y'')^{m-1}}.$$

According to (2.6.2), the most general class of third order ODEs admitting an integrating factor of the considered form reads

$$y''' = -\frac{G_{y'}}{f}(y'')^{m+1} - \frac{G_x + y'G_y}{f}(y'')^m - \frac{1}{1-m}\left(\frac{f_x}{f}y'' + \frac{f_y}{f}y'y'' + \frac{f_{y'}}{f}(y'')^2\right), \tag{2.6.28}$$

where we first assume $G_{y'} \neq 0$ and $G_x + y'G_y \neq 0$. The right–hand–side of (2.6.28) is a polynomial expression in y''. Its degree in y'' provides a candidate for $m+1$, i.e. we take the degree with respect to y'' and subtract 1 to obtain a candidate for m. In the next step, we define T_1 as the coefficient of y''. Finally, we define T_2 as the coefficient of $(y'')^2$. This provides

$$T_1 = -\frac{1}{1-m}\Big(\frac{f_x}{f} + \frac{f_y}{f}y'\Big), \qquad\qquad T_2 = -\frac{1}{1-m}\frac{f_{y'}}{f}.$$

If $T_1 + y''T_2$ is a total derivative[21], then

$$\mu = \mu(x,y,y',y'') = \exp\big((m-1)\, D^{-1}(T_1 + y''T_2)\big)\,\frac{1}{(y'')^m} \tag{2.6.29}$$

is a candidate for an integrating factor, where the definition for $D^{-1}(T_1+y''T_2)$ is given in (2.6.24).

If either $G_{y'} = 0$ or $G_x + y'G_y = 0$, then (2.6.28) reads

$$y''' = \frac{H(x,y,y')}{f}(y'')^l - \frac{1}{1-m}\Big(\frac{f_x}{f}y'' + \frac{f_y}{f}y'y'' + \frac{f_{y'}}{f}(y'')^2\Big) \tag{2.6.30}$$

for some function H in its arguments and some $l \in \mathbb{N}$, $l \geq 3$. From (2.6.30) we obtain two candidates for m, since we could have $l = m+1$ or $l = m$. This gives the two candidates for an integrating factor

$$\mu_1 = \mu_1(x,y,y',y'') = \exp\big((l-2)\, D^{-1}(T_1 + y''T_2)\big)\,\frac{1}{(y'')^{l-1}}, \tag{2.6.31}$$

and

$$\mu_2 = \mu_2(x,y,y',y'') = \exp\big((l-1)\, D^{-1}(T_1 + y''T_2)\big)\,\frac{1}{(y'')^{l}}. \tag{2.6.32}$$

Summing up these results, we obtain:

Algorithm 2.22. (Computing integrating factors of the form $\mu = f(x,y,y')\frac{1}{(y'')^m}$, $m \geq 3$, of third order ODEs) Given a third order ODE $y''' = \Phi(x,y,y',y'')$ proceed as follows:

[21]As in the discussion of the former case $\mu = f(x,y,y')\,(y'')^m$, these definitions still make sense, if $T_1 = 0$ or $T_2 = 0$. See also the remark in footnote 19 and Example 2.25 below.

1. Check the ODE for exactness, i.e. set
$$\Psi(x,y,y',y'',y''') := y''' - \Phi(x,y,y',y'')$$
and compute
$$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$
If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x,y,y',y'')$ is a polynomial expression in y'' of the form
$$H_1(x,y,y')(y'')^{m+1} + H_2(x,y,y')(y'')^m + T_1(x,y,y')y'' + T_2(x,y,y')(y'')^2,$$
where $H_1 \neq 0$, $H_2 \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments. If this is the case and
$$(T_1+T_2y'')_y - \frac{d}{dx}(T_1+T_2y'')_{y'} + \frac{d^2}{dx^2}(T_1+T_2y'')_{y''} - \frac{d^3}{dx^3}(T_1+T_2y'')_{y'''} = 0,$$
i.e. $T_1(x,y,y') + T_2(x,y,y')y''$ is a total derivative, go to step 4.

3. Check, whether $\Phi(x,y,y',y'')$ is a polynomial expression in y'' of the form
$$H(x,y,y')(y'')^l + T_1(x,y,y')y'' + T_2(x,y,y')(y'')^2,$$
where $H \neq 0$, T_1 and T_2 can be arbitrary functions of their arguments. If this is the case and
$$(T_1+T_2y'')_y - \frac{d}{dx}(T_1+T_2y'')_{y'} + \frac{d^2}{dx^2}(T_1+T_2y'')_{y''} - \frac{d^3}{dx^3}(T_1+T_2y'')_{y'''} = 0,$$
i.e. $T_1(x,y,y') + T_2(x,y,y')y''$ is a total derivative, go to step 5.

4. Compute (2.6.29) as a candidate for μ and go to step 6.

5. Compute (2.6.31) and (2.6.32) as candidates for μ and go to step 6.

6. Compute
$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}$$
for each candidate for μ found in the previous steps. If $A = 0$ for such μ, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. □

Note that we assumed $m \geq 3$, since in the case $m = 2$, the general class of ODEs (2.6.28) reads

$$y''' = -\frac{G_{y'}}{f}(y'')^3 - \frac{G_x + y'G_y}{f}(y'')^2 - \frac{1}{1-m}\Big(\frac{f_x}{f}y'' + \frac{f_y}{f}y'y'' + \frac{f_{y'}}{f}(y'')^2\Big)$$

and, hence, the definition of T_2 as the coefficient of $(y'')^2$ would also involve $\frac{G_x+y'G_y}{f}$, which makes the situation more complicated to find out the function f by means of pattern matching methods as used in Algorithm 2.22. The case $\mu = f(x, y, y')\frac{1}{y''}$ is treated separately below. Before we come to treat that case, we first consider two examples.

Example 2.23. In step 1 Algorithm 2.22 recognizes that the ODE

$$\begin{aligned} y''' =& \frac{3\,x^2\,y^2\,y'' - 9\,x\,y\,(y')^2\,(y'')^5 + 3\,x\,(y')^4\,(y'')^4 - 9\,y^2\,(y')^2\,(y'')^5}{9\,y^2\,y' + 3\,x^3\,y^2 + 3\,y^3} + \\ & \frac{y^2\,y'\,y'' + 3\,y^2\,(y'')^2 - 15\,y\,(y')^4\,(y'')^5 - 3\,y\,(y')^3\,(y'')^4 + 3\,(y')^6\,(y'')^4}{9\,y^2\,y' + 3\,x^3\,y^2 + 3\,y^3} \end{aligned}$$

is not exact. Writing the ODE in the form

$$\begin{aligned} y''' = & -\frac{\frac{2\,(y')^4}{y} + \frac{3\,(y')^2\,((y')^2+x+y)}{y}}{x^3 + y + 3\,y'}(y'')^5 - \frac{y'\,\big(\frac{(y')^3}{y} - \frac{(y')^3\,((y')^2+x+y)}{y^2}\big) + \frac{(y')^3}{y}}{x^3 + y + 3\,y'}(y'')^4 + \\ & \frac{x^2 + y'}{x^3 + y + 3\,y'}\,y'' + \frac{1}{x^3 + y + 3\,y'}\,(y'')^2, \end{aligned}$$

it follows that the right–hand–side is a polynomial expression in y''. The algorithm computes $m = 4$ and proves that $T_1 + y''T_2$ is a total derivative, where

$$T_1 = \frac{x^2 + y'}{x^3 + y + 3\,y'}, \quad T_2 = \frac{1}{x^3 + y + 3\,y'}.$$

Hence, in step 4 a candidate for an integrating follows to be

$$\begin{aligned} \mu(x, y, y', y'') &= \exp\big(3\,D^{-1}(T_1 + y''T_2)\big)\,\frac{1}{(y'')^4} \\ &= (x^3 + y + 3\,y')\frac{1}{(y'')^4}, \end{aligned}$$

for which it is proved in the framework of step 6 that it is indeed an integrating factor for the given ODE (constants of integration in the computation of $\mu(x, y', y'')$ can be ignored). ⋄

As for Algorithm 2.19 we note that the test for exactness in the last step of the algorithm is essential, since multiplication of the ODE with the candidate computed may not always provide a total derivative.

Example 2.24. The ODE

$$\begin{aligned} y''' = & \frac{3\,x^2\,y^2\,y'' - 6\,x\,y\,(y')^2\,(y'')^4 + 2\,x\,(y')^4\,(y'')^3 + 2\,y^3\,y'\,y''}{2\,x^3\,y^2 + 2\,y^4 + 2\,y^2\,(y')^4} + \\ & \frac{4\,y^2\,(y')^3\,(y'')^2 - 6\,y^2\,(y')^2\,(y'')^4 - 10\,y\,(y')^4\,(y'')^4 + 2\,(y')^6\,(y'')^3}{2\,x^3\,y^2 + 2\,y^4 + 2\,y^2\,(y')^4} \end{aligned}$$

can be written in the form

$$\begin{aligned} y''' = & -\frac{\frac{2\,(y')^4}{y} + \frac{3\,(y')^2\,((y')^2+x+y)}{y}}{x^3 + y^2 + (y')^4}\,(y'')^4 - \frac{y'\,\left(\frac{(y')^3}{y} - \frac{(y')^3\,((y')^2+x+y)}{y^2}\right)}{x^3 + y^2 + (y')^4}\,(y'')^3 + \\ & \frac{3\,x^2\, + y\,y'}{2\,(x^3 + y^2 + (y')^4)}y'' + \frac{2\,(y')^3}{x^3 + y^2 + (y')^4}\,(y'')^2. \end{aligned}$$

It is not exact and the right–hand–side is a polynomial expression in y''. In step 2, Algorithm 2.22 computes $m = 3$ and verifies that $T_1 + y''T_2$ is a total derivative, where

$$T_1 = \frac{3\,x^2\, + y\,y'}{2\,(x^3 + y^2 + (y')^4)}, \quad T_2 = \frac{2\,(y')^3}{x^3 + y^2 + (y')^4}.$$

Step 4 then provides the candidate

$$\begin{aligned} \mu(x, y, y', y'') &= \exp\left(2\,D^{-1}(T_1 + y''T_2)\right)\frac{1}{(y'')^3} \\ &= (x^3 + y^2 + (y')^4)\frac{1}{(y'')^3}, \end{aligned}$$

(constants of integration in the computation of $\mu(x, y', y'')$ can be ignored), but again, as in Example 2.21, multiplication of the given ODE with this candidate does not provide an exact ODE[22]. ⋄

The following example demonstrates on the one hand that Algorithm 2.22 also works in the case that the expressions T_1 and T_2 introduced above are equal to zero. On the other hand it gives a hint that also complete classes of (i.e. ODEs involving further symbolic arguments than only the independent variable x, the dependent variable $y(x)$ and its derivatives with respect to x) can be treated.

[22]See the last part of Example 2.21 for remarks on the reason for this failure.

Example 2.25. Consider the class of ODEs of the form

$$y''' = \frac{3y' + a}{(y')^2 + 1}(y'')^k,$$

where $a \in \mathbb{R}$ and $k \in \mathbb{N}$, $k \geq 4$. This class of ODEs is a generalized version of ODE 7.12 in [45] on page 603. Applying Algorithm 2.22 to this ODE, we find $m = k$ or $m = k-1$ and $T_1 = T_2 = 0$. Hence, step 4 provides the two candidates

$$\begin{aligned} \mu_1(x, y, y', y'') &= \exp\big((k-2)\, D^{-1}(T_1 + y''T_2)\big)\, \frac{1}{(y'')^{k-1}} \\ &= \frac{1}{(y'')^{k-1}} \end{aligned}$$

and

$$\begin{aligned} \mu_2(x, y, y', y'') &= \exp\big((k-1)\, D^{-1}(T_1 + y''T_2)\big)\, \frac{1}{(y'')^{k}} \\ &= \frac{1}{(y'')^{k}}, \end{aligned}$$

where we evaluated $D^{-1}(T_1 + y''T_2)$ to 0 to avoid constants of integration. Indeed, multiplication of the ODE by μ_1 provides an exact equation, i.e. Algorithm 2.22 succeeds in finding an integrating factor for this class of ODEs. The candidate μ_2 is not an integrating, as one can easily verify by using the Euler operator as done in the last step of Algorithm 2.22. $\diamond$

The case $\mu = f(x, y, y')\frac{1}{y''}$. The case $\mu = f(x, y, y')\frac{1}{(y'')^m}$, $m \in \mathbb{N}$, $m \geq 3$. Under the assumption on the form of the integrating factor, the partial integrals appearing in (2.6.2) are given by

$$\begin{aligned} \int \mu\, dy'' &= f \ln(y''), & \Big(\int \mu\, dy''\Big)_x &= f_x \ln(y''), \\ \Big(\int \mu\, dy''\Big)_y &= f_y \ln(y''), & \Big(\int \mu\, dy''\Big)_{y'} &= f_{y'} \ln(y''). \end{aligned}$$

The most general class of third order ODEs (2.6.2) reduces to

$$y''' = -\frac{G_x + y'G_y}{f}y'' - \frac{G_{y'}}{f}(y'')^2 - \Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \frac{f_{y'}}{f}y''\Big)y'' \ln(y''). \qquad (2.6.33)$$

The right–hand–side is a linear polynomial in $\ln(y'')$, where we assume for the further discussion that $\frac{d}{dx}f \neq 0$, since otherwise (2.6.33) reduces to the class of ODEs

$$y''' = -\frac{G_x + y'G_y}{f}y'' - \frac{G_{y'}}{f}(y'')^2,$$

for which one can easily check using the Euler operator, whether $\frac{1}{y''}$ is an integrating factor[23].

Taking the coefficients of $\ln(y'')$ in the right–hand-side of (2.6.33) provides

$$-\frac{f_x}{f}y'' - \frac{f_y}{f}y'y'' - \frac{f_{y'}}{f}(y'')^2,$$

i.e. the resulting coefficients must be a quadratic polynomial in y''. Define T_1 as the negative coefficient of y'' and T_2 as the negative coefficient of $(y'')^2$, i.e.

$$T_1 = \frac{f_x}{f} + \frac{f_y}{f}y', \qquad T_2 = \frac{f_{y'}}{f}.$$

Since $T_1 + y''T_2$ is the total derivative of $\ln(f)$ with respect to x, a candidate for an integrating factor is found to be

$$\mu = \mu(x, y, y', y'') = \frac{1}{y''}\exp\left(D^{-1}(T_1 + y''T_2)\right), \tag{2.6.34}$$

where the definition for $D^{-1}(T_1 + y''T_2)$ is given in (2.6.24).

We obtain:

Algorithm 2.26. (Computing integrating factors of the form $\mu = f(x, y, y')\frac{1}{y''}$) Given a third order ODE $y''' = \Phi(x, y, y', y'')$ proceed as follows:

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', y'', y''') := y''' - \Phi(x, y, y', y'')$$

   and compute

   $$A := \Psi_y - \frac{d}{dx}\Psi_{y'} + \frac{d^2}{dx^2}\Psi_{y''} - \frac{d^3}{dx^3}\Psi_{y'''}.$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

[23]Note that in this situation f is a constant. Hence, one can check, whether the ODE is a quadratic polynomial in y'' and whether $-\frac{G_x+y'G_y+G_{y'}y''}{f}$ is a total derivative using the Euler operator. If this is the case, $\frac{1}{y''}$ is indeed an integrating factor.

2. Check, whether $\Phi(x, y, y', y'')$ is a linear polynomial expression in $\ln(y'')$ of the form
$$H_1(x, y, y', y'') + H_2(x, y, y', y'')\,\ln(y''),$$
where H_1 and $H_2 \neq 0$ can be arbitrary functions in their arguments. If this is not the case, stop the algorithm.

3. Check, whether $H_2(x, y, y', y'')$ is a quadratic polynomial in y'' of the form
$$S_1(x, y, y')y'' + S_2(x, y, y')(y'')^2,$$
where S_1 and S_2 are arbitrary functions of their arguments, and
$$(S_1+y''S_2)_y - \frac{d}{dx}(S_1+y''S_2)_{y'} + \frac{d^2}{dx^2}(S_1+y''S_2)_{y''} - \frac{d^3}{dx^3}(S_1+y''S_2)_{y'''} = 0,$$
i.e. if $S_1 + y''S_2$ is a total derivative. If this is not the case, then stop the algorithm.

4. Define $T_1 = -S_1$ and $T_2 = -S_2$ and compute (2.6.34) as a candidate for μ.

5. Compute
$$A := (\mu\,\Psi)_y - \frac{d}{dx}(\mu\,\Psi)_{y'} + \frac{d^2}{dx^2}(\mu\,\Psi)_{y''} - \frac{d^3}{dx^3}(\mu\,\Psi)_{y'''}.$$
If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

We apply Algorithm 2.26 in two concrete examples:

Example 2.27. The ODE

$$y''' = -\frac{3\,x^4\,y\,(y')^2\,(y'')^2 - x^4\,(y')^4\,y'' - x^3\,(y')^6\,y'' - 2\,y^2\,y''\,\ln(y'')}{x^3\,y^4\,(y')^3 + x\,y^2} + \frac{5\,x^3\,y\,(y')^4\,(y'')^2 + 3\,x^3\,y^2\,(y')^2\,(y'')^2 + x^3\,y\,(y')^3\,y''}{x^3\,y^4\,(y')^3 + x\,y^2} + \frac{2\,x^3\,y^3\,(y')^4\,y''\,\ln(y'') + 3\,x^3\,y^4\,(y')^2\,(y'')^2\,\ln(y'')}{x^3\,y^4\,(y')^3 + x\,y^2}$$

is not exact. Writing the ODE in the form

$$y'' = -\frac{y'\left(\frac{(y')^3}{y} - \frac{(y')^3\left((y')^2+x+y\right)}{y^2}\right) + \frac{(y')^3}{y}}{y^2\,(y')^3 + \frac{1}{x^2}} y'' - \frac{\frac{2\,(y')^4}{y} + \frac{3\,(y')^2\left((y')^2+x+y\right)}{y}}{y^2\,(y')^3 + \frac{1}{x^2}} (y'')^2 - \left(\frac{2\,y\,(y')^4\,y''}{y^2\,(y')^3 + \frac{1}{x^2}} - \frac{2\,y''}{x^3\,y^2\,(y')^3 + x} + \frac{3\,y^2\,(y')^2\,(y'')^2}{y^2\,(y')^3 + \frac{1}{x^2}}\right) \ln\left(y''\right),$$

we note that the right–hand–side is a linear polynomial in $\ln(y'')$. After recognizing that the given ODE is not exact, Algorithm 2.26 determines

$$H_2 = -\frac{2\,y\,(y')^4\,y''}{y^2\,(y')^3 + \frac{1}{x^2}} + \frac{2\,y''}{x^3\,y^2\,(y')^3 + x} - \frac{3\,y^2\,(y')^2\,(y'')^2}{y^2\,(y')^3 + \frac{1}{x^2}}$$

as the coefficient of $\ln(y'')$. Since H_2 is a quadratic polynomial in y'', the algorithm computes

$$S_1 = -\frac{2\,y\,(y')^4}{y^2\,(y')^3 + \frac{1}{x^2}} + \frac{2}{x^3\,y^2\,(y')^3 + x}, \quad S_2 = -\frac{3\,y^2\,(y')^2}{y^2\,(y')^3 + \frac{1}{x^2}}$$

in step 3 and verifies that $S_1 + y''S_2$ is a total derivative. Finally, step 5 provides the candidate

$$\begin{aligned} \mu(x, y, y', y'') &= \frac{1}{y''} \exp\left(D^{-1}(T_1 + y''T_2)\right) \\ &= \frac{1}{y''}\left(y^2\,(y')^3 + \frac{1}{x^2}\right) \end{aligned}$$

where $T_1 = -S_1$ and $T_2 = -S_2$. In step 6 it is verified that this is indeed an integrating factor for the ODE under consideration. $\diamond$

As in the case of Algorithms 2.19 and 2.22, the test for exactness in step 6 of 2.26 is essential:

Example 2.28. The ODE

$$y''' = \frac{y''\left(y'\left(\frac{(y')^3}{y^3} + \frac{(y')^3\left(x^3+\frac{1}{y}+(y')^3\right)}{y^2}\right) - \frac{(y')^3\left(x^3+\frac{1}{y}+(y')^3\right)}{y}\right)}{x+y+y'} - \frac{(y')^3\,(y'')^2\left(x^3+\frac{1}{y}+(y')^3\right)}{y\,\left(x+y+y'\right)} - \left(\frac{y''+y'y''+(y'')^2}{x+y+y'}\right) \ln\left(y''\right)$$

is not exact and its right–hand–side is a linear polynomial in $\ln(y'')$. The expression

$$H_2 = \frac{y'' + y'y'' + (y'')^2}{x + y + y'}$$

determined in step 2 of the algorithm is a quadratic polynomial in y'' and in step 3 it is verified that $S_1 + y''S_2$ is a total derivative, where

$$S_1 = \frac{1 + y'}{x + y + y'}, \quad S_2 = \frac{1}{x + y + y'}.$$

The candidate obtained in step 4 is computed to be

$$\mu(x, y, y', y'') = (x + y + y')\frac{1}{y''},$$

but this is not an integrating factor for the ODE under consideration[24]. $\diamond$

In the next section we generalize the algorithms presented in this section to be able to treat n-th order ODEs, $n \geq 3$.

2.7 Generalizations to higher order ODEs

In this section we present possible generalizations of the results described in Section 2.6 for classes of higher order ODEs.

Consider a general n-th order ODE of the form

$$y^{(n)} = \Phi(x, y, y', \ldots, y(n-1)), \quad n \geq 3. \tag{2.7.1}$$

By Theorem 2.5, the most general class of n-th order ODEs of the form (2.7.1) having an integrating factor $\mu = \mu(x, y, y', \ldots, y^{(n-1)})$ is

$$\begin{aligned} y^{(n)} = -\frac{1}{\mu}\Big[\quad & \left(G + \int \mu\, dy^{(n-1)}\right)_x + y'\left(G + \int \mu\, dy^{(n-1)}\right)_y + \\ & y''\left(G + \int \mu\, dy^{(n-1)}\right)_{y'} + \ldots + y^{(n-1)}\left(G + \int \mu\, dy^{(n-1)}\right)_{y^{(n-2)}}\Big] \end{aligned}$$

for an arbitrary function $G = G(x, y, y', \ldots, y^{(n-1)})$ in its argument. In the following we consider the integrating factors:

[24] See the last part of Example 2.21 for remarks on the reason for this failure.

- $\mu = \mu(y^{(i)}, y^{(n-2)})$, $0 \le i \le n-3$,
- $\mu = \mu(x, y^{(n-2)})$,
- $\mu = \mu(y^{(i)}, y^{(j)})$, $0 \le i < j \le n-3$,
- $\mu = f(x, y, y', \dots, y^{(n-2)})(y^{(n-1)})^m$, $m \in \mathbb{Z} \setminus \{-2, 0\}$.

We do not present further examples of concrete ODEs, where the algorithms stated in this section are applied. We refer to the examples in the preceding section, where the corresponding algorithms for third order ODEs have been applied, instead.

2.7.1 Integrating factors $\mu(y^{(i)}, y^{(n-2)})$, $0 \le i \le n-3$.

As in the preceding sections we first compute the most general class of n-th order ODEs admitting an integrating factor of the prescribed form. This class is

$$\begin{aligned} y^{(n)} = &-\frac{\mu_{y^{(n-2)}}}{\mu}\,(y^{(n-1)})^2 - \frac{\mu_{y^{(i)}}\,y^{(i+1)} + G_{y^{(n-2)}}}{\mu}\,y^{(n-1)} \\ &- \frac{G_x + y'\,G_y + y''\,G_{y'} + \ldots + y^{(n-2)}\,G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

Since μ and G do not depend on $y^{(n-1)}$, this ODE is a quadratic polynomial in $y^{(n-1)}$. Similar to the way of proceeding in the former section, we assume that $G = G(x, y, y', \dots, y^{(n-2)})$ does not depend on $y^{(n-2)}$. This provides

$$\begin{aligned} y^{(n)} = &-\frac{\mu_{y^{(n-2)}}}{\mu}\,(y^{(n-1)})^2 - \frac{\mu_{y^{(i)}}\,y^{(i+1)}}{\mu}\,y^{(n-1)} \\ &- \frac{G_x + y'\,G_y + y''\,G_{y'} + \ldots + y^{(n-2)}\,G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

We define

$$a(y^{(i)}, y^{(n-2)}) := -\frac{\mu_{y^{(n-2)}}}{\mu}, \qquad b(y^{(i)}, y^{(i+1)}, y^{(n-2)}) := -\frac{\mu_{y^{(i)}}\,y^{(i+1)}}{\mu}, \tag{2.7.2}$$

i.e. the coefficients of $(y^{(n-1)})^2$ and $y^{(n-1)}$ of the right–hand–side of the ODE viewed as a polynomial in $y^{(n-1)}$. Note that $b(y^{(i)}, y^{(i+1)}, y^{(n-2)})$ itself can be viewed as linear polynomial in $y^{(i+1)}$ without constant term with respect to $y^{(i+1)}$, since μ does not depend on $y^{(i+1)}$ by assumption. The expressions $a(y^{(i)}, y^{(n-2)})$ and $b(y^{(i)}, y^{(i+1)}, y^{(n-2)})$ must satisfy

$$a_{y^{(i)}}(y^{(i)}, y^{(n-2)}) = \Big(\frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i+1)}}\Big)_{y^{(n-2)}}. \tag{2.7.3}$$

We choose

$$\int a(y^{(i)}, y^{(n-2)})\, dy^{(n-2)} + \frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i+1)}}\, dy^{(i)}$$

to denote the function in $y^{(i)}$ and $y^{(n-2)}$ (note, that in fact $\frac{b(y^{(i)},y^{(i+1)},y^{(n-2)})}{y^{(i+1)}}$ does not depend on $y^{(i+1)}$, since $b(y^{(i)}, y^{(i+1)}, y^{(n-2)})$ is a linear polynomial in $y^{(i+1)}$ without constant term with respect to $y^{(i+1)}$ by assumption) uniquely determined up to the addition of a constant, such that its partial derivatives with respect to $y^{(n-2)}$ and $y^{(i+1)}$ fulfill

$$\Big(\int a(y^{(i)}, y^{(n-2)})\, dy^{(n-2)} + \frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i+1)}}\, dy^{(i)}\Big)_{y^{(n-2)}} = a(y^{(i)}, y^{(n-2)})$$

and

$$\Big(\int a(y^{(i)}, y^{(n-2)})\, dy^{(n-2)} + \frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i+1)}}\, dy^{(i)}\Big)_{y^{(i)}} = \frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i)}}.$$

Then a candidate for an integrating factor can be written in the form

$$\mu(y^{(i)}, y^{(n-2)}) = \exp\Big(-\int a(y^{(i)}, y^{(n-2)})\, dy^{(n-2)} + \frac{b(y^{(i)}, y^{(i+1)}, y^{(n-2)})}{y^{(i+1)}}\, dy^{(i)}\Big). \tag{2.7.4}$$

As a generalization of Algorithm 2.11 we obtain:

Algorithm 2.29. (Computing integrating factors of the form $\mu = \mu(y^{(i)}, y^{(n-2)})$, $0 \le i \le n-3$, of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', \ldots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})$$

   and compute

   $$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial \Psi(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}}\Big).$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a quadratic polynomial in $y^{(n-1)}$. If not, stop the algorithm.

3. Check, whether the coefficients of $(y^{(n-1)})^2$ and $y^{(n-1)}$ in $\Phi(x, y, y', \ldots, y^{(n-1)})$ viewed as a polynomial in $y^{(n-1)}$ only depend in $y^{(i)}$, $y^{(i+1)}$ and $y^{(n-2)}$. If not, stop the algorithm.

4. Determine the functions $a(y^{(i)}, y^{(n-2)})$ and $b(y^{(i)}, y^{(i+1)}, y^{(n-2)})$ as in (2.7.2) and check if $b(y^{(i)}, y^{(i+1)}, y^{(n-2)})$ is a linear polynomial in $y^{(i+1)}$ with constant term 0 (constant with respect to $y^{(i+1)}$). If not, stop the algorithm.

5. Check, if (2.7.3) is fulfilled. If not, stop the algorithm.

6. Compute (2.7.4) as a candidate for μ.

7. Compute

$$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i} \Big(\frac{\partial\big(\mu\, \Psi(x, y, y', \ldots, y^{(n)})\big)}{\partial y^{(i)}} \Big).$$

If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

2.7.2 Integrating factors $\mu(x, y^{(n-2)})$.

By Theorem 2.5 the most general class of n-th order ODEs admitting an integrating of the form $\mu = \mu(x, y^{(n-2)})$ is

$$\begin{aligned} y^{(n)} = &- \frac{\mu_{y^{(n-2)}}}{\mu} (y^{(n-1)})^2 - \frac{\mu_x + G_{y^{(n-2)}}}{\mu} y^{(n-1)} \\ &- \frac{G_x + y'\, G_y + y''\, G_{y'} + \ldots + y^{(n-2)}\, G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

The assumption that $G = G(x, y, y', \ldots, y^{(n-2)})$ does not depend on $y^{(n-2)}$ provides the general class of ODEs

$$\begin{aligned} y^{(n)} = &- \frac{\mu_{y^{(n-2)}}}{\mu} (y^{(n-1)})^2 - \frac{\mu_x}{\mu} y^{(n-1)} \\ &- \frac{G_x + y'\, G_y + y''\, G_{y'} + \ldots + y^{(n-2)}\, G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

Since μ and G do not depend on $y^{(n-1)}$, this ODE can be viewed as a quadratic polynomial in $y^{(n-1)}$. We read–off the coefficients

$$a(x, y^{(n-2)}) := -\frac{\mu_{y^{(n-2)}}}{\mu}, \qquad b(x, y^{(n-2)}) := -\frac{\mu_x}{\mu}. \tag{2.7.5}$$

For $a(x, y^{(n-2)})$ and $b(x, y^{(n-2)})$ the symmetry of second derivatives must hold, i.e. we have

$$a_x(x, y^{(n-2)}) = b_{y^{(n-2)}}(x, y^{(n-2)}). \tag{2.7.6}$$

If we denote by

$$\int a(x, y^{(n-2)})\, dy^{(n-2)} + b(x, y^{(n-2)})\, dx$$

the function in x and $y^{(n-2)}$ uniquely determined up to the addition of a constant of integration, whose partial derivatives with respect to $y^{(n-2)}$ and x satisfy

$$\Big(\int a(x, y^{(n-2)})\, dy^{(n-2)} + b(x, y^{(n-2)})\, dx\Big)_{y^{(n-2)}} = a(x, y^{(n-2)})$$

and

$$\Big(\int a(x, y^{(n-2)})\, dy^{(n-2)} + b(x, y^{(n-2)})\, dx\Big)_{x} = b(x, y^{(n-2)}),$$

a candidate for an integrating factor is obtained to be

$$\mu(x, y^{(n-2)}) = \exp\Big(-\int a(x, y^{(n-2)})\, dy^{(n-2)} + b(x, y^{(n-2)})\, dx\Big). \tag{2.7.7}$$

We summarize our results in the following algorithm, which is a generalization of Algorithm 2.9.

Algorithm 2.30. (Computing integrating factors of the form $\mu = \mu(x, y^{(n-2)})$ of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \dots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', \dots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \dots, y^{(n-1)})$$

and compute

$$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial \Psi(x, y, y', \dots, y^{(n)})}{\partial y^{(i)}}\Big).$$

If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', \dots, y^{(n-1)})$ is a quadratic polynomial in $y^{(n-1)}$. If not, stop the algorithm.

3. Check, whether the coefficients of $(y^{(n-1)})^2$ and $y^{(n-1)}$ in $\Phi(x, y, y', \dots, y^{(n-1)})$ viewed as a polynomial in $y^{(n-1)}$ only depend on x and $y^{(n-2)}$. If not, stop the algorithm.

4. Determine the functions $a(x, y^{(n-2)})$ and $b(x, y^{(n-2)})$ as in (2.7.5) and check if (2.7.6) is fulfilled. If not, stop the algorithm.

5. Compute (2.7.7) as a candidate for μ.

6. Compute

$$A := \sum_{i=0}^{n} (-1)^i \, \frac{d^i}{dx^i} \Big(\frac{\partial\big(\mu \, \Psi(x, y, y', \dots, y^{(n)})\big)}{\partial y^{(i)}} \Big).$$

   If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

2.7.3 Integrating factors $\mu(y^{(i)}, y^{(j)})$, $0 \le i < j \le n-3$.

By Theorem 2.5, the most general class of n-th order ODEs admitting an integrating of the form $\mu = \mu(y^{(i)}, y^{(j)})$ is

$$\begin{aligned} y^{(n)} = &- \frac{\mu_{y^{(i)}} \, y^{(i+1)} + \mu_{y^{(j)}} \, y^{(j+1)} + G_{y^{(n-2)}}}{\mu} \, y^{(n-1)} \\ &- \frac{G_x + y' \, G_y + y'' \, G_{y'} + \ldots + y^{(n-2)} \, G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

Assuming that $G = G(x, y, y', \dots, y^{(n-2)})$ does not depend on $y^{(n-2)}$ provides the general class of ODEs

$$\begin{aligned} y^{(n)} = &- \frac{\mu_{y^{(i)}} \, y^{(i+1)} + \mu_{y^{(j)}} \, y^{(j+1)}}{\mu} \, y^{(n-1)} \\ &- \frac{G_x + y' \, G_y + y'' \, G_{y'} + \ldots + y^{(n-2)} \, G_{y^{(n-3)}}}{\mu}. \end{aligned}$$

This ODE can be viewed as a linear polynomial in $y^{(n-1)}$, since μ and G do not depend on $y^{(n-1)}$. Note that the case $\mu = \mu(y^{(i)}, y^{(n-2)})$ has already been discussed in the previous Section 2.7.2. We have to find a way to compute a candidate for an integrating factor from the coefficient

$$-\frac{\mu_{y^{(i)}}\, y^{(i+1)} + \mu_{y^{(j)}}\, y^{(j+1)}}{\mu} \tag{2.7.8}$$

of $y^{(n-1)}$. Our idea to find a representation for μ is based on the fact that we can determine the coefficients of $y^{(i+1)}$ and $y^{(j+1)}$ in the above expression. This is only possible in general and by simple means, if $i + 1 < j$. In the case $i + 1 = j$, the above expression contains in fact the linear polynomial $-\frac{\mu_{y^{(j)}}}{\mu}\, y^{(j+1)}$, but the term $-\frac{\mu_{y^{(i)}}}{\mu}\, y^{(i+1)}$ may not be a linear polynomial in $y^{(i+1)}$, since the logarithmic derivative $\frac{\mu_{y^{(i)}}}{\mu}$ in general depends on $y^{(i+1)} = y^{(j)}$, too. Hence, we assume $i + 1 < j$ in the following.

In this case, (2.7.8) is the sum of the two linear polynomials $-\frac{\mu_{y^{(j)}}}{\mu}\, y^{(j+1)}$ and $-\frac{\mu_{y^{(i)}}}{\mu}\, y^{(i+1)}$. We define

$$a(y^{(i)}, y^{(j)}) := -\frac{\mu_{y^{(i)}}}{\mu}, \qquad b(y^{(i)}, y^{(j)}) := -\frac{\mu_{y^{(j)}}}{\mu}. \tag{2.7.9}$$

Then the symmetry condition for second derivatives provides the usual condition for $a(y^{(i)}, y^{(j)})$ and $b(y^{(i)}, y^{(j)})$, which reads

$$a_{y^{(j)}}(y^{(i)}, y^{(j)}) = b_{y^{(i)}}(y^{(i)}, y^{(j)}). \tag{2.7.10}$$

Again, proceeding as in the treatment of the former cases, we denote by

$$\int a(y^{(i)}, y^{(j)})\, dy^{(i)} + b(y^{(i)}, y^{(j)})\, dy^{(j)}$$

the function in $y^{(i)}$ and $y^{(j)}$ uniquely determined up to the addition of a constant of integration, such that its partial derivatives with respect to $y^{(i)}$ and $y^{(j)}$ are given by

$$\Big(\int a(y^{(i)}, y^{(j)})\, dy^{(i)} + b(y^{(i)}, y^{(j)})\, dy^{(j)} \Big)_{y^{(i)}} = a(y^{(i)}, y^{(j)})$$

and

$$\Big(\int a(y^{(i)}, y^{(j)})\, dy^{(i)} + b(y^{(i)}, y^{(j)})\, dy^{(j)} \Big)_{y^{(j)}} = b(y^{(i)}, y^{(j)}).$$

Hence, a candidate for an integrating factor is obtained to be

$$\mu(y^{(i)}, y^{(j)}) = \exp\Big(-\int a(y^{(i)}, y^{(j)})\, dy^{(i)} + b(y^{(i)}, y^{(j)})\, dy^{(j)}\Big). \tag{2.7.11}$$

We summarize our results in the following algorithm, which is a more general version of Algorithm 2.7:

Algorithm 2.31. (Computing integrating factors of the form $\mu = \mu(y^{(i)}, y^{(j)})$, $0 \le i < j \le n-3$, $i+1 < j$, of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \dots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', \dots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \dots, y^{(n-1)})$$

   and compute

   $$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial \Psi(x, y, y', \dots, y^{(n)})}{\partial y^{(i)}}\Big).$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', \dots, y^{(n-1)})$ is a linear polynomial in $y^{(n-1)}$. If not, stop the algorithm.

3. Check, whether the coefficient of $y^{(n-1)}$ in $\Phi(x, y, y', \dots, y^{(n-1)})$ viewed as a linear polynomial in $y^{(n-1)}$ is itself the sum of two linear polynomials in $y^{(i+1)}$ and $y^{(j+1)}$, $i+1 < j$, with constant terms 0. If not, stop the algorithm.

4. Determine the functions $a(y^{(i)}, y^{(j)})$ and $b(y^{(i)}, y^{(j)})$ as in (2.7.9) and check, if (2.7.10) is fulfilled. If not, stop the algorithm.

5. Compute (2.7.11) as a candidate for μ.

6. Compute

   $$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\big(\mu\, \Psi(x, y, y', \dots, y^{(n)})\big)}{\partial y^{(i)}}\Big).$$

   If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. △

Proof. The correctness of the algorithm follows from the preceding discussion. □

2.7.4 Integrating factors $\mu(y^{(n-1)})$.

This paragraph is dedicated to present a heuristic for the case of integrating factors of the form $\mu = \mu(y^{(n-1)})$ for n-th order ODEs. The heuristic is an analogon to Algorithm 2.15 for finding integrating factors of the form $\mu = \mu(y'')$ of third order ODEs. By Theorem 2.5, the most general class of n-th order ODEs admitting an integrating factor $\mu = \mu(y^{(n-1)})$ is

$$y^{(n)} = -\frac{G_{y^{(n-2)}}}{\mu}\, y^{(n-1)} - \frac{G_x + y'\, G_y + y''\, G_{y'} + \ldots + y^{(n-2)}\, G_{y^{(n-3)}}}{\mu}. \quad (2.7.12)$$

As in the case of integrating factors $\mu = \mu(y'')$ for third order ODEs, we assume that $\mu(0) \neq 0$. If this is the case we can conclude that $\frac{1}{\mu}$ is a proper factor of the right–hand–side of (2.7.12), since $G = G(x, y, y', \ldots, y^{(n-2)})$ does not depend on $y^{(n-1)}$. Thus, if $\mu(0) \neq 0$, we can try to factor the right–hand–side of (2.7.12) and check, if there is a factor $g(y^{(n-1)})$. If this is the case, we check, whether the right–hand–side divided by $g(y^{(n-1)})$ is a linear polynomial in $y^{(n-1)}$ and a total derivative. In this situation $\frac{1}{g(y^{(n-1)})}$ is a candidate for an integrating factor of (2.7.12).

Of course, one should first try to find an integrating factor of a given n-th order ODE by using Algorithms 2.29, 2.30 and 2.31, since the following algorithm is more a heuristic because of the factorization of an arbitrary expression in step 2 (as mentioned in the context of Algorithm 2.15 for third order ODEs, the factorization of arbitrary expressions requires special heuristics, which are offered by most of the current general purpose computer algebra systems such as Maple or MuPAD).

Algorithm 2.32. (Computing integrating factors of the form $\mu = \mu(y^{(n-1)})$ of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$ proceed as follows.

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', \ldots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})$$

   and compute

   $$A := \sum_{i=0}^{n} (-1)^i \, \frac{d^i}{dx^i} \Big(\frac{\partial \Psi(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}} \Big).$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Try to find out, if $\Phi(x, y, y', \ldots, y^{(n-1)})$ contains a factor $g(y^{(n-1)})$ depending only on $y^{(n-1)}$. If this is not the case, stop the algorithm.

3. Let $\Phi(x, y, y', \ldots, y^{(n-1)}) = g(y^{(n-1)})\, h(x, y, y', \ldots, y^{(n-1)})$. Check if $h(x, y, y', \ldots, y^{(n-1)})$ is a linear polynomial in $y^{(n-1)}$ and if

$$\sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial h(x, y, y', \ldots, y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$

i.e. if $h(x, y, y', \ldots, y^{(n-1)})$ is a total derivative. If this is not the case, stop the algorithm.

4. Choose $\mu = \frac{1}{g(y^{(n-1)})}$ as a candidate for an integrating factor of the given ODE.

5. Compute

$$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\big(\mu\, \Psi(x, y, y', \ldots, y^{(n)})\big)}{\partial y^{(i)}}\Big).$$

If $A = 0$, then μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

2.7.5 Integrating factors $f(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^m$.

We close this section with the generalization of Algorithms 2.19, 2.22 and 2.26. As in Section 2.6.5, we first treat the case of integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^m$, $m \in \mathbb{N}$, where f can be an arbitrary function in its arguments.

The case $\mu = f(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^m$, $m \in \mathbb{N}$. By Theorem 2.5, the most general class of n-th order ODEs admitting an integrating factor of the prescribed form reads

$$\begin{aligned} y^{(n)} = &-\frac{G_x + y' G_y + \ldots + y^{(n-2)} G_{y^{(n-3)}}}{f (y^{(n-1)})^m} - \frac{G_{y^{(n-2)}}}{f (y^{(n-1)})^{m-1}} \\ &- \frac{1}{m+1}\Big(\frac{f_x}{f} + \frac{f_y}{f} y' + \ldots + \frac{f_{y^{(n-3)}}}{f} y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f} y^{(n-1)}\Big) y^{(n-1)}. \end{aligned} \tag{2.7.13}$$

Note that $f(y^{(n-1)})^m$ and $f(y^{(n-1)})^{m-1}$ denotes the multiplication of f by $(y^{(n-1)})^m$ and $(y^{(n-1)})^{m-1}$, respectively, and not the application of the function f to $(y^{(n-1)})^m$ and $(y^{(n-1)})^{m-1}$.

If $G_x + y'G_y + \ldots + y^{(n-2)} \neq 0$ and $G_{y^{(n-2)}} \neq 0$, then (2.7.13) is a rational expression in $y^{(n-1)}$. The highest power of $y^{(n-1)}$ appearing in a denominator of the right–hand–side of (2.7.13) gives a candidate for the power of $y^{(n-1)}$ appearing in the integrating factor to be found. The expression

$$-\frac{1}{m+1}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)}$$

is a quadratic polynomial in $y^{(n-1)}$, i.e. we can read–off the two quantities

$$T_1 = -\frac{1}{m+1}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)}\Big), \quad T_2 = -\frac{1}{m+1}\frac{f_{y^{(n-2)}}}{f},$$

i.e. T_1 is the coefficient of $y^{(n-1)}$ and T_2 is the coefficient of $(y^{(n-1)})^2$ in the polynomial part of the right–hand–side of (2.7.13)[25].

Since $T_1 + y^{(n-1)}T_2$ is the total derivative of $-\frac{1}{m+1}\ln(f)$, a candidate for an integrating factor is given by

$$\mu = \exp\big(-(m+1)\, D^{-1}(T_1 + y^{(n-1)}T_2)\big)\,(y^{(n-1)})^m, \tag{2.7.14}$$

where by $D^{-1}(T_1+y^{(n-1)}T_2)$ we denote a function $F(x, y, y', \ldots, y^{(n-2)})$ uniquely determined up to the addition of a constant of integration, such that its total derivative with respect to x equals $T_1 + y^{(n-1)}T_2$, i.e.

$$\frac{d}{dx}D^{-1}(T_1 + y^{(n-1)}T_2) = \frac{d}{dx}F(x, y, y', \ldots, y^{(n-2)}) = T_1 + y^{(n-1)}T_2. \tag{2.7.15}$$

If either $G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}} = 0$ or $G_{y^{(n-2)}} = 0$, (2.7.13) reduces to the form

$$\begin{aligned} y^{(n)} = {} & \frac{H(x, y, y', \ldots, y^{(n-2)})}{f(y^{(n-1)})^l} - \\ & \frac{1}{m+1}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)} \end{aligned} \tag{2.7.16}$$

[25] Note that also in the case $m = 1$, these coefficients can be determined. If $m = 1$, then the expression $-\frac{G_{y^{(n-2)}}}{f(y^{(n-1)})^{m-1}}$ in the right–hand–side of (2.7.13) contributes a constant term with respect to $y^{(n-1)}$, but the quadratic polynomial $T_1 y^{(n-1)} + T_2(y^{(n-1)})^2$ by assumption does not have a term constant with respect to $y^{(n-1)}$.

for some non–zero function H in its arguments. From (2.7.16) we find $l = m-1$ in the case $G_x + y' G_y + \ldots + y^{(n-2)} = 0$ and $l = m$ in the case $G_{y^{(n-2)}} = 0$. Since we cannot know a priori with which case we are faced with, we obtain the two candidates

$$\mu_1 = \exp\left(-(l+2)\, D^{-1}(T_1 + y^{(n-1)} T_2)\right) (y^{(n-1)})^{l+1} \tag{2.7.17}$$

and

$$\mu_2 = \exp\left(-(l+1)\, D^{-1}(T_1 + y^{(n-1)} T_2)\right) (y^{(n-1)})^{l} \tag{2.7.18}$$

as possible integrating factors. As in the case of third order ODEs discussed in Section 2.6.5, we exclude the case that $G_x + y' G_y + \ldots + y^{(n-2)} G_{y^{(n-3)}} = 0 = G_{y^{(n-2)}}$, since then (2.7.13) reduces to a quadratic polynomial in $y^{(n-1)}$ and the computation of a candidate for an integrating factor does not seem to be possible without solving any auxiliary ODEs or partial differential equations. Our results are summarized in the following algorithm.

Algorithm 2.33. (Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^m, m \in \mathbb{N}$) of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', \ldots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})$$

and compute

$$A := \sum_{i=0}^{n} (-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial \Psi(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}}\Big).$$

If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a rational expression in $y^{(n-1)}$ of the form

$$\frac{H_1(x, y, y', \ldots, y^{(n-2)})}{(y^{(n-1)})^m} + \frac{H_2(x, y, y', \ldots, y^{(n-2)})}{(y^{(n-1)})^{m-1}} +$$
$$T_1(x, y, y', \ldots, y^{(n-2)}) y^{(n-1)} + T_2(x, y, y', \ldots, y^{(n-2)}) (y^{(n-1)})^2,$$

where $H_1 \neq 0$, $H_2 \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments. If this is the case and

$$\sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial(T_1 + T_2 y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$

i.e. $T_1 + T_2 y^{(n-1)}$ is a total derivative, go to step 4.

3. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a rational expression in $y^{(n-1)}$ of the form

$$\frac{H(x, y, y', \ldots, y^{(n-2)})}{(y^{(n-1)})^l} +$$
$$T_1(x, y, y', \ldots, y^{(n-2)}) y^{(n-1)} + T_2(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^2,$$

where $H \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments and $l \in \mathbb{N}$. If this is the case and

$$\sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial(T_1 + T_2 y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$

i.e. $T_1 + T_2 y^{(n-1)}$ is a total derivative, go to step 5. If this is not the case, stop the algorithm.

4. Compute (2.7.14) as a candidate for μ and go to step 6.

5. Compute (2.7.17) and (2.7.18) as candidates for μ and go to step 6.

6. Compute
$$A := \sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\big(\mu\, \Psi(x, y, y', \ldots, y^{(n)})\big)}{\partial y^{(i)}}\Big)$$
for each candidate for μ found in the previous steps. If $A = 0$ for such a μ, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

Next we generalize Algorithm 2.22.

The case $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{(y^{(n-1)})^m}$, $m \in \mathbb{N}$, $m \geq 3$. The general class of ODEs admitting such an integrating factor is derived from Theorem 2.5 to be

$$\begin{aligned} y^{(n)} &= -\frac{G_{y^{(n-2)}}}{f}(y^{(n-1)})^{m+1} - \frac{G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}}}{f}(y^{(n-1)})^m \\ &\quad - \frac{1}{1-m}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)}. \end{aligned} \tag{2.7.19}$$

The right–hand–side of (2.7.19) is a polynomial expression in $y^{(n-1)}$. If $G_{y^{(n-2)}} \neq 0$ and $G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}} \neq 0$, then a candidate for the power of $y^{(n-1)}$ appearing in the integrating factor to be computed is obtained from the highest power of $y^{(n-1)}$ minus 1. As in the discussion of the former case,

$$-\frac{1}{1-m}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)}$$

is a quadratic polynomial in $y^{(n-1)}$ and we can read–off

$$T_1 = -\frac{1}{1-m}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)}\Big), \quad T_2 = -\frac{1}{1-m}\frac{f_{y^{(n-2)}}}{f},$$

i.e. T_1 is the coefficient of $y^{(n-1)}$ and T_2 is the coefficient of $(y^{(n-1)})^2$. Then $T_1 + y^{(n-1)}T_2$ is the total derivative of $-\frac{1}{1-m}\ln(f)$ and a candidate for an integrating factor is

$$\mu = \exp\big((m-1)\,D^{-1}(T_1 + y^{(n-1)}T_2)\big)\,\frac{1}{(y^{(n-1)})^m}, \tag{2.7.20}$$

where the definition of $D^{-1}(T_1 + y^{(n-1)}T_2)$ is given in (2.7.15).

If either $G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}} = 0$ or $G_{y^{(n-2)}} = 0$, then (2.7.19) reduces to the class of ODEs

$$\begin{aligned} y^{(n)} &= \frac{H(x, y, y', \ldots, y^{(n-2)})}{f}(y^{(n-1)})^l \\ &\quad - \frac{1}{1-m}\Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)} + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)}, \end{aligned} \tag{2.7.21}$$

where $H \neq 0$ is an arbitrary function in its arguments. From the right–hand–side of (2.7.21) we again find two candidates for m, namely l itself and $l-1$. This provides the two candidates

$$\mu_1 = \exp\big((l-2)\, D^{-1}(T_1 + y^{(n-1)}T_2)\big)\, \frac{1}{(y^{(n-1)})^{l-1}} \tag{2.7.22}$$

and

$$\mu_2 = \exp\big((l-1)\, D^{-1}(T_1 + y^{(n-1)}T_2)\big)\, \frac{1}{(y^{(n-1)})^{l}} \tag{2.7.23}$$

for integrating factors. Summing up the results, we can state:

Algorithm 2.34. (Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{(y^{(n-1)})^m}, m \in \mathbb{N}, m \geq 3$) of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

$$\Psi(x, y, y', \ldots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})$$

and compute

$$A := \sum_{i=0}^{n} (-1)^i \, \frac{d^i}{dx^i}\Big(\frac{\partial \Psi(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}}\Big).$$

If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

2. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a polynomial expression in $y^{(n-1)}$ of the form

$$H_1(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^{m+1} + H_2(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^{m} +$$
$$T_1(x, y, y', \ldots, y^{(n-2)})y^{(n-1)} + T_2(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^2,$$

where $H_1 \neq 0$, $H_2 \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments. If this is the case and

$$\sum_{i=0}^{n} (-1)^i \, \frac{d^i}{dx^i}\Big(\frac{\partial (T_1 + T_2 y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$

i.e. $T_1 + T_2 y^{(n-1)}$ is a total derivative, go to step 4.

3. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a polynomial expression in $y^{(n-1)}$ of the form

$$H(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^l + \\ T_1(x, y, y', \ldots, y^{(n-2)})y^{(n-1)} + T_2(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^2,$$

   where $H \neq 0$, T_1 and T_2 can be arbitrary functions in their arguments and $l \in \mathbb{N}$, $l \geq 3$. If this is the case and

$$\sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial(T_1 + T_2 y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$

   i.e. $T_1 + T_2 y^{(n-1)}$ is a total derivative, go to step 5. If this is not the case, stop the algorithm.

4. Compute (2.7.20) as a candidate for μ and go to step 6.

5. Compute (2.7.22) and (2.7.23) as candidates for μ and go to step 6.

6. Compute

$$A := \sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\big(\mu\,\Psi(x, y, y', \ldots, y^{(n)})\big)}{\partial y^{(i)}}\Big)$$

   for each candidate for μ found in the previous steps. If $A = 0$ for such a μ, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

The case $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{y^{(n-1)}}$. We generalize Algorithm 2.26. Under the assumption that μ is of the prescribed form, the most general class of ODEs admitting such an integrating factors is obtained by Theorem 2.5 to be

$$y^{(n)} = -\frac{G_{y^{(n-2)}}}{f}(y^{(n-1)})^2 - \frac{G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}}}{f}y^{(n-1)} \\ - \Big(\frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-2)}}}{f}y^{(n-1)}\Big)y^{(n-1)}\ln(y^{(n-1)}). \qquad (2.7.24)$$

As in the discussion of the situation for third order ODEs in the framework of Algorithm 2.26, the right–hand–side of (2.7.24) is a linear polynomial in $\ln(y'')$, where we assume for the further discussion that $\frac{d}{dx}f \neq 0$, since otherwise (2.7.24) reduces to the class of ODEs

$$y^{(n)} = -\frac{G_{y^{(n-2)}}}{f}(y^{(n-1)})^2 - \frac{G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}}}{f}y^{(n-1)},$$

for which one can easily check using the Euler operator, whether $\frac{1}{y^{(n-1)}}$ is an integrating factor[26]. If $\frac{d}{dx}f \neq 0$, the right–hand–side of (2.7.24) is a linear polynomial in $\ln(y^{(n-1)})$ and we can read–off

$$T_1 = \frac{f_x}{f} + \frac{f_y}{f}y' + \ldots + \frac{f_{y^{(n-3)}}}{f}y^{(n-2)}, \quad T_2 = \frac{f_{y^{(n-2)}}}{f},$$

since $T_1 y^{(n-1)} + T_2(y^{(n-1)})^2$ can be found as the negative of the formal coefficient of $\ln(y^{(n-1)})$. Furthermore $T_1 y^{(n-1)} + T_2(y^{(n-1)})^2$ is a quadratic polynomial in $y^{(n-1)}$ and, hence, T_1 and T_2 can be read–off as the coefficients of $y^{(n-1)}$ and $(y^{(n-1)})^2$, respectively. Since $T_1 + T_2 y^{(n-1)}$ is the total derivative of $\ln(f)$, a candidate for an integrating factor is found to be

$$\mu = \exp\left(D^{-1}(T_1 + y^{(n-1)}T_2)\right)\frac{1}{y^{(n-1)}}, \tag{2.7.25}$$

where the definition of $D^{-1}(T_1 + y^{(n-1)}T_2)$ is provided by (2.7.15).

Finally, we obtain:

Algorithm 2.35. (Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{y^{(n-1)}}$) of n-th order ODEs) Given an n-th order ODE $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$ proceed as follows:

1. Check the ODE for exactness, i.e. set

   $$\Psi(x, y, y', \ldots, y^{(n)}) := y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)})$$

   and compute

   $$A := \sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\Psi(x, y, y', \ldots, y^{(n)})}{\partial y^{(i)}}\Big).$$

   If $A = 0$ the input ODE is already exact and can be integrated directly. Stop the algorithm.

[26] Note that in this situation f is a constant. Hence, one can check, whether the ODE is a quadratic polynomial in $y^{(n-1)}$ and whether $-\frac{G_x + y'G_y + \ldots + y^{(n-2)}G_{y^{(n-3)}} + G_{y^{(n-2)}}y^{(n-1)}}{f}$ is a total derivative using the Euler operator. If this is the case, $\frac{1}{y^{(n-1)}}$ is indeed an integrating factor.

2. Check, whether $\Phi(x, y, y', \ldots, y^{(n-1)})$ is a linear polynomial in $\ln(y^{(n-1)})$ of the form
$$H_1(x, y, y', \ldots, y^{(n-1)}) + H_2(x, y, y', \ldots, y^{(n-1)}) \ln(y^{(n-1)}),$$
where H_1 and $H_2 \neq 0$ can be arbitrary functions in their arguments. If this is not the case, stop the algorithm.

3. Check, whether $H_2(x, y, y', \ldots, y^{(n-1)})$ is a polynomial expression in $y^{(n-1)}$ of the form
$$S_1(x, y, y', \ldots, y^{(n-2)})y^{(n-1)} + S_2(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^2,$$
where S_1 and S_2 are arbitrary functions in their arguments, and if
$$\sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial(S_1 + S_2 y^{(n-1)})}{\partial y^{(i)}}\Big) = 0,$$
i.e. if $S_1 + S_2 y^{(n-1)}$ is a total derivative. If this is not the case, stop the algorithm.

4. Define $T_1 = -S_1$ and $T_2 = -S_2$ and compute (2.7.25) as a candidate for μ.

5. Compute
$$A := \sum_{i=0}^{n}(-1)^i \frac{d^i}{dx^i}\Big(\frac{\partial\big(\mu\, \Psi(x, y, y', \ldots, y^{(n)})\big)}{\partial y^{(i)}}\Big).$$
If $A = 0$, μ is an integrating factor of the ODE. Return the result for μ. Otherwise stop the algorithm. $\triangle$

Proof. The correctness of the algorithm follows from the preceding discussion. $\square$

2.8 Conclusions

2.8.1 Resumé

Of course, all algorithms and heuristics presented in this chapter treat very special classes of ODEs. It is not difficult to find examples, where none of the algorithms stated will return a satisfying result. Nevertheless, we think that the simplicity and efficiency of the Algorithms 2.7, 2.9, 2.11, 2.26, 2.19 and 2.22 for third order ODEs and the Algorithms 2.29, 2.30, 2.31, 2.33, 2.34 and 2.35 for n-th order ODEs stated above are good reasons for taking them into account in a symbolic ODE solving environment.

First of all the algorithms are easy to implement. The Euler operator (see Theorem 2.6) as a computational tool to check for the exactness of an ODE can be implemented efficiently in any computer algebra system (see also Section 3.6.3 and Section 3.6.4 of [3] for details on an implementation in the computer algebra system MATHEMATICA).

There are not many further requirements of the algorithms. In most cases it suffices to be able to recognize polynomial structures or rational expressions in a certain derivative of the dependent variable of a given ODE — a feature also available in most computer algebra systems. The candidates for the integrating factors can be computed directly without solving any additional ODEs or systems of linear partial differential equations. One even does not have to solve symbolic algebraic equations. This means that the above presented heuristics are completely independent of a computer algebra system's symbolic solver for such equations.

The only exceptions are Algorithms 2.15 and 2.32, which require a factoring routine for arbitrary expressions. From a truly mathematical standpoint these two algorithms are not of that interest. Nevertheless, we did a rough–and–ready implementation of these algorithms in the computer algebra system MuPAD[27], which worked fine and efficient in the case of simple examples. In the case of ODEs involving large and complex expressions we cannot expect satisfying results. Hence, an implementation of these two algorithms should always try to find integrating factors of the other types discussed with the help of the more systematic heuristics presented in the framework of Algorithms 2.7, 2.9, 2.11, 2.26, 2.19, 2.22, 2.29, 2.30, 2.31, 2.33, 2.34 and 2.35.

In practice, for general non–linear ODEs of order at least 2, algorithms for computing integrating factors (or at least candidates for integrating factors as presented in this chapter) seem to be of more relevance (in the context of ODE solvers of computer algebra systems) than Lie point symmetry methods as intro-

[27]The computer algebra system MuPAD by the company SciFace Software in Paderborn, Germany.

duced by Cheb-Terrab et. al. in [7], [9], [10], [12] and [13]. Indeed, according to our knowledge, Cheb-Terrab et. al. did not publish any algorithms for solving ODEs of order higher than 2 using Lie point symmetry methods. For ODEs of order at least 2, integrating factor methods are preferred. In fact, even the Lie point symmetry methods discussed in Section 1.2 can be seen as methods indirectly using integrating factors to reduce the order of the ODE under consideration. The reason is that given the infinitesimal generator

$$\xi(x,y)\,\frac{\partial}{\partial x}+\eta(x,y)\,\frac{\partial}{\partial y}$$

of a Lie point symmetry of a first order ODE $y'=\Phi(x,y)$, then

$$\frac{1}{\eta(x,y)-\xi(x,y)\,\Phi(x,y)}$$

is an integrating factor for $y'=\Phi(x,y)$. This connection between Lie point symmetries of first order ODEs and integrating factors is discussed in detail by H. Stephani in [64], Chapter I, Section 5.1 and by P. J. Olver in [51], Chapter 2, Theorem 2.48. See also Chapter 1 and Chapter 2 of [19].

2.8.2 Open problems and perspectives

1. The main theoretical question remaining open is the question, if it is possible to modify the algorithms for third and higher order non–linear ODEs discussed in this chapter, such that the test for exactness in the last step of the algorithms can be removed. In other words: is it possible not only to find necessary, but also sufficient conditions for the existence of integrating factors of the special forms considered? The main problem is that we do not want to use any rather theoretical sufficient conditions, but sufficient conditions, which can be verified without solving any auxiliary ODEs or even partial differential equations. At the current state of our research we do not see an alternative to the application of the Euler operator to establish sufficient conditions.

2. Another point remaining open is the question, whether even in the case an integrating factor is found, the integrator of a general purpose computer algebra system is able to compute the desired reduction of order. Of course, it is still a step forward to know that the integral of a certain differential expression corresponding to an ODE exists and thereby a

> reduction of order can be achieved. Nevertheless, if the integration cannot be performed, because the integrator of a computer algebra system is not powerful enough or the expression to be integrated is very complex, such that it is hard to find a suitable integration strategy, a reduced order ODE cannot be computed in closed form. Hence, further reductions of order using the algorithms discussed in this chapter or the symmetry methods presented in Chapter 1, cannot be applied.

In this chapter, the focus of attention was put on special classes of third and n-th order ODEs. In the next chapter we present new methods for finding integrating factors of other classes of ODEs, which arise from the application of skew symmetric differential operators. For such classes we give algorithms to determine integrating factors, which can also be computed without solving any additional auxiliary ODEs or partial differential equations. Furthermore, we can state explicit recursion formulas for the symbolic computation of integrals arising from the ODEs under consideration multiplied by their integrating factors: The reduction of order and the computation of the integrating factors can be achieved independent of the power of the integrator available in the computer algebra system, where the methods are to be implemented. This is a clear advantage to the methods presented in this chapter, where we always stated that the computation of the integrating factors considered involves an integration, but we did not state in detail, how to compute the integration efficiently in practice. However, for such methods as discussed in the following chapter, a new and extensive algebraic setup has to be developed.

Chapter 3

Skew symmetric hierarchies

In this chapter we discuss the computation of integrating factors for classes of ODEs obtained by the application of certain skew symmetric operators. The first part of this chapter serves to fix the notation and to introduce the underlying algebraic structure. Afterwards we introduce the classes of ODEs, on which we put the focus of our attention. A theoretical characterization on how to obtain integrating factors for these ODEs is established. Then we introduce the notion of the fundamental form of a skew symmetric operator, which serves to establish closed formulas for the efficient computation of the desired integrating factors. The last part of the chapter is dedicated to present closed formulas for the efficient integration of the total derivatives obtained by multiplication of the members of the class of ODEs under consideration with a suitable integrating factor.

In the preceding chapter we used the Euler operator (see Theorem 2.6) to check, whether a given differential expression is a total derivative or not. This was an essential aspect of all algorithms to find integrating factors of third and higher order ODEs without solving any auxiliary partial differential equations or additional ODEs. The methods of the last chapter mainly based on skillful observation of the "form" of a third or higher order ODE and the afterwards "extraction" of the integrating factor from the ODE under consideration.

The approaches presented within this chapter will not use pattern matching methods to determine integrating factors in the sense of Cheb-Terrab et. al. (see [11]). Moreover we will not have to make use of the Euler operator (see Theorem 2.6) to prove the exactness of a differential expression. We will arrange the underlying algebraic structure in such a way that we do no longer compute only "candidates" for integrating factors. In other words: in contrast to the last chapter, where we mainly considered necessary conditions for the existence of integrating factors, we now give necessary and sufficient conditions for the existence of integrating factors of the class of ODEs under consideration and directly compute them.

3.1 Basic terminology and general definitions

In the following, x denotes the independent variable and $u = u(x)$ the dependent variable. By $D = \frac{d}{dx}$ we denote the total differentiation with respect to x. All differentiations performed within this chapter will be total differentiations with respect to x so that no ambiguity in the notation will arise from simply writing D instead of $\frac{d}{dx}$.

We consider the smallest algebra $\mathcal{A} = \mathcal{A}(x, u, D)$ over some suitable number field (e.g. the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$) containing x, $u = u(x)$ and a fixed set $\mathcal{F}$ of infinitely often differentiable functions $f : \mathbb{R} \to \mathbb{R}$ in the variable x. The algebra $\mathcal{A}$ is assumed to be closed with respect to the application of the differential operator D to its elements, i.e. $D : \mathcal{A} \to \mathcal{A}$. The subalgebra $\mathcal{F}$ is assumed to be closed against taking integrals.

We use the abbreviations $u_x = Du$, $u_{xx} = Du_x$, $u_{xxx} = Du_{xx}$ etc. for the formal derivatives of the dependent variable $u = u(x)$ with respect to x. For $n \in \mathbb{N}$ we also denote the n-th derivative $\frac{d^n}{dx^n}u = D^n u$ by $u^{(n)}$. $\mathcal{A}$ is assumed to bear the algebra structure of pointwise multiplication.

To our mind, the choice of the structure of an algebra of functions and variables in the above sense seemed to fit best into the framework of a computer algebra system. Again, as in the preceding part of this thesis, we put particular emphasis on the fact that the methods for the computation of integrating factors presented in this chapter are formulated as close as possible to a final implementation in a computer algebra system. The computer algebra system MuPAD with its object oriented domains concept provides the necessary prerequisites for the implementation of the underlying mathematical structures in this chapter[1].

[1]The MuPAD Tutorial [18] gives detailed information on data structures and the MuPAD programming language. Especially, the object oriented concept of "domains" offers an essential basis for practical implementations of the algorithms presented in the framework of

As an example for such an algebra, consider the algebra containing all polynomials with real coefficients in x, u and the derivatives of u with respect to x.

As we know, $\mathcal{F}$ denotes the subset of all elements of $\mathcal{A}$, which do neither contain the dependent variable u nor any derivatives of u with respect to x. In other words: if we view u, u_x, u_{xx}, etc. as formal variables, then $A \in \mathcal{A}$ is an element of $\mathcal{F}$ if and only if the formal partial derivatives of A with respect to u, u_x, u_{xx}, etc. vanish. Additionally, the number field (the set of constants), over which the algebra $\mathcal{A}$ is to be considered, is interpreted as a subset of $\mathcal{F}$.

For an element $A \in \mathcal{A} \setminus \mathcal{F}$ by $\mathrm{ord}_u(A)$ we denote the order of the highest derivative of u with respect to x contained in A. We use the convention $\mathrm{ord}_u(A) = 0$, if A does not contain any derivative of u with respect to x, but only u itself. For arbitrary elements $B \in \mathcal{F} \setminus \{0\}$ we define $\mathrm{ord}_u(B) = -\infty$ (B does not depend on u or any of the derivatives of u with respect to x). For technical reasons, which will become clear in the next section, we say that the zero element 0 has every order. It follows that

$$\mathrm{ord}_u : \mathcal{A} \setminus \{0\} \to \mathbb{N}_0 \cup \{-\infty\}$$

is a sub–additive and sub–multiplicative map, i.e.

$$\mathrm{ord}_u(A + B) \leq \max\{\mathrm{ord}_u(A), \mathrm{ord}_u(B)\}$$

and

$$\mathrm{ord}_u(AB) \leq \max\{\mathrm{ord}_u(A), \mathrm{ord}_u(B)\}$$

for all $A, B \in \mathcal{A} \setminus \{0\}$.

By $\mathcal{N}$ we denote the set of all total derivatives in $\mathcal{A}$, i.e.

$$\mathcal{N} = \{Q \mid \text{there is a } P \in \mathcal{A}, \text{ such that } Q = DP\}.$$

By definition $\mathcal{N} \supset \mathcal{F}$. The set $\mathcal{N}$ is a linear space, since differentiation is a linear operation, but in general the product of two elements of $\mathcal{N}$ is not again contained in $\mathcal{N}$. Take for example the first derivative u_x of the dependent variable. Then clearly $u_x \in \mathcal{N}$, since $u \in \mathcal{A}$ and $u_x = Du$, but $u_x^2 \notin \mathcal{N}$, since u_x^2 cannot be integrated formally.

---

this chapter (see also Appendix D of [50] and [37] as well as chapters 3 and 5 of [49]).

For the course of this chapter we will use the abbreviation D^{-1} for the symbolic integral of an element of $\mathcal{N}$ with respect to x, i.e.

$$D^{-1}A = \int A\,dx$$

for all $A \in \mathcal{N}$. In the framework of our setting, D^{-1} is viewed as a linear operator $\mathcal{N} \to \mathcal{A}$. Whenever it is necessary to specify lower and upper bound for the integration, we use the standard notation for integrals, so that no ambiguity in the notation will arise. In most situations D^{-1} is treated as the formal inverse of the differential operator D, i.e. constants of integration will be ignored.

For each element $K \in \mathcal{A} \setminus \mathcal{F}$, $\mathrm{ord}_u(K) > 0$, the equation $K = 0$ is a differential equation in the variable $u = u(x)$. If K already is an element of $\mathcal{N}$, then this means that K can be integrated and the order of the ODE $K = 0$ can be reduced by 1. Note that integration of elements of $\mathcal{N}$ provides again an element of $\mathcal{A}$, i.e. no symbolic integrals will remain, which cannot be computed explicitly. In the terminology of the preceding chapter, $\mathcal{N}$ is exactly the subset of all elements in $\mathcal{A}$, which vanish when applying the Euler operator (see Theorem 2.6) to them.

On the algebra $\mathcal{A}$ we define the relation $\sim_{\mathcal{N}}$ as follows: $A \sim_{\mathcal{N}} B$ holds for $A, B \in \mathcal{A}$ if and only if $A - B \in \mathcal{N}$. This relation clearly defines an equivalence relation on $\mathcal{A}$. For $A \in \mathcal{A}$ we denote by $[A]_{\mathcal{N}}$ the equivalence class of A with respect to $\sim_{\mathcal{N}}$, i.e.

$$[A]_{\mathcal{N}} = \{B \in \mathcal{A} \mid B \sim_{\mathcal{N}} A\}.$$

The set of all such equivalence classes is denoted by $\mathcal{A}/\mathcal{N}$, i.e.

$$\mathcal{A}/\mathcal{N} = \{[A]_{\mathcal{N}} \mid A \in \mathcal{A}\}.$$

The question, if for two given elements $A, B \in \mathcal{A}$ their equivalence classes coincide, i.e. $[A]_{\mathcal{N}} = [B]_{\mathcal{N}}$, is reduced to the question, whether $A - B \in \mathcal{N}$. To decide this, we can apply the Euler operator presented in Theorem 2.6 to the differential expression $A - B$ and see, if the result vanishes.

Of course, applying the Euler operator is an efficient method for testing, if a differential expression is a total derivative. Nevertheless, it does not help to compute the integral $D^{-1}(A - B)$, if $A - B$ is contained in $\mathcal{N}$ — something we are of course interested in when trying to reduce the order of an ODE after multiplication by a suitable integrating factor. Hence, an alternative way to decide, whether $A - B \in \mathcal{N}$ and, if this is the case, to compute $D^{-1}(A - B)$, is presented in Section 3.2 of this chapter when we discuss the so–called canonical form of such differential expressions.

With the help of the equivalence classes $[A]_{\mathcal{N}}$, $A \in \mathcal{A}$, we define the bilinear form

$$\langle \cdot\,, \cdot \rangle : \mathcal{A} \times \mathcal{A} \to \mathcal{A}/\mathcal{N}, \quad \langle A, B \rangle := [AB]_{\mathcal{N}}.$$

The map $\langle \cdot\,, \cdot \rangle : \mathcal{A} \times \mathcal{A} \to \mathcal{A}/\mathcal{N}$ maps two elements of $\mathcal{A}$ to the equivalence class of their product in $\mathcal{A}$. Since we assume the multiplication in $\mathcal{A}$ to be commutative, it follows

$$\langle A, B \rangle = \langle B, A \rangle$$

for all $A, B \in \mathcal{A}$, i.e. $\langle \cdot\,, \cdot \rangle : \mathcal{A} \times \mathcal{A} \to \mathcal{A}/\mathcal{N}$ is a symmetric bilinear form. It will be referred to as *density valued scalar product* and we will also call the equivalence classes from $\mathcal{A}/\mathcal{N}$ *densities*.

This density valued scalar product can be used to give an alternative characterization of total derivatives in $\mathcal{A}$:

Lemma 3.1. *Let $A \in \mathcal{A}$. Then $A \in \mathcal{N}$ if and only if $\langle 1, A \rangle = 0$.*

Proof. We have $A \in \mathcal{N}$ if and only if $[A]_{\mathcal{N}} = 0$ if and only if $\langle 1, A \rangle = 0$. This proves the assertion of the lemma. □

The notion of an integrating factor for elements of $\mathcal{A}$ can now be formulated as follows:

Definition 3.2. Let $K \in \mathcal{A} \setminus \mathcal{F}$, $G \in \mathcal{A}$ and $\mathrm{ord}_u(K) > \mathrm{ord}_u(G)$. Then G is called an *integrating factor* for $K = 0$, if the density valued scalar product of G and K vanishes, i.e.

$$\langle G, K \rangle = 0.$$

If G is an integrating factor for $K = 0$, the product GK is an element of $\mathcal{N}$ and the integral $D^{-1}(GK)$ is an element of $\mathcal{A}$. In the situation of Definition 3.2, $D^{-1}(GK) = c$, c some constant, will be referred to as a *conserved quantity* for $K = 0$ (note that $\mathrm{ord}_u(D^{-1}(GK)) < \mathrm{ord}_u(K)$).

Example 3.3. Consider the elements $u, u_x \in \mathcal{A}$. Then u is an integrating factor for $u_x = 0$, since $D(\frac{1}{2}u^2) = uu_x$ and, hence, $\langle u, u_x \rangle = 0$. ◇

We also introduce the more general notion of extended integrating factors, to which we will come back in the later course of this chapter:

Definition 3.4. Let $K \in \mathcal{A} \setminus \mathcal{F}$, $\mathrm{ord}_u(K) > 0$, and $G \in \mathcal{A}$. Then G is called an *extended integrating factor* for $K = 0$, if the density valued scalar product of G and K vanishes, i.e.

$$\langle G, K \rangle = 0.$$

The only difference between Definition 3.4 and Definition 3.2 is that the order of an extended integrating factor may be higher than or equal to the order of the ODE $K = 0$ under consideration. E.g. let $K = u_x$ and $G = u_{xx}$. Then G is an extended integrating factor of $K = 0$, since $D^{-1}(KG) = \frac{1}{2}u_x^2$. Of course, this does not produce a reduction of the order of the ODE $K = 0$, since $\mathrm{ord}_u(K) = 1 = \mathrm{ord}_u(D^{-1}(GK))$. We will come back to the notion of extended integrating factors and their use in Section 3.3.

Finally, we need the notion of skew symmetric operators $\mathcal{A} \to \mathcal{A}$:

Definition 3.5. An operator $\Theta : \mathcal{A} \to \mathcal{A}$ is called *skew symmetric with respect to the density valued scalar product* $\langle \cdot, \cdot \rangle$ or for short *skew symmetric*, if

$$\langle A, \Theta(B) \rangle = -\langle \Theta(A), B \rangle.$$

Example 3.6. (i) The differential operator D is a skew symmetric $\mathcal{A} \to \mathcal{N}$, since for any $A, B \in \mathcal{A}$ we have

$$\begin{aligned} \langle A, DB \rangle + \langle DA, B \rangle &= [AB_x + A_x B]_{\mathcal{N}} \\ &= [(AB)_x]_{\mathcal{N}}, \end{aligned}$$

i.e. $\langle A, DB \rangle + \langle DA, B \rangle = 0$.

(ii) The operator $\Theta : \mathcal{A} \to \mathcal{A}$, $\Theta = D^2 u - uD^2$ is a skew symmetric operator, since

$$\begin{aligned} \langle A, \Theta(B) \rangle + \langle \Theta(A), B \rangle &= \langle A, u_{xx} B + 2u_x B_x \rangle + \langle u_{xx} A + 2u_x A_x, B \rangle \\ &= [2u_{xx} AB + 2u_x (AB)_x]_{\mathcal{N}} \\ &= [(2u_x (AB))_x]_{\mathcal{N}} \end{aligned}$$

and $(2u_x(AB))_x$ is a total derivative, i.e. $[(2u_x(AB))_x]_{\mathcal{N}} = 0$.

(iii) All operators $\Theta : \mathcal{A} \to \mathcal{A}$ of the form $\Theta = \sum_{k=0}^{n} D^{2k+1}$, $n \in \mathbb{N}$, are skew symmetric. $\diamond$

3.2 Canonical form of differential expressions

Assume, $A = A(x, u, u_x, u_{xx}, \ldots) \in \mathcal{A}$. From the definitions introduced in the last section, we know that $D^{-1}(A)$ is an element of $\mathcal{A}$ if and only if $A \in \mathcal{N}$. This section is dedicated to present an algorithm to decide, whether $D^{-1}(A)$ for an arbitrary element of $A \in \mathcal{A}$ is contained in $\mathcal{A}$, and to determine the representation of its integral.

In other words: we present an algorithm to decide, whether or not a given element of $\mathcal{A}$ is a total derivative with respect to x. But the algorithm will even accomplish more: If $A \in \mathcal{N}$, then it will help to compute the element $D^{-1}(A) \in \mathcal{A}$. In order to do this, we split $\mathcal{A}$ into a direct sum

$$\mathcal{A} = \mathcal{N} \oplus \mathcal{J}$$

and give an algorithm, which decomposes each element of $\mathcal{A}$ into these components, and furthermore represents the integrals of elements from $\mathcal{N}$ by elements of $\mathcal{A}$. Since the sum is direct, the representations obtained by this method are unique. The summand $\mathcal{J}$ we call the *non–integrable part* of $\mathcal{A}$, and $\mathcal{N}$ is also referred to as the *integrable part* of $\mathcal{A}$.

Definition 3.7. Let $n \in \mathbb{N}$.

(i) The set $\mathcal{J}_n \subset \mathcal{A}$ is defined to be the set of all elements $A = A(x, u, u_x, \ldots, u^{(n)}) \in \mathcal{A}$, $n = \mathrm{ord}_u(A)$, with the property that when A is expanded in a formal Taylor series with respect to $u^{(n)}$ (highest derivative of u with respect to x in A), then the terms of polynomial order 0 and 1 do vanish.

(ii) By $\mathcal{J}_0$ we denote the set of all $A = A(x, u) \in \mathcal{A}$, such that in the formal Taylor series expansion of A with respect to u the term of polynomial order 0 vanishes.

Definition 3.8. Let $\mathcal{J}_k \subset \mathcal{A}$, $k \in \mathbb{N}_0$, be as in Definition 3.7. Then we define:

$$\mathcal{J} := \bigoplus_{k=0}^{\infty} \mathcal{J}_k.$$

Remark 3.9. Obviously, in the intersection $\mathcal{J} \cap \mathcal{N}$ there can be no other element than 0, since in any term of the form $D(A)$, $A \in \mathcal{A}$, which is not already contained in $\mathcal{F}$, the highest derivative of u with respect to x appears linearly.

Furthermore, any term, where there is no u contained at all, can be integrated anyway. If in a term the highest order derivative $u^{(n)}$ appears linearly, then it can be written as the sum of a derivative of a term, which contains $u^{(n-1)}$ as highest order derivative, and some terms containing only derivatives $u^{(i)}$, where $i < n$ (integration by parts).

From this remark clearly follows the decomposition and the algorithm to compute it. The rest of this section is now dedicated to present an algorithm to compute a canonical representation of $D^{-1}(A)$ for an arbitrary element $A \in \mathcal{A}$ using the ideas sketched above.

In the following we demand that any of the differential expressions treated is given in completely expanded form, i.e. as a sum of terms, where each term is a pure multiplicative expression. The sum of all terms in A depending on the highest derivative of u with respect to x will be called the *highest order term* of A.

Definition 3.10. For an element $A = A(x, u, u_x, u_{xx}, \ldots, u^{(n)}) \in \mathcal{A}$, the formal integral $D^{-1}(A)$ is said to be in *canonical form with respect to* $u^{(n)}$, $n \in \mathbb{N}_0$, if for each of its terms one of the following condition holds:

- The term does not contain any symbolic integration D^{-1}.
- The term is of the form $D^{-1}(Q(x, u, u_x, u_{xx}, \ldots, u^{(n)}))$, where $Q = Q(x, u, u_x, u_{xx}, \ldots, u^{(n)})$ is an element of $\mathcal{J}_n$.
- The term is of the form $D^{-1}(R(x, u, u_x, \ldots, u^{(n-1)}))$ for some $R(x, u, u_x, \ldots, u^{(n-1)}) \in \mathcal{A}$ (a situation, which can only occur in the case $n \geq 1$).

In the following we give an outline of our ideas how to compute the canonical form with respect to $u^{(n)}$ of a general expression $D^{-1}(A)$ for $A = A(x, u, u_x, u_{xx}, \ldots, u^{(n)}) \in \mathcal{A}$. The formal Taylor series expansion of A with respect to $u^{(n)}$ provides a decomposition of A of the form

$$\begin{aligned} A = A_0(x, u, u_x, u_{xx}, \ldots, u^{(k)}) + A_1(x, u, u_x, u_{xx}, \ldots, u^{(l)})\, u^{(n)} + \\ \text{terms of polynomial order } \geq 2 \text{ in } u^{(n)}, \end{aligned}$$

where $k, l \in \mathbb{N}$ with $k, l < n$, i.e. A_0 and A_1 do not depend on $u^{(n)}$. This decomposition of A is a central ingredient for the computation of the canonical form with respect to $u^{(n)}$.

Lemma 3.11. *Let $A(x, u, u_x, u_{xx}, \ldots, u^{(n)}) \in \mathcal{A}$ and let $u^{(n)}$ be the highest order derivative of u with respect to x appearing in A. Then the highest order term of $\frac{d}{dx}\int_0^{u^{(n)}} A(x, u, u_x, u_{xx}, \ldots, \xi)\, d\xi$ is $u^{(n+1)}A(x, u, u_x, u_{xx}, \ldots, u^{(n)})$.*

Proof. The assertion of the Lemma follows directly from the chain rule of differential calculus. It provides:

$$\begin{aligned}
&\frac{d}{dx}\int_0^{u^{(n)}} A(x, u, u_x, u_{xx}, \ldots, \xi)\, d\xi \\
= \quad &\Big(\int_0^{u^{(n)}} A(x, u, u_x, u_{xx}, \ldots, \xi)\, d\xi\Big)_x + \\
&u_x\Big(\int_0^{u^{(n)}} A(x, u, u_x, u_{xx}, \ldots, \xi)\, d\xi\Big)_u + \\
&u_{xx}\Big(\int_0^{u^{(n)}} A(x, u, u_x, u_{xx}, \ldots, \xi)\, d\xi\Big)_{u_x} + \ldots + \\
&u^{(n+1)}A(x, u, u_x, u_{xx}, \ldots, u^{(n)}),
\end{aligned}$$

where we have used x, u, u_x, etc. as indices to denote the partial derivatives $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial u}$, $\frac{\partial}{\partial u_x}$ etc. In the above sum the only term depending on the $(n+1)$-st derivative of u is $u^{(n+1)}A(x, u, u_x, u_{xx}, \ldots, u^{(n)})$. This proves the assertion. $\square$

With the help of Lemma 3.11 we can construct the canonical form with respect to $u^{(n)}$ of a given expression $D^{-1}(A(x, u, u_x, u_{xx}, \ldots, u^{(n)}))$. Before we consider the general case, we have a look at an example.

Example 3.12. Let $A = A(x, u, u_x, u_{xx}) = x\, u\, u_x\, u_{xx}$, i.e. we consider the case $n = 2$. We want to compute the canonical form of $D^{-1}(A)$ with respect to u_{xx}. We proceed as follows: The formal Taylor series expansion of A with respect to the highest order derivative provides the representation

$$\begin{aligned}
A(x, u, u_x, u_{xx}) =&A(x, u, u_x, 0) + u_{xx}A_{u_{xx}}(x, u, u_x, 0)+ \\
&\big(A(x, u, u_x, u_{xx}) - u_{xx}A_{u_{xx}}(x, u, x_0, 0)- \\
&A(x, u, x, 0)\big).
\end{aligned}$$

Application of D^{-1} gives

$$\begin{aligned}
D^{-1}\big(A(x, u, u_x, u_{xx})\big) =&D^{-1}\big(A(x, u, u_x, 0)\big) + D^{-1}\big(u_{xx}A_{u_{xx}}(x, u, u_x, 0)\big)+ \\
&D^{-1}\big(A(x, u, u_x, u_{xx}) - u_{xx}A_{u_{xx}}(x, u, u_x, 0)- \\
&\quad A(x, u, u_x, 0)\big).
\end{aligned}$$

Finally we write $D^{-1}(u_{xx}A_{u_{xx}}(x,u,u_x,0))$ in the form

$$\begin{aligned}D^{-1}\big(u_{xx}A_{u_{xx}}(x,u,u_x,0)\big) =&D^{-1}\Big(u_{xx}A_{u_{xx}}(x,u,u_x,0)-\\ &\frac{d}{dx}\int_0^{u_x} A_{u_{xx}}(x,u,\xi,0)\,d\xi\Big)+\\ &\int_0^{u_x} A_{u_{xx}}(x,u,\xi,0)\,d\xi.\end{aligned}$$

This provides the canonical form of $D^{-1}(A)$ with respect to u_{xx}, since by these formulas we find

$$\begin{aligned}D^{-1}(A) &= D^{-1}(x\,u\,u_x\,u_{xx})\\ &= D^{-1}\Big(u_{xx}\,(x\,u\,u_x) - \frac{d}{dx}\Big(\frac{1}{2}\,x\,u\,u_x^2\Big)\Big) + \frac{1}{2}\,x\,u\,u_x^2\\ &= \frac{1}{2}\,x\,u\,u_x^2 - \frac{1}{2}\,D^{-1}(u\,u_x^2 + x\,u_x^3).\end{aligned}$$

◇

We can generalize this example in the form of the following theorem:

Theorem 3.13. *Let $A = A(x,u,u_x,u_{xx},\dots,u^{(n-1)},u^{(n)}) \in \mathcal{A}$ and $u^{(n)}$ the highest order derivative of u with respect to x in A. Then the right–hand–side of*

$$\begin{aligned}D^{-1}(A) =&D^{-1}\big(A(x,u,u_x,u_{xx},\dots,u^{(n-1)},0)\big)+\\ &D^{-1}\Big(u^{(n)}A_{u^{(n)}}(x,u,u_x,u_{xx},\dots,u^{(n-1)},0)-\\ &\quad\frac{d}{dx}\int_0^{u^{(n-1)}} A_{u^{(n)}}(x,u,u_x,u_{xx},\dots,\xi,0)\,d\xi\Big)+\\ &\int_0^{u^{(n-1)}} A_{u^{(n)}}(x,u,u_x,u_{xx},\dots,\xi,0)\,d\xi+\\ &D^{-1}\big(A(x,u,u_x,u_{xx},\dots,u^{(n-1)},u^{(n)})-\\ &\quad u^{(n)}A_{u^{(n)}}(x,u,u_x,u_{xx},\dots,u^{(n-1)},0)-\\ &\quad A(x,u,u_x,u_{xx},\dots,u^{(n-1)},0)\big)\end{aligned}$$

is in canonical form with respect to $u^{(n)}$.

Proof. First of all, total differentiation of both sides of the above stated identity for $D^{-1}(A)$ shows that the equals sign is valid. Those terms, which do not contain $u^{(n)}$, are in canonical form with respect to $u^{(n)}$. Hence, the first term

$$D^{-1}\big(A(x,u,u_x,u_{xx},\ldots,u^{(n-1)},0)\big)$$

is in canonical form with respect to $u^{(n)}$. The same holds for

$$\int_0^{u^{(n-1)}} A_{u^{(n)}}(x,u,u_x,u_{xx},\ldots,\xi,0)\,d\xi.$$

The last term of the right–hand–side is in canonical form with respect to $u^{(n)}$, since by construction the formal Taylor series expansion of the integrand with respect to $u^{(n)}$ only contains terms of order ≥ 2 in $u^{(n)}$. Finally,

$$D^{-1}\Big(u^{(n)}A_{u^{(n)}}(x,u,u_x,u_{xx},\ldots,u^{(n-1)},0)- \frac{d}{dx}\int_0^{u^{(n-1)}} A_{u^{(n)}}(x,u,u_x,u_{xx},\ldots,\xi,0)\,d\xi\Big)$$

does not contain $u^{(n)}$ by Lemma 3.11. □

We now generalize Definition 3.10:

Definition 3.14. For an element $A = A(x,u,u_x,u_{xx},\ldots,u^{(n)}) \in \mathcal{A}$, the formal integral $D^{-1}(A)$ is said to be in *canonical form* if

$$D^{-1}(A) = \widetilde{A} + \sum_{k=0}^{n} D^{-1}(A_k),$$

where $\widetilde{A} \in \mathcal{A}$ and $A_k = A_k(x,u,u_x,\ldots,u^{(k)}) \in \mathcal{J}_k$, $0 \leq k \leq n$.

An algorithm to compute the canonical form is provided by a successive application of the formula in Theorem 3.13.

Algorithm 3.15. (Computation of the canonical form) Let $A = A(x,u,u_x,u_{xx},\ldots,u^{(n)})$.

1. Set $T := D^{-1}(A)$, $i := n$ and $C := 0$.

2. Use the formula in Theorem 3.13 to compute the canonical form of T with respect to $u^{(i)}$. This gives a decomposition of the form

$$\begin{aligned} T = & A_0(x, u, u_x, \ldots, u^{(i-1)}) + \\ & D^{-1}(A_1(x, u, u_x, \ldots, u^{(i-1)})) + \\ & D^{-1}(A_2(x, u, u_x, \ldots, u^{(i)})), \end{aligned}$$

   where $A_0, A_1 \in \mathcal{A}$ and $A_2 \in \mathcal{J}_i$.

3. Set $C := C + A_0 + D^{-1}(A_2)$, $T := D^{-1}(A_1)$ and $i := i - 1$.

4. If $i \geq 1$ and $A_1 \neq 0$, go back to step 2.

5. Set $C := C + T$ and return C, the canonical form of $D^{-1}(A)$. $\triangle$

Proof. (*Correctness*) The correctness of the algorithm follows directly from Theorem 3.13. $\square$

Remark 3.16. For an arbitrary element $A = A(x, u, u_x, \ldots, u^{(n)}) \in \mathcal{A}$, Algorithm 3.15 finds $\widetilde{A} \in \mathcal{A}$ and $A_k = A_k(x, u, u_x, \ldots, u^{(k)}) \in \mathcal{J}_k$, $0 \leq k \leq n$, such that

$$D^{-1}(A) = \widetilde{A} + \sum_{k=0}^{n} D^{-1}(A_k).$$

Application of D provides the decomposition

$$A = D(\widetilde{A}) + \sum_{k=0}^{n} A_k,$$

where $D(\widetilde{A})$ is contained in the integrable part $\mathcal{N}$ of $\mathcal{A}$ and $\sum_{k=0}^{n} A_k$ is an element of the non–integrable part $\mathcal{J}$ of $\mathcal{A}$, more precisely of $\bigoplus_{k=0}^{n} \mathcal{J}_k$.

We apply Algorithm 3.15 in the next two examples, before we return to our main aim, the systematic computation of integrating factors, in the next section:

Example 3.17. Assume we search for integrating factors of the form $\eta = \eta(x) \in \mathcal{F}$ for the homogeneous linear ODE $u_{xx} + pu = 0$ in $u = u(x)$ and $p = p(x)$ some generic element of $\mathcal{F}$. Application of Algorithm 3.15 to the expression $D^{-1}(\eta u_{xx} + \eta pu)$ to compute its canonical form provides:

$$\begin{aligned} D^{-1}(\eta u_{xx} + \eta pu) &= \eta u_x - D^{-1}(u_x \eta_x) + D^{-1}(\eta pu) \\ &= \eta u_x - u\eta_x + D^{-1}(u\eta_{xx} + up\eta), \end{aligned}$$

i.e. the canonical form of $D^{-1}(\eta u_{xx} + \eta pu)$ is $\eta u_x - u\eta_x + D^{-1}(u\eta_{xx} + up\eta)$. Whenever $\eta(x)$ is an integrating factor of $u_{xx} + pu = 0$, the symbolic integral $D^{-1}(u\eta_{xx} + up\eta)$ must vanish, i.e. we obtain $\eta_{xx} + p\eta = 0$, and it follows that $\eta(x)$ is an integrating factor of $u_{xx} + pu = 0$ if it is a solution of $u_{xx} + pu = 0$. From a practical standpoint this fact is useless for solving the homogeneous linear ODE $u_{xx} + pu$, but it mirrors the fact that one can always find another solution, when one solution of the ODE is already known: the solution, i.e. the integrating factor, allows a reduction of the order of the ODE by 1 resulting in a first order ODE, which can be solved e.g. by the well–known method of separation of variables. $\diamond$

Example 3.18. Assume, we want to find integrating factors of the form $\mu = \mu(x, u, u_x) = au + bu_x$, $a = a(x), b = b(x) \in \mathcal{F}$ for the homogeneous linear ODE $u_{xx} + pu = 0$ in $u = u(x)$ and $p = p(x)$ some generic element of $\mathcal{F}$. Application of Algorithm 3.15 to the expression $D^{-1}(\mu u_{xx} + \mu pu)$ to compute its canonical form provides:

$$\begin{aligned}D^{-1}(\mu u_{xx} + \mu pu) =& D^{-1}(auu_{xx} + apu^2 + \\ & bu_x u_{xx} + bpu_x u) \\ =& -\frac{1}{2}a_x u^2 + \frac{1}{2}bu_x^2 + auu_x + \frac{1}{2}bpu^2 + \\ & D^{-1}\Big(-\frac{1}{2}b_x u_x^2 + apu^2 - \frac{1}{2}bp_x u^2 - \\ & \frac{1}{2}pb_x u^2 - au_x^2 + a_{xx}u^2\Big).\end{aligned}$$

If $\mu = au + bu_x$ is an integrating factor for $u_{xx} + pu = 0$, then the above integrand must vanish. Considering the integrand as a polynomial in u_x, we can make the ansatz $a = -\frac{1}{2}b_x$. If we insert this into the above expression, we obtain

$$\begin{aligned}D^{-1}(\mu u_{xx} + \mu pu) =& \frac{1}{2}bu_x^2 + \frac{1}{4}b_{xx}u^2 + \\ & \frac{1}{2}bpu^2 - \frac{1}{2}b_x uu_x + \\ & D^{-1}\Big(-\frac{1}{4}u^2(b_{xxx} + 4pb_x + 2p_x b)\Big).\end{aligned}$$

Hence, if $\mu = au + bu_x$, $a = -\frac{1}{2}b_x$, is an integrating factor of $u_{xx} + pu = 0$, we should have

$$-\frac{1}{4}u^2(b_{xxx} + 4pb_x + 2p_x b) = 0.$$

Again, as in the preceding example, this information does not seem to help in finding an integrating factor for $u_{xx} + pu = 0$, since computing b means solving the homogeneous linear ODE

$$b_{xxx} + 4pb_x + 2p_x b = 0,$$

which — considered as an ODE in b for given p — is of order 3. Hence, determining μ seems to be harder than directly solving the original ODE. But the above equation gives us some more information: If u is indeed a solution of the original ODE $u_{xx} + pu = 0$, then we obtain by ignoring constants of integration

$$0 = \frac{1}{2}bu_x^2 + \frac{1}{4}b_{xx}u^2 + \frac{1}{2}bpu^2 - \frac{1}{2}b_x uu_x + D^{-1}\Big(-\frac{1}{4}u^2(b_{xxx} + 4pb_x + 2p_x b)\Big).$$

If we write the equation as

$$D^{-1}\Big(\frac{1}{4}u^2(b_{xxx} + 4pb_x + 2p_x b)\Big) = \frac{1}{2}bu_x^2 + \frac{1}{4}b_{xx}u^2 + \frac{1}{2}bpu^2 - \frac{1}{2}b_x uu_x,$$

we see that u^2 is an integrating factor for the third order homogeneous linear ODE $b_{xxx} + 4pb_x + 2p_x b = 0$. Hence, finding an integrating factor for any third order homogeneous linear ODE of the form $b_{xxx} + 4pb_x + 2p_x b = 0$ can be reduced to computing powers of solutions of the corresponding second order homogeneous linear ODE we started with. $\diamond$

3.3 Integrating factors and skew symmetric operators

Assume that $\Theta : \mathcal{A} \to \mathcal{A}$ is a skew symmetric operator. Then we obtain for $K_0 := \Theta(1) \in \mathcal{A}$:

$$\begin{aligned}\langle 1, K_0\rangle = \langle 1, \Theta(1)\rangle &= -\langle \Theta(1), 1\rangle \\ &= -\langle K_0, 1\rangle = -\langle 1, K_0\rangle,\end{aligned}$$

i.e. $\langle 1, K_0\rangle = 0$ and, hence, by Lemma 3.1 we have $K_0 \in \mathcal{N}$. Since $K_0 \in \mathcal{N}$, the integral $D^{-1}K_0$ is an element of $\mathcal{A}$. We set $G_0 := D^{-1}K_0$ and compute $K_1 := \Theta(G_0)$. Then again $K_1 \in \mathcal{N}$, since

$$\begin{aligned}\langle 1, K_1\rangle = \langle 1, \Theta(G_0)\rangle &= -\langle \Theta(1), G_0\rangle \\ &= -\langle K_0, G_0\rangle = -\langle DG_0, G_0\rangle \\ &= \Big[-\frac{1}{2}(G_0^2)_x\Big]_{\mathcal{N}} = 0.\end{aligned}$$

Furthermore, G_0K_1 is an element of $\mathcal{N}$, since

$$\begin{aligned}\langle G_0, K_1\rangle &= \langle G_0, \Theta(G_0)\rangle = -\langle\Theta(G_0), G_0\rangle\\ &= -\langle K_1, G_0\rangle = -\langle G_0, K_1\rangle,\end{aligned}$$

which means $\langle G_0, K_1\rangle = 0$ and, henceforth, $G_0K_1 \in \mathcal{N}$. By the same arguments it follows:

$$\begin{aligned}\langle G_0, K_0\rangle &= \langle G_0, \Theta(1)\rangle = -\langle\Theta(G_0), 1\rangle\\ &= -\langle K_1, 1\rangle = -\langle 1, K_1\rangle\\ &= 0,\end{aligned}$$

as we have already seen above. If $G_0, K_0, K_1 \in \mathcal{A}\setminus\mathcal{F}$ and $\mathrm{ord}_u(G_0) < \mathrm{ord}_u(K_0)$ as well as $\mathrm{ord}_u(G_0) < \mathrm{ord}_u(K_1)$, it follows that G_0 is an integrating factor for the ODEs $K_0 = 0$ and $K_1 = 0$.

This exemplary computations serve as a motivation for the following more general theorem:

Theorem 3.19. (Skew symmetric hierarchy) *Let Θ be a skew symmetric operator $\mathcal{A} \to \mathcal{A}$. We define:*

$$\begin{aligned} K_0 &:= \Theta(1), & G_0 &:= D^{-1}K_0,\\ K_i &:= \Theta(G_{i-1}), & G_i &:= D^{-1}K_i,\end{aligned}$$

for all $i \in \mathbb{N}$. Then the following hold:

(i) $G_i \in \mathcal{A}$ *for all* $i \in \mathbb{N}_0$.

(ii) $K_i \in \mathcal{N}$ *for all* $i \in \mathbb{N}_0$.

(iii) $G_iK_j \in \mathcal{N}$ *for all* $i, j \in \mathbb{N}_0$.

Proof. The assertion in (i) follows, when we have proved (ii). For (ii) we have to show that $\langle 1, K_i\rangle = 0$ for all $i \in \mathbb{N}_0$. In the motivation for this theorem (see above), we already treated the case $i = 0$ and proved that $K_0 \in \mathcal{N}$ under the assumptions of the theorem. Let $i \geq 1$. We have:

$$\begin{aligned}
\langle 1, K_i \rangle &= \langle 1, \Theta(G_{i-1}) \rangle = -\langle \Theta(1), G_{i-1} \rangle = -\langle K_0, G_{i-1} \rangle \\
&= -\langle DG_0, G_{i-1} \rangle = \langle G_0, DG_{i-1} \rangle = \langle G_0, D(D^{-1}K_{i-1}) \rangle \\
&= \langle G_0, K_{i-1} \rangle \\
&= \langle G_0, \Theta(G_{i-2}) \rangle = -\langle \Theta(G_0), G_{i-2} \rangle = -\langle K_1, G_{i-2} \rangle \\
&= -\langle DG_1, G_{i-2} \rangle = \langle G_1, DG_{i-2} \rangle = \langle G_1, D(D^{-1}K_{i-2}) \rangle \\
&= \langle G_1, K_{i-2} \rangle \\
&= \cdots \\
&= \langle G_{i-1}, K_0 \rangle \\
&= \langle G_{i-1}, \Theta(1) \rangle = -\langle \Theta(G_{i-1}), 1 \rangle = -\langle K_i, 1 \rangle \\
&= -\langle 1, K_i \rangle,
\end{aligned}$$

which means that $2\langle 1, K_i \rangle = 0$ and, hence, $K_i \in \mathcal{N}$.

For (iii) we have to prove that $\langle G_i, K_j \rangle = 0$ for all $i, j \in \mathbb{N}_0$. We have:

$$\begin{aligned}
\langle G_i, K_j \rangle &= \langle G_i, \Theta(G_{j-1}) \rangle = -\langle \Theta(G_i), G_{j-1} \rangle = -\langle K_{i+1}, G_{j-1} \rangle \\
&= -\langle DG_{i+1}, G_{j-1} \rangle = \langle G_{i+1}, DG_{j-1} \rangle \\
&= \langle G_{i+1}, K_{j-1} \rangle \\
&= \langle G_{i+1}, \Theta(G_{j-2}) \rangle = -\langle \Theta(G_{i+1}), G_{j-2} \rangle = -\langle K_{i+2}, G_{j-2} \rangle \\
&= -\langle DG_{i+2}, G_{j-2} \rangle = \langle G_{i+2}, DG_{j-2} \rangle \\
&= \langle G_{i+2}, K_{j-2} \rangle \\
&= \cdots \\
&= \langle G_{i+j}, K_0 \rangle \\
&= \langle G_{i+j}, \Theta(1) \rangle = -\langle \Theta(G_{i+j}), 1 \rangle = -\langle K_{i+j+1}, 1 \rangle \\
&= 0,
\end{aligned}$$

where in the last step we have used the identity, which we already proved for (ii). This completes the proof of the theorem. □

The elements K_i and G_j, $i, j \in \mathbb{N}_0$, will also be referred to as the *members of the skew symmetric hierarchy* generated by the corresponding skew symmetric operator Θ.

Remark 3.20. In the notation of the above theorem we assume that $K_i \in \mathcal{A} \backslash \mathcal{F}$ for some $i \in \mathbb{N}$ and $\mathrm{ord}_u(K_i) > 0$. Then $K_i = 0$ is an ODE of order $\mathrm{ord}_u(K_i)$ in u.

Since $K_i \in \mathcal{N}$ by (ii), we can compute $D^{-1}K_i$ obtaining G_i and $G_i = c$, c some constant of integration, which reduces the order of the ODE under consideration by 1. Furthermore, if $K_j \in \mathcal{A} \setminus \mathcal{F}$ and $\operatorname{ord}_u(K_j) < \operatorname{ord}_u(G_i)$ for all $0 \leq j < i$, then by (iii) each of the elements K_j is an integrating factor for the ODE $G_i = c$.

Here the constant of integration on the right–hand–side of the reduced ODE is not a problem, since by the theorem $K_jG_i \in \mathcal{N}$ as well as $K_j \in \mathcal{N}$, such that multiplication of $G_i = c$ by K_j provides $K_jG_i = cK_j$ and both sides can be integrated resulting in another reduction of the order of the ODE under consideration.

Alternatively, we can proceed with the ODE $K_i = 0$ as follows: if $G_j \in \mathcal{A} \setminus \mathcal{F}$ and $\operatorname{ord}_u(G_j) < \operatorname{ord}_u(K_i)$, than by the theorem each of the G_j is an integrating factor for the ODE $K_i = 0$ and we can use these to reduce the order of $K_i = 0$ without reducing the order by first integrating K_i.

We discuss the application of Theorem 3.19 in concrete examples.

Example 3.21. Consider the skew symmetric operator

$$\Theta := D^3 + Du^3 + u^3D.$$

Following Theorem 3.19, we define

$$\begin{aligned}
K_0 :=& \quad \Theta(1) = && \quad 3u_xu^2, \\
G_0 :=& \quad D^{-1}K_0 = && \quad u^3, \\
K_1 :=& \quad \Theta(G_0) = && \quad 3u_{xxx}u^2 + 18u_{xx}u_xu + 9u_xu^5 + 6u_x^3, \\
G_1 :=& \quad D^{-1}K_1 = && \quad 3u_{xx}u^2 + 6u_x^2u + \frac{3}{2}u^6.
\end{aligned}$$

We do not present the results for K_i, G_i, $i \geq 2$, since the expressions arising from the computations become so large that it does not make sense to state them here. We will consider some more complex examples in the last part of the chapter[2]. We want to consider the ODE $K_1 = 0$, i.e.

$$3u_{xxx}u^2 + 18u_{xx}u_xu + 9u_xu^5 + 6u_x^3 = 0. \tag{3.3.1}$$

As discussed above, Theorem 3.19 provides two different strategies to successively reduce the order of (3.3.1):

[2]The expression swell — even after applying appropriate simplification algorithms to the arising expression — makes the use of a computer algebra system indispensable.

Strategy 1. Since K_1 is an element of $\mathcal{N}$, we can integrate it and obtain the reduced second order ODE $G_1 = c_1$, c_1 some constant of integration. In the next step we use K_0, which is an integrating factor for $G_1 = c_1$ and obtain a first order ODE, which can formally be written as $D^{-1}(K_0(G_1 - c_1)) = c_2$, c_2 another constant of integration.

Strategy 2. We do not integrate K_1, but use the integrating factors G_0 and G_1 for (3.3.1) to obtain two second order ODEs $D^{-1}(G_0 K_1) = c_3$ and $D^{-1}(G_1 K_1) = c_4$, where c_3 and c_4 are constants of integration. Then we solve one of the equations for u_{xx} and insert the resulting expression for u_{xx} into the second equation to obtain a first order ODE, which is then again a reduced form of (3.3.1).

We start to compute a reduction of (3.3.1) using Strategy 1. Integration of (3.3.1) provides the second order ODE

$$G_1 = 3u_{xx}u^2 + 6u_x^2 u + \frac{3}{2}u^6 = c_1,$$

where c_1 is a constant of integration. Multiplication of this equation by its integrating factor K_0 and afterwards integrating the arising expression gives

$$\frac{1}{2}u^9 + \frac{9}{2}u^4 u_x^2 - c_1 u^3 = c_2. \tag{3.3.2}$$

Now using Strategy 2 we multiply both sides of (3.3.1) first by G_0, integrate the arising expression and obtain

$$u^9 + 3u_{xx}u^5 + \frac{3}{2}u^4 u_x^2 = c_3, \tag{3.3.3}$$

c_3 some constant of integration. Using G_1 as integrating factor for (3.3.1) and proceeding analogously, we find

$$\frac{9}{8}u^{12} + \frac{9}{2}u^8 u_{xx} + 9u^7 u_x^2 + \frac{9}{2}u^4 u_{xx}^2 + 18u^3 u_x^2 u_{xx} + 18u^2 u_x^4 = c_4, \tag{3.3.4}$$

c_4 some constant of integration. From (3.3.3) we get

$$u_{xx} = -\frac{2u^9 + 3u^4 u_x^2 - 2c_3}{6u^5}.$$

Inserting this into (3.3.4) gives

$$\frac{4c_3^2 + 4c_3 u^9 + 36c_3 u^4 u_x^2 + u^{18} + 18u^{13}u_x^2 + 81u^8 u_x^4}{8u^6} = c_4. \tag{3.3.5}$$

In fact, both ODEs (3.3.2) and (3.3.5) are reductions of (3.3.1) giving a complete implicit characterization of its solutions. Choosing $c_1 = c_2 = c_3 = c_4 = 0$ (i.e. we only consider a special set of solutions by prescribing the values for the constants of integrations), we find that (3.3.2) provides the special reduction

$$u^9 + 9u^4u_x^2 = 0$$

and from (3.3.5) we obtain

$$u^{12} + 18u^7u_x^2 + 81u^2u_x^4 = 0,$$

where the left–hand–side of this last reduction can be obtained as the square of $u^6 + 9uu_x^2$, i.e. a factor of $u^9 + 9u^4u_x^2$. In fact, Strategy 1 provides an easier representation of the reduced ODE and (3.3.2) can be solved by the method of separation of variables, for example. $\diamond$

Remark 3.22. In the notation of Theorem 3.19, we consider the ODE $K_i = 0$, $K_i \in \mathcal{A} \setminus \mathcal{F}$, for some fixed $i > 0$. Assume that $\operatorname{ord}_u(K_i) = n > 1$ and $G_j \in \mathcal{A} \setminus \mathcal{F}$ for all $j \in \mathbb{N}_0$. By the Theorem 3.19 we know that $G_jK_i \in \mathcal{N}$ for all $j \in \mathbb{N}_0$. Whenever $\operatorname{ord}_u(G_j) < n$, then G_j is an integrating factor for $K_i = 0$ and the order of $K_i = 0$ can be reduced by 1.

But not only those elements G_j, such that $\operatorname{ord}_u(G_j) < n$, contribute a reduction of the order of $K_i = 0$. Assume that $G_lK_i \in \mathcal{N}$ for some fixed $l > i$ and $\operatorname{ord}_u(G_l) \geq n$, i.e. G_l is an extended integrating factor. If we can solve $K_i = 0$ for the highest order derivative of u with respect to x, then we can compute $D^{-1}(G_lK_i) = c$, c some constant of integration, and reduce $D^{-1}(G_lK_i) = c$ modulo $K_i = 0$, which again gives a conserved quantity for $K_i = 0$, i.e. a reduction of the order of $K_i = 0$ by 1.

This procedure is illustrated by the next example.

Example 3.23. Let

$$\Theta := D^3 + Du^2 + u^2D.$$

Due to Theorem 3.19, we compute K_1, G_1 and G_2:

$$\begin{aligned}
K_1 = \quad & \Theta(G_0) = & 2uu_{xxx} + 6u_xu_{xx} + 6u^3u_x, \\
G_1 = \quad & D^{-1}K_1 = & 2uu_{xx} + 2u_x^2 + \frac{3}{2}u^4, \\
G_2 = \quad & D^{-1}K_2 = & 2uu_{xxxx} + 8u_xu_{xxx} + 6u_{xx}^2 + 10u^3u_{xx} + 20u^2u_x^2 + \frac{5}{2}u^6.
\end{aligned}$$

We want to reduce the ODE

$$2uu_{xxx} + 6u_x u_{xx} + 6u^3 u_x = 0, \tag{3.3.6}$$

i.e. $K_1 = 0$, to a first order ODE using G_1 and G_2 (note that $\mathrm{ord}_u(G_2) = 4 > 3 = \mathrm{ord}_u(K_1)$, i.e. G_2 is an extended integrating factor). Therefore we first integrate K_1 and obtain the reduced second order ODE $G_1 = c_1$, c_1 some constant of integration. Additionally, we use the fact that $G_2 K_1 \in \mathcal{N}$, i.e. we first integrate $G_2 K_1$ obtaining $D^{-1}(G_2 K_1) = c_2$, c_2 some constant of integration, and then reduce $D^{-1}(G_2 K_1)$ modulo $K_1 = 0$ (since K_1 is linear in u_{xxx}, this can easily be done). This provides another conserved quantity for (3.3.6), i.e. we obtain the two second order ODEs:

$$\frac{3}{2}u^4 + 2u_{xx}u + 2u_x^2 = c_1, \tag{3.3.7}$$

$$\frac{1}{2}u^3(u^3 + 2u_{xx})(3u^4 + 4u_{xx}u + 4u_x^2) = c_2, \tag{3.3.8}$$

where c_1 and c_2 are constants of integration. Because of

$$\begin{aligned} 3u^4 + 4u_{xx}u + 4u_x^2 &= 2\Big(\frac{3}{2}u^4 + 2u_{xx}u + 2u_x^2\Big) \\ &= 2c_1, \end{aligned}$$

we find

$$\frac{3}{2}u^4 + 2u_{xx}u + 2u_x^2 = c_1, \tag{3.3.9}$$

$$c_1 u^3(u^3 + 2u_{xx}) = c_2. \tag{3.3.10}$$

We solve (3.3.9) for u_{xx} and insert the result for u_{xx} into (3.3.8) obtaining

$$c_1^2 u^2 - \frac{c_1}{2}u^6 - 2c_1 u^2 u_x^2 = c_2, \tag{3.3.11}$$

which is the desired reduction of (3.3.6) to a first order ODE. Equation (3.3.11) can now be solved for u_x and be treated by standard methods like, e.g., separation of variables. $\diamond$

Remark 3.24. It may happen that the reductions of $K_i = 0$, K_i a member of a skew symmetric hierarchy as introduced in Theorem 3.19, obtained by using the elements G_j, are not independent, i.e. not every equation of the set of reduced equations obtained in this way may be helpful to produce further reductions of the order of the ODE under consideration. This may lead to the fact that the

ODE under consideration cannot be reduced to a first order ODE or even be integrated completely. When the elements G_j computed due to Theorem 3.19 do not produce independent conserved quantities, by which further reductions of order can be reached, we speak of a *premature recurrence*. The following examples demonstrate this effect.

Example 3.25. Consider

$$\Theta := D^3 u + uD^3 + Du\varphi + \varphi u D,$$

where $\varphi = \varphi(x)$ is a generic element of $\mathcal{F}$, i.e. φ is treated as a symbolic function depending only on x in the following. With the help of Theorem 3.19 we obtain

$$\begin{aligned}
K_1 = \quad & \Theta(G_0) = \quad & u\varphi_x + \varphi u_x + u_{xxx}, \\
G_1 = \quad & D^{-1}K_1 = \quad & u\varphi + u_{xx}, \\
G_2 = \quad & D^{-1}K_2 = \quad & \frac{3}{2}u^2\varphi^2 + 2\varphi_{xx}u^2 + 5u\varphi u_{xx} + 5\varphi_x u u_x + \\
& & 2u_{xxxx}u + \varphi u_x^2 + u_{xxx}u_x + \frac{3}{2}u_{xx}^2.
\end{aligned}$$

Consider the ODE

$$u\varphi_x + \varphi u_x + u_{xxx} = 0, \tag{3.3.12}$$

i.e. $K_1 = 0$. Integration of this ODE provides $G_1 = c_1$, c_1 some constant of integration. Furthermore, multiplication of (3.3.12) by G_2, integration of the result and afterwards reduction modulo $K_1 = 0$ (which is again no problem, since K_1 is linear in u_{xxx}) provides the two conserved quantities

$$u\varphi + u_{xx} = c_1, \tag{3.3.13}$$

$$\frac{1}{2}(u\varphi + u_{xx})^3 = c_2, \tag{3.3.14}$$

for constants of integration c_1 and c_2. Since the left–hand–side of (3.3.14) is up to a constant multiple a power of the left–hand–side of (3.3.13), no further reduction of the order of (3.3.12) can be achieved. Even when computing G_3, G_4 and G_5 (for which, due to the large size of the expressions, we do not present the explicit results here) and proceeding in the same manner, we only find the trivial conserved quantities

$$\begin{aligned}
\frac{5}{8}(u\varphi + u_{xx})^4 &= c_3, \\
\frac{7}{8}(u\varphi + u_{xx})^5 &= c_4, \\
\frac{21}{16}(u\varphi + u_{xx})^6 &= c_5.
\end{aligned}$$

These results are not unexpected, since (3.3.12) is just the differentiated general second order homogeneous linear ODE (3.3.13), for which no general closed form solution for a generic function φ exists. $\diamond$

Example 3.26. A similar situation as discussed in Example 3.25 arises from the consideration of the skew symmetric operator

$$\Theta := D^3 + 2Du + 2uD.$$

Constructing the skew symmetric hierarchy in the sense of Theorem 3.19 with the help of Θ, we found no way to reduce the equation $K_3 = 0$, which reads in detail

$$\begin{aligned} 140\, u^3\, u_x + 70\, u_{xxx}\, u^2 + 280\, u_{xx}\, u\, u_x + 14\, u_{xxxxx}\, u + \\ 70\, u_x^3 + 42\, u_{xxxx}\, u_x + 70\, u_{xx}\, u_{xxx} + u_{xxxxxxx} = 0 \end{aligned}$$

to a first order ODE only using the integrating factors G_i of the skew symmetric hierarchy. We conjecture that the reason lies in the fact that this ODE admits further symmetries. For example, the member

$$K_2 = 30\, u_x\, u^2 + 10\, u_{xxx}\, u + 20\, u_x\, u_{xx} + u_{xxxxx}$$

of the skew symmetric hierarchy generated by Θ is the generator of a symmetry of the equation $K_3 = 0$. For such cases where sufficiently many symmetries can be detected from other members of the skew symmetric hierarchy under consideration, Theorem 1.26 by B. Fuchssteiner can be used to find an implicit characterization of the solutions of the ODE in terms of conserved quantities. $\diamond$

The characterization of those classes of skew symmetric operators, for which such recurrence phenomena appear, is still an open problem (see also Subsection 3.6.4 on open problems and further perspectives).

Remark 3.27. Further skew symmetric hierarchies can be obtained by using slight variations of Theorem 3.19. Assume, that Θ is a skew symmetric operator $\mathcal{A} \to \mathcal{A}$ with $\Theta(1) = 0$. Then a skew symmetric hierarchy for Θ can be obtained by choosing an arbitrary element $G_0 \in \mathcal{A}$ and defining

$$\begin{aligned} K_0 &:= DG_0, \\ K_i &:= \Theta(G_{i-1}), \qquad G_i := D^{-1}K_i, \end{aligned}$$

for all $i \in \mathbb{N}$. Then, as in Theorem 1.26, the following hold:

(i) $G_i \in \mathcal{A}$ for all $i \in \mathbb{N}_0$.

(ii) $K_i \in \mathcal{N}$ for all $i \in \mathbb{N}_0$.

(iii) $G_i K_j \in \mathcal{N}$ for all $i, j \in \mathbb{N}_0$.

The proof of this statement is similar to the proof of Theorem 3.19. Assertion (i) follows, when (ii) has been proved. To see that (ii) is correct, consider

$$\langle 1, K_0 \rangle = \langle 1, DG_0 \rangle = -\langle D1, G_0 \rangle = 0$$

and for $i \geq 1$

$$\langle 1, K_i \rangle = \langle 1, \Theta(G_{i-1}) \rangle = -\langle \Theta(1), G_{i-1} \rangle = -\langle 0, G_{i-1} \rangle = 0,$$

which proves that $K_i \in \mathcal{N}$ for all $i \in \mathbb{N}_0$. Hence, also $G_i = D^{-1} K_i$, $i \in \mathbb{N}$, are elements of the algebra $\mathcal{A}$. By the proof of (iii) in Theorem 3.19, we know that for $i, j \in \mathbb{N}$ we have

$$\langle G_i, K_j \rangle = -\langle K_{i+j}, G_0 \rangle.$$

Furthermore, we observe that

$$\begin{aligned} \langle G_i, K_j \rangle &= \langle G_i, DG_j \rangle = -\langle DG_i, G_j \rangle = -\langle K_i, G_j \rangle \\ &= -\langle \Theta(G_{i-1}), G_j \rangle = \langle G_{i-1}, \Theta(G_j) \rangle \\ &= \langle G_{i-1}, K_{j+1} \rangle. \end{aligned}$$

Proceeding in this manner, we obtain

$$\langle G_i, K_j \rangle = \langle G_{i-1}, K_{j+1} \rangle = \ldots = \langle G_0, K_{i+j} \rangle = \langle K_{i+j}, G_0 \rangle,$$

i.e. $-\langle K_{i+j}, G_0 \rangle = \langle K_{i+j}, G_0 \rangle$ and, hence, it follows $\langle G_i, K_j \rangle = 0$. This means $G_i K_j \in \mathcal{N}$.

Example 3.28. Remark 3.27 allows to create a non–trivial skew symmetric hierarchy using the skew symmetric operator

$$\Theta = \sum_{k=0}^{n} D^{2k+1}, \quad n \in \mathbb{N},$$

whereas using Θ in the construction of Theorem 3.19 only produces a set of constants. $\diamond$

Remark 3.29. Whereas Example 3.28 may not have been the most meaningful one, let us present now a related one, which is much more meaningful and which shows that the methods developed so far open a wide avenue for generalizations[3]. Consider some $G_0 \in \mathcal{A}$ and two skew symmetric operators $\Theta_0, \Theta_1 : \mathcal{A} \to \mathcal{A}$, such that additionally

(1) $(\Theta_0 - D)G_0 = 0$,

(2) $\Theta_0(1) = \Theta_1(1) = 0$.

If we consider

$$\Theta = \Theta_1 + \Theta_0 G_0 D^{-1} G_0 \Theta_0$$

and define $D^{-1}(0) = 0$, then for all $n \in \mathbb{N}_0$ the elements

$$\begin{aligned} K_n &= (\Theta D^{-1})^n D G_0, \\ G_n &= D^{-1}(\Theta D^{-1})^n D G_0 \end{aligned}$$

are well–defined elements of $\mathcal{A}$, such that

(i) $K_i \in \mathcal{N}$ for all $i \in \mathbb{N}_0$,

(ii) $G_i K_j \in \mathcal{N}$ for all $i, j \in \mathbb{N}_0$.

Of course, the surprising fact is that, although a lot of integral operators are involved, we never leave $\mathcal{A}$. This is surprising, because by definition Θ is not an operator in $\mathcal{A}$ at all.

Sketch of a proof. We define

$$\mathcal{A}_{G_0}^{\perp} := \{H \in \mathcal{A} \mid \langle G_0, \Theta_0(H)\rangle = 0\}.$$

Then Θ is a well–defined map from $\mathcal{A}_{G_0}^{\perp}$ to $\mathcal{A}$, because the integration within the operator can always be carried out. Furthermore, with respect to $\mathcal{A}_{G_0}^{\perp}$, the operator Θ is skew symmetric, i.e. for $H_1, H_2 \in \mathcal{A}_{G_0}^{\perp}$, we have

$$\langle H_1, \Theta(H_2)\rangle = -\langle H_2, \Theta(H_1)\rangle,$$

which is easily proved by integration by parts.

[3] Which however we shall not discuss here, since this would require that we leave the transparent and simple algebraic structure, we presented so far.

We proceed by induction: For a fixed $n \in \mathbb{N}$, we assume that $G_i, K_j \in \mathcal{A}$ for all $j \leq n, i \leq (n-1)$ and that (i) and (ii) from above are valid for these.

This assumption is true for $n = 1$, since due to the fact that $\Theta_0(G_0) = DG_0$, we have $K_0 = DG_0 \in \mathcal{N}$ and

$$K_1 = \Theta_1(G_0) + \frac{1}{2}\Theta_0(G_0^3),$$

which implies

$$\begin{aligned}
\langle 1, K_1 \rangle &= \langle 1, \Theta_1(G_0) \rangle + \frac{1}{2} \langle 1, \Theta_0(G_0^3) \rangle \\
&= -\langle \Theta_1(1), G_0 \rangle - \frac{1}{2} \langle \Theta_0(1), G_0^3 \rangle \\
&= 0.
\end{aligned}$$

The last equality comes from (2) above. Hence, $K_0, K_1 \in \mathcal{N}$. Because of

$$\langle G_0, K_0 \rangle = \langle G_0, DG_0 \rangle = 0$$

it follows $G_0\, K_0 \in \mathcal{N}$. Finally,

$$\begin{aligned}
\langle G_0, K_1 \rangle &= \underbrace{\langle G_0, \Theta_1(G_0) \rangle}_{=0} + \frac{1}{2} \langle G_0, \Theta_0(G_0^3) \rangle \\
&= -\frac{1}{2} \langle \Theta_0(G_0), G_0^3 \rangle = -\frac{1}{2} \langle DG_0, G_0^3 \rangle \\
&= 0,
\end{aligned}$$

which proves $G_0\, K_1 \in \mathcal{N}$.

By assumption we have $K_n \in \mathcal{N}$. Hence, $G_n = D^{-1}K_n$ is an element of $\mathcal{A}$. Furthermore,

$$\begin{aligned}
\langle G_0, \Theta_0(G_n) \rangle &= -\langle G_n, \Theta_0(G_0) \rangle = -\langle G_n, DG_0 \rangle \\
&= \langle G_0, DG_n \rangle = \langle G_0, K_n \rangle \\
&= 0
\end{aligned}$$

by assumption and (1) above. Therefore, $G_n \in \mathcal{A}_{G_0}^{\perp}$ and $K_{n+1} = \Theta(G_n)$ is well–defined. It follows

$$\langle 1, K_{n+1} \rangle = \langle 1, \Theta(G_n) \rangle = -\langle \Theta(1), G_n \rangle = 0,$$

i.e. $K_{n+1} \in \mathcal{N}$ as desired.

It remains to prove that

$$\langle G_n, K_n \rangle = 0, \quad \langle G_n, K_{n+1} \rangle = 0, \quad \langle G_{n-1}, K_{n+1} \rangle = 0 \tag{3.3.15}$$

as well as

$$\langle G_n, K_j \rangle = 0, \quad \langle G_i, K_{n+1} \rangle = 0 \tag{3.3.16}$$

for $j < n$, $i < (n-1)$. We observe that due to skew symmetry

$$\langle G_n, K_n \rangle = \langle G_n, DG_n \rangle = 0$$

and

$$\langle G_n, K_{n+1} \rangle = \langle G_n, \Theta(G_n) \rangle = 0$$

as well as

$$\begin{aligned} \langle G_{n-1}, K_{n+1} \rangle &= \langle G_{n-1}, \Theta(G_n) \rangle = -\langle \Theta(G_{n-1}), G_n \rangle \\ &= -\langle K_n, G_n \rangle = 0. \end{aligned}$$

Hence, (3.3.15) is proved. And (3.3.16) also follows by skew symmetry and the assumption (ii). We find:

$$\langle G_n, K_j \rangle = \langle G_n, DG_j \rangle = -\langle K_n, G_j \rangle = 0$$

and

$$\begin{aligned} \langle G_i, K_{n+1} \rangle &= \langle G_i, \Theta(G_n) \rangle = -\langle G_n, \Theta(G_i) \rangle \\ &= -\langle G_n, K_{i+1} \rangle = -\langle G_n, DG_{i+1} \rangle \\ &= \langle G_{i+1}, DG_n \rangle = \langle G_{i+1}, K_n \rangle \\ &= 0. \end{aligned}$$

Example 3.30. One of the examples, which can be generated due to Remark 3.29, is well–known from the theory of integrable partial differential equations. One takes $G_0 = u_x$ and

$$\Theta = D^3 + DG_0D^{-1}G_0D.$$

Then the elements K_i, $i \in \mathbb{N}_0$, generated by that are the well–known vector fields from the modified Korteweg–de Vries equation (see [25] and [26] for details). This also is an example, where premature recurrence occurs, this for the reason that we have additional symmetries. $\diamond$

The next section is dedicated to establish the notion of the fundamental form of a skew symmetric operator $\Theta : \mathcal{A} \to \mathcal{A}$, which will later on serve as an essential algorithmic tool for the efficient computation of the integrations, which have to be performed on the one hand to compute the skew symmetric hierarchies introduced in Theorem 3.19 and on the other hand to reduce the order of a given ODE as discussed in Examples 3.21, 3.23 and 3.25.

3.4 Fundamental forms of skew symmetric operators

From an algorithmical point of view, the computation of the G_i's as integrals over the K_i's in Theorem 3.19 is the most expensive (and difficult) part of all computations involved in building the skew symmetric hierarchies. In the following we introduce suitable quadratic forms, which facilitate these computations considerably, and which will lead to recursion formulas for the G_i's.

We start with some obvious but helpful fact:

Lemma 3.31. *Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric differential operator and $A, B \in \mathcal{A}$. Then $A\Theta(B) + B\Theta(A) \in \mathcal{N}$. In other words: there is a $P \in \mathcal{A}$ such that*

$$A\Theta(B) + B\Theta(A) = D(P).$$

Proof. We have for $T := A\Theta(B) + B\Theta(A)$ that

$$\langle 1, T\rangle = \langle A, \Theta(B)\rangle + \langle B, \Theta(A)\rangle,$$

which is zero, because Θ is skew symmetric. □

Independently of what we aim at with this lemma, in the context of algorithmic approaches this lemma also gives us new methods of constructing skew symmetric operators:

Remark 3.32. Let us denote by $(\Theta(A))$ the element of $\mathcal{A}$ given by the application of Θ to A. Therefore, as an operator, $(\Theta(A))$ is a multiplication operator. Then the operator

$$A\Theta + (\Theta(A))$$

maps $\mathcal{A}$ into $\mathcal{N}$. Hence, the formal integral operator

$$\Psi = (\Theta A - (\Theta(A)))D^{-1}(A\Theta + (\Theta(A)))$$

is a skew symmetric operator[4] in $\mathcal{A}$.

Proof. Since D^{-1} is not an operator in $\mathcal{A}$, we have to give a different representation for Ψ in order to see that the operator is skew symmetric. As shown in the Lemma above, we may find an operator $\Gamma : \mathcal{A} \to \mathcal{A}$ such that

$$A\Theta + (\Theta(A)) = D\Gamma.$$

Using this we may write Ψ as

$$\Psi = -(D\Gamma)^\star D^{-1}(D\Gamma) = \Gamma^\star D\Gamma,$$

where $\Gamma^\star$ denotes the transposed of Γ with respect to the given scalar product. The right–hand–side obviously is skew symmetric. □

Definition 3.33. Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric operator. For general elements $A, B \in \mathcal{A}$ we define a bilinear form

$$F_\Theta : \mathcal{A} \times \mathcal{A} \to \mathcal{A}, \quad F_\Theta(A,B) := D^{-1}(A\Theta(B)) + D^{-1}(B\Theta(A)).$$

Furthermore we consider the quadratic form

$$Q_\Theta : \mathcal{A} \to \mathcal{A}, \quad Q_\Theta(A) = F_\Theta(A,A).$$

We call F_Θ the *fundamental form* of the skew symmetric operator Θ and Q_Θ its *quadratic form*, respectively.

Note that $F_\Theta : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ is indeed mapping elements $A, B \in \mathcal{A}$ to an element of $\mathcal{A}$, since Lemma 3.31 guarantees that the integrals involved can be computed, such that no symbolic rest integrals remain.

Since F_Θ is a symmetric bilinear form, we can, as usual (parallelogram law), recover this from Q_Θ:

$$F_\Theta(A,B) = \frac{1}{2}\big(Q_\Theta(A+B) - Q_\Theta(A) - Q_\Theta(B)\big) \tag{3.4.1}$$

for all $A, B \in \mathcal{A}$. Therefore, for the computation of F_Θ we should make use of an algorithm for Q_Θ. And for this we use, after a change of algebra, the canonical form provided in Section 3.2.

[4]This way of constructing new skew symmetric operators is closely related to what we did in Remark 3.29 and Example 3.30.

For this consider $Q_\Theta(A(x))$ in the formal variable $A(x)$. We know that this is integrable and we take that as an element in the algebra $\mathcal{G}$, which is the algebra generated in the same way as $\mathcal{A}$, but where now the role of the $u(x)$ is taken by $A(x)$ instead, and where the $u(x)$ and other functions in $\mathcal{A}$ are considered as belonging to the function set. Then, since being integrable does not depend on this change of variable, we can now apply our algorithm, which then consists of integration by parts of the highest derivative $A^{(n)}(x)$ of $A(x)$ in the considered expression. Being integrable means that this derivative can never occur as a quadratic term (or a term of even higher polynomial order in this derivative).

The algorithm we obtain by application of the results of Section 3.2 is:

Algorithm 3.34. (Symbolic computation of the quadratic form) Let $\Theta : \mathcal{A} \to \mathcal{A}$ be an operator, and let $A(x) \in \mathcal{G}$ be a variable, i.e. an arbitrary symbolic element of $\mathcal{G}$.

1. Define
$$T := 2\,A(x)\,\Theta(A(x))$$
and
$$Q := 0.$$
T will be the term still to be integrated, whereas Q will be the result of integrations already performed. Both are elements of $\mathcal{G}$.

2. Now repeat the following steps, until $T = 0$ or the algorithm breaks off.

   - Let S be the term of T, which contains the highest order derivative $A^{(n)}(x)$ of $A(x)$ with respect to x. Then S is of the form
   $$S = A^{(n)}(x)\,A^{(m)}(x)\,C(x,u,u_x,u_{xx},\ldots),$$
   and $n > m$, since T is integrable. We make a case differentiation:
     - If $n > m - 1$, then we put
     $$Q := Q + A^{(n-1)}(x)A^{(m)}(x)\,C(x,u,u_x,u_{xx},\ldots),$$
     $$T := T - D\Big(A^{(n-1)}(x)\,A^{(m)}(x)\,C(x,u,u_x,u_{xx},\ldots)\Big).$$
     - If $n = m - 1$, then we put
     $$Q := Q + \frac{1}{2}A^{(n-1)}(x)\,A^{(n-1)}(x)\,C(x,u,u_x,u_{xx},\ldots),$$
     $$T := T - \frac{1}{2}D\Big(A^{(n-1)}(x)\,A^{(n-1)}(x)\,C(x,u,u_x,u_{xx},\ldots)\Big).$$

- ∘ If $n = m$, then we break off the algorithm.

3. Return Q as the result of the algorithm. △

The algorithm terminates and if Θ is skew symmetric, then the returned Q is

$$Q = D^{-1}(2A(x)\Theta(A(x)) = Q_\Theta(A(x))).$$

We briefly discuss the correctness of the algorithm.

Proof. A simple observation shows that during the algorithm we have at each step $T, Q \in \mathcal{G}$ and

$$D^{-1}(2A(x)\Theta(A(x))) = Q + D^{-1}(T). \tag{3.4.2}$$

Hence, when Θ is skew symmetric, then, since the left–hand–side is integrable, T always will stay integrable. This means that T never has its highest derivative in $A(x)$ as a quadratic term (or a term of even higher polynomial order in this derivative). So the algorithm only breaks off, if $T = 0$, but then by (3.4.2), Q obviously returns the desired integral of

$$2A(x)\Theta(A(x)).$$

The algorithm clearly terminates, because after each round of the steps of computation summarized under 2, the order of the highest derivative of $A(x)$ is reduced by 1. □

The following example demonstrates, how Algorithm 3.34 proceeds to compute the quadratic form of a given skew symmetric operator:

Example 3.35. Assume, we want to compute the quadratic form Q_Θ of the skew symmetric operator

$$\Theta = D^3 + Du^2 + u^2 D.$$

Using the same notation as in Algorithm 3.34, we perform the following steps of computation:

1. We define

$$\begin{aligned} T &:= 2\, A(x)\, \Theta(A(x)) \\ &= 4\, u\, u_x\, A(x)^2 + 4\, u^2\, A(x)\, A_x(x) + 2\, A(x)\, A_{xxx}(x), \\ Q &:= 0. \end{aligned}$$

2. Now we are at the beginning of the first round of the steps of computations summarized under 2 of the algorithm.

   - We define
     $$S := 2\,A(x)\,A_{xxx}(x),$$
     which is the term with the highest order derivative of $A(x)$ with respect to x in T, i.e. we are in the situation $n = 3$ and $m = 0$.
     - We compute
       $$\begin{aligned} Q :&= Q + 2\,A(x)\,A_{xx}(x) \\ &= 2\,A(x)\,A_{xx}(x), \end{aligned}$$
       $$\begin{aligned} T :&= T - D(2\,A(x)\,A_{xx}(x)) \\ &= 4\,u\,u_x\,A(x)^2 + 4\,u^2\,A(x)\,A_x(x) - 2\,A_x(x)\,A_{xx}(x). \end{aligned}$$
   - Now
     $$S := -2\,A_x(x)\,A_{xx}(x)$$
     is the term with the highest order derivative of $A(x)$ with respect to x in T, i.e. we are in the situation $n = 2$ and $m = 1$.
     - We compute
       $$\begin{aligned} Q :&= Q + \frac{1}{2}(-2\,A_x(x)\,A_x(x)) \\ &= 2\,A(x)\,A_{xx}(x) - A_x(x)^2, \end{aligned}$$
       $$\begin{aligned} T :&= T - \frac{1}{2}\,D(-2\,A_x(x)^2) \\ &= 4\,u\,u_x\,A(x)^2 + 4\,u^2\,A(x)\,A_x(x). \end{aligned}$$
   - At the current state of the computation,
     $$S := 4\,u^2\,A(x)\,A_x(x)$$
     is the term with the highest order derivative of $A(x)$ with respect to x in T, i.e. we are in the situation $n = 1$ and $m = 0$.
     - We compute
       $$\begin{aligned} Q :&= Q + \frac{1}{2}(4\,u^2 A(x)^2) \\ &= 2\,u^2\,A(x)^2 + 2\,A(x)\,A_{xx}(x) - A_x(x)^2, \end{aligned}$$
       $$\begin{aligned} T :&= T - \frac{1}{2}\,D(4\,u^2\,A(x)^2) \\ &= 0. \end{aligned}$$

3. We finally obtain

$$Q = 2\,u^2\,A(x)^2 + 2\,A(x)\,A_{xx}(x) - A_x(x)^2.$$

Indeed, Q gives the desired result for $Q_\Theta(A(x))$. $\diamond$

Algorithm 3.34 and application of (3.4.1) then directly leads to the following algorithm for computing the fundamental form of a skew symmetric operator:

Algorithm 3.36. (Symbolic computation of the fundamental form) Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric operator and $A_1, A_2 \in \mathcal{A}$.

1. Compute the quadratic form $Q_\Theta(A(x))$, $A(x) \in \mathcal{G}$, with the help of Algorithm 3.34.

2. Compute

$$F_\Theta(A_1, A_2) = \frac{1}{2}\big(Q_\Theta(A_1 + A_2) - Q_\Theta(A_1) - Q_\Theta(A_2)\big)$$

   by simply inserting the concrete elements A_1, A_2 and $A_1 + A_2$ into the symbolic representation obtained for Q_Θ in step 1. $\triangle$

Proof. The correctness of the algorithm follows directly from the correctness of Algorithm 3.34 and (3.4.1). $\square$

Example 3.37. Let us again consider the skew symmetric differential operator

$$\Theta = D^3 + Du^2 + u^2 D.$$

In Example 3.35, we saw that

$$Q_\Theta(A(x)) = 2\,u^2\,A(x)^2 + 2\,A(x)\,A_{xx}(x) - A_x(x)^2.$$

Hence, using Algorithm 3.36 to compute $F_\Theta(A_1, A_2)$, where $A_1, A_2 \in \mathcal{A}$, we find

$$F_\Theta(A_1, A_2) = 2\,A_1\,A_2\,u^2 + A_1\,A_{2xx} + A_{1xx}\,A_2 - A_{1x}A_{2x},$$

which is the desired result. $\diamond$

3.5 Recursion formulas for symbolic integration

We now focus our attention on the computation of the elements G_i introduced in Theorem 3.19. The fundamental form F_Θ of a skew symmetric operator $\Theta : \mathcal{A} \to \mathcal{A}$ will help us to do so. The next example gives a motivation.

Example 3.38. We consider the situation of Theorem 3.19, i.e. $\Theta : \mathcal{A} \to \mathcal{A}$ denotes a skew symmetric operator and $K_0 := \Theta(1)$, $G_0 := D^{-1}K_0$, $K_1 = \Theta(G_0)$. We want to compute G_1 with the help of the fundamental form F_Θ of the skew symmetric operator Θ. We have:

$$\begin{aligned} F_\Theta(1, G_0) &= D^{-1}(1\Theta(G_0)) + D^{-1}(G_0\Theta(1)) \\ &= D^{-1}(K_1) + D^{-1}(G_0K_0) \\ &= G_1 + D^{-1}(G_0K_0) \end{aligned}$$

Because of $K_0 = DG_0$, we can compute the integral $D^{-1}(G_0K_0)$ with the help of integration by parts:

$$\begin{aligned} D^{-1}(G_0K_0) &= D^{-1}(G_0DG_0) \\ &= G_0^2 - D^{-1}(DG_0G_0) \\ &= G_0^2 - D^{-1}(G_0K_0), \end{aligned}$$

i.e. $D^{-1}(G_0K_0) = \frac{1}{2}G_0^2$. Hence, we obtain:

$$G_1 = F_\Theta(1, G_0) - \frac{1}{2}G_0^2. \tag{3.5.1}$$

Again, it is easy to compute K_2 via $K_2 = \Theta(G_1)$, since Θ involves no integrations to be computed when applying it to an element of $\mathcal{A}$. For the computation of G_2 we proceed similarly to the way we did above:

$$\begin{aligned} F_\Theta(1, G_1) &= D^{-1}(1\Theta(G_1)) + D^{-1}(G_1\Theta(1)) \\ &= D^{-1}(K_2) + D^{-1}(G_1K_0) \\ &= G_2 + D^{-1}(G_1K_0). \end{aligned}$$

The integral $D^{-1}(G_1K_0)$ can again be computed with the help of integration by parts, which provides:

$$\begin{aligned} D^{-1}(G_1K_0) &= D^{-1}(G_1DG_0) \\ &= G_0G_1 - D^{-1}(D(G_1)G_0) \\ &= G_0G_1 - D^{-1}(G_0K_1). \end{aligned}$$

The integral $D^{-1}(G_0K_1)$ can be expressed in terms of the fundamental form F_Θ applied to the pair (G_0, G_0):

$$\begin{aligned} F_\Theta(G_0, G_0) &= 2D^{-1}(G_0\Theta(G_0)) \\ &= 2D^{-1}(G_0K_1), \end{aligned}$$

hence, $D^{-1}(G_0K_1) = \frac{1}{2}F_\Theta(G_0, G_0)$ and finally we have

$$G_2 = F_\Theta(1, G_1) + \frac{1}{2}F_\Theta(G_0, G_0) - G_0G_1. \tag{3.5.2}$$

Equations (3.5.1) and (3.5.2) suggest that we can compute G_i in terms of G_j, $j < i$, and the fundamental form F_Θ. $\diamond$

Before we come to a generalization of the results presented in the preceding example, we state the following lemma, which we will need in the proof of the theorem on recursion formulas for the computation of the elements G_i of a skew symmetric hierarchy:

Lemma 3.39. *Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric operator and $K_i, G_i \in \mathcal{A}$, $i \in \mathbb{N}_0$, as defined in Theorem 3.19. Then for all $i \in \mathbb{N}_0$:*

(i) $D^{-1}(G_iK_{i+1}) = \dfrac{1}{2}F_\Theta(G_i, G_i)$.

(ii) $D^{-1}(G_iK_i) = \dfrac{1}{2}G_i^2$.

(iii) $G_{i+1} = F_\Theta(1, G_i) - D^{-1}(G_iK_0)$.

Proof. The assertions of the lemma follow directly from the definitions of the K_i's and G_i's and from integration by parts:

$$\begin{aligned} F_\Theta(G_i, G_i) &= 2D^{-1}(G_i\Theta(G_i)) \\ &= 2D^{-1}(G_iK_{i+1}), \end{aligned}$$

which proves (i). Furthermore, we have:

$$D^{-1}(G_iK_i) = D^{-1}(G_iDG_i) = \frac{1}{2}G_i^2.$$

This proves (ii). The assertion in (iii) follows directly from the definition of the fundamental form of the skew symmetric operator Θ:

$$\begin{aligned} F_\Theta(1, G_i) &= D^{-1}(1\Theta(G_i)) + D^{-1}(G_i\Theta(1)) \\ &= D^{-1}(K_{i+1}) + D^{-1}(G_iK_0) \\ &= G_{i+1} + D^{-1}(G_iK_0), \end{aligned}$$

i.e. $G_{i+1} = F_\Theta(1, G_i) - D^{-1}(G_iK_0)$. This completes the proof of the lemma. $\square$

The following theorem generalizes the results of Example 3.38 and provides the basis for a more elegant and efficient algorithm for computing integrating factors and reductions of ODEs arising from a skew symmetric hierarchy:

Theorem 3.40. (Recursion formula to compute the G_i's) *Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric operator and $K_i, G_i \in \mathcal{A}$, $i \in \mathbb{N}_0$ as defined in Theorem 3.19. Let $F_\Theta : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ be the fundamental form of the skew symmetric operator Θ. Then G_{i+1} can be computed with the help of F_Θ and G_j, $0 \leq j \leq i$, for all $i \in \mathbb{N}_0$.*

More precisely, if we formally put $G_{-1} = 1$, the following recursion formula holds:

$$G_{i+1} = \frac{1}{2} \sum_{k=-1}^{i} F_\Theta(G_k, G_{i-1-k}) - \frac{1}{2} \sum_{k=0}^{i} G_k\, G_{i-k}. \tag{3.5.3}$$

Proof. By Lemma 3.39 (iii) we have:

$$G_{i+1} = F_\Theta(1, G_i) - D^{-1}(G_i\, K_0),$$

i.e. it remains to show that

$$D^{-1}(G_i\, K_0) = \frac{1}{2} \sum_{k=0}^{i} G_k\, G_{i-k} - \frac{1}{2} \sum_{k=0}^{i-1} F_\Theta(G_k, G_{i-1-k}).$$

Integration by parts provides:

$$\begin{aligned} D^{-1}(G_i K_0) &= D^{-1}(G_i D G_0) \\ &= G_0 G_i - D^{-1}(D(G_i) G_0) \\ &= G_0 G_i - D^{-1}(K_i G_0). \end{aligned}$$

Furthermore

$$\begin{aligned} F_\Theta(G_0, G_{i-1}) &= D^{-1}(G_0 \Theta(G_{i-1})) + D^{-1}(\Theta(G_0) G_{i-1}) \\ &= D^{-1}(G_0 K_i) + D^{-1}(K_1 G_{i-1}), \end{aligned}$$

which shows $D^{-1}(K_i G_0) = F_\Theta(G_0, G_{i-1}) - D^{-1}(G_{i-1} K_1)$. To summarize, we have found:

$$D^{-1}(G_i\, K_0) = G_0 G_i - F_\Theta(G_0, G_{i-1}) + D^{-1}(G_{i-1} K_1).$$

We continue proceeding this way:

- $D^{-1}(G_{j-l}K_l)$ is computed by integration by parts, first, and
- the remaining integral $D^{-1}(K_{j-l}G_l)$ is expressed by $F_\Theta(G_l, G_{j-l-1}) - D^{-1}(G_{j-l-1}K_{l+1})$.

Depending on the fact, whether i is an even or an odd number, two different cases have to be distinguished.

We first consider the case, where i is an even number. We can apply the above procedure as long as we obtain the following representation of $D^{-1}(G_i\, K_0)$:

$$\begin{aligned} D^{-1}(G_iK_0) = {} & G_0G_i + G_1G_{i-1} + \ldots + G_{\frac{i-2}{2}}G_{\frac{i+2}{2}} - \\ & F_\Theta(G_0, G_{i-1}) - F_\Theta(G_1, G_{i-2}) - \ldots - F_\Theta\Big(G_{\frac{i-2}{2}}, G_{\frac{i}{2}}\Big) + \\ & D^{-1}\Big(G_{\frac{i}{2}}K_{\frac{i}{2}}\Big). \end{aligned}$$

By Lemma 3.39 (ii) we have $D^{-1}(G_{\frac{i}{2}}K_{\frac{i}{2}}) = \frac{1}{2}G^2_{\frac{i}{2}}$. Thus, it follows:

$$\begin{aligned} D^{-1}(G_iK_0) &= G_0G_i + G_1G_{i-1} + \ldots + G_{\frac{i-2}{2}}G_{\frac{i+2}{2}} + \frac{1}{2}G^2_{\frac{i}{2}} - \\ & \quad F_\Theta(G_0, G_{i-1}) - F_\Theta(G_1, G_{i-2}) - \ldots - F_\Theta\Big(G_{\frac{i-2}{2}}, G_{\frac{i}{2}}\Big) \\ &= \frac{1}{2}\sum_{k=0}^{i} G_k\, G_{i-k} - \frac{1}{2}\sum_{k=0}^{i-1} F_\Theta(G_k, G_{i-1-k}). \end{aligned}$$

If i is an odd number, we can apply the above procedure as long as we obtain the following representation of $D^{-1}(G_i\, K_0)$:

$$\begin{aligned} D^{-1}(G_iK_0) = {} & G_0G_i + G_1G_{i-1} + \ldots + G_{\frac{i-1}{2}}G_{\frac{i+1}{2}} - \\ & F_\Theta(G_0, G_{i-1}) - F_\Theta(G_1, G_{i-2}) - \ldots - F_\Theta\Big(G_{\frac{i-3}{2}}, G_{\frac{i+1}{2}}\Big) - \\ & F_\Theta\Big(G_{\frac{i-1}{2}}, G_{\frac{i-1}{2}}\Big) + D^{-1}\Big(G_{\frac{i-1}{2}}K_{\frac{i+1}{2}}\Big). \end{aligned}$$

By Lemma 3.39 (i) we have $D^{-1}(G_{\frac{i-1}{2}}K_{\frac{i+1}{2}}) = \frac{1}{2}F_\Theta(G_{\frac{i-1}{2}}, G_{\frac{i-1}{2}})$. Hence, in this case, it follows:

$$\begin{aligned} D^{-1}(G_i K_0) = & \; G_0 G_i + G_1 G_{i-1} + \ldots + G_{\frac{i-1}{2}} G_{\frac{i+1}{2}} - \\ & F_\Theta(G_0, G_{i-1}) - F_\Theta(G_1, G_{i-2}) - \ldots - F_\Theta\Big(G_{\frac{i-3}{2}}, G_{\frac{i+1}{2}}\Big) - \\ & \frac{1}{2} F_\Theta\Big(G_{\frac{i-1}{2}}, G_{\frac{i-1}{2}}\Big) \\ = & \; \frac{1}{2} \sum_{k=0}^{i} G_k \, G_{i-k} - \frac{1}{2} \sum_{k=0}^{i-1} F_\Theta(G_k, G_{i-1-k}). \end{aligned}$$

This proves (3.5.3) and we are done. □

Example 3.41. Consider the skew symmetric operator $\Theta := D^3 + u^2 D + D u^2$ and

$$\begin{aligned} K_0 &= \Theta(1) = 2 u u_x, \\ G_0 &= D^{-1}(K_0) = u^2. \end{aligned}$$

We now compute G_1, G_2, G_3 and G_4 with the help of the new recursion formula (3.5.3). The fundamental form of Θ for arbitrary symbolic elements $A, B \in \mathcal{A}$ is given by

$$F_\Theta(A, B) = 2ABu^2 + AB_{xx} + BA_{xx} - A_x B_x,$$

which we already saw in Example 3.37. The recursion formula (3.5.3) provides:

$$\begin{aligned} G_1 &= F_\Theta(1, G_0) - \frac{1}{2} G_0^2 \\ &= \frac{3u^4}{2} + 2u_{xx} u + 2u_x^2, \end{aligned}$$

$$\begin{aligned} G_2 &= F_\Theta(1, G_1) - G_0 G_1 + \frac{1}{2} F_\Theta(G_0, G_0) \\ &= \frac{5u^6}{2} + 10u^3 u_{xx} + 20u^2 u_x^2 + 2u_{xxxx} u + 8u_{xxx} u_x + 6x_{xx}^2, \end{aligned}$$

$$\begin{aligned} G_3 = & \; F_\Theta(1, G_2) - G_0 G_2 - \frac{1}{2} G_1^2 + F_\Theta(G_0, G_1) \\ = & \; \frac{35u^8}{8} + 35u^5 u_{xx} + 105u^4 u_x^2 + 14u_{xxxx} u^3 + 112u^2 u_x u_{xxx} + \\ & 84u^2 u_{xx}^2 + 252 u u_x^2 u_{xx} + 2u_{xxxxxx} u + 42u_x^4 + 12u_{xxxxx} u_x + \\ & 30u_{xxxx} u_{xx} + 20u_{xxx}^2, \end{aligned}$$

$$
\begin{aligned}
G_4 &= F_\Theta(1, G_3) - G_0G_3 - G_1G_2 + F_\Theta(G_0, G_2) + \frac{1}{2}F_\Theta(G_1, G_1) \\
&= \frac{63u^{10}}{8} + 105u^7u_{xx} + 420u^6u_x^2 + 63u^5u_{xxxx} + 756u^4u_xu_{xxx} + \\
&\quad 567u^4u_{xx}^2 + 3192u^3u_x^2u_{xx} + 18u_{xxxxxx}u^3 + 1302u^2u_x^4 + \\
&\quad 216u_{xxxxx}u^2u_x + 498u^2u_{xx}u_{xxxx} + 318u^2u_{xxx}^2 + 768uu_x^2u_{xxxx} + \\
&\quad 2820uu_xu_{xx}u_{xxx} + 684uu_{xx}^3 + 2u_{xxxxxxxx}u + 912u_x^3u_{xxx} + \\
&\quad 1926u_x^2u_{xx}^2 + 16u_{xxxxxxx}u_x + 56u_{xxxxxx}u_{xx} + 112u_{xxxxx}u_{xxx} + \\
&\quad 70u_{xxxx}^2.
\end{aligned}
$$

These results are the same as if G_1, G_2, G_3 and G_4 are computed directly via integration of K_1, K_2, K_3 and K_4.

Alternatively and for comparison, Algorithm 3.15 can be used to perform the necessary computations (these computations are just the determination of the canonical forms of the expressions $D^{-1}(K_1), D^{-1}(K_2), D^{-1}(K_3)$ and $D^{-1}(K_4)$). $\diamond$

Theorem 3.40 provides an efficient method to compute the G_i's with the help of a recursion formula. This formula makes use of the fundamental form F_Θ of the skew symmetric operator $\Theta : \mathcal{A} \to \mathcal{A}$. In the following we will see that we can even give such recursion formulas to compute the integrals $D^{-1}(G_iK_j)$ with the help of the fundamental form F_Θ. This means that we can integrate equations of the form $G_iK_j = 0$ — and thereby produce reductions of the ODE $K_j = 0$ — as efficiently as we can compute the integrating factors G_i due to Theorem 3.40. The recursion formulas to compute the integrals $D^{-1}(G_iK_j)$ are summarized in the following theorem:

Theorem 3.42. (Recursion formulas for the computation of $D^{-1}(G_iK_j)$) *Let $\Theta : \mathcal{A} \to \mathcal{A}$ be a skew symmetric operator and $K_i, G_i \in \mathcal{A}$, $i \in \mathbb{N}_0$, as defined in Theorem 3.19. Let $F_\Theta : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ be the fundamental form of the skew symmetric operator Θ.*

Then for $i, j \in \mathbb{N}_0$, the integrals $D^{-1}(G_iK_j)$ are given by:

(i) If $j = i$, then $D^{-1}(G_i\,K_i)$ is obtained due to Lemma 3.39 (ii).

(ii) If $j > i$, then

$$
D^{-1}(G_i\,K_j) = \frac{1}{2} \sum_{\substack{r+s=i+j-1 \\ i \le r \le j-1}} F_\Theta(G_r, G_s) - \frac{1}{2} \sum_{\substack{r+s=i+j \\ i+1 \le r \le j-1}} G_r\,G_s. \tag{3.5.4}
$$

(iii) If $j < i$, then

$$D^{-1}(G_i\, K_j) = G_i\, G_j - D^{-1}(G_j K_i) \tag{3.5.5}$$

and $D^{-1}(G_j K_i)$ is computed due to (ii).

Proof. We only have to prove (ii) and (iii).

(ii) For $j = i+1$, (3.5.4) gives $\frac{1}{2}F_\Theta(G_i, G_i)$, which is the correct result due to Lemma 3.39 (i). Let now $j > i+1$. We have:

$$\begin{aligned} F_\Theta(G_i, G_{j-1}) &= D^{-1}(G_i \Theta(G_{j-1})) + D^{-1}(\Theta(G_i) G_{j-1}) \\ &= D^{-1}(G_i K_j) + D^{-1}(G_{j-1} K_{i+1}), \end{aligned}$$

i.e. $D^{-1}(G_i K_j) = F_\Theta(G_i, G_{j-1}) - D^{-1}(G_{j-1}, K_{i+1})$. With the help of integration by parts, we obtain:

$$\begin{aligned} D^{-1}(G_{j-1} K_{i+1}) &= D^{-1}(G_{j-1} D G_{i+1}) \\ &= G_{i+1} G_{j-1} - D^{-1}(D(G_{j-1}) G_{i+1}) \\ &= G_{i+1} G_{j-1} - D^{-1}(G_{i+1} K_{j-1}), \end{aligned}$$

and, hence, all in all

$$D^{-1}(G_i K_j) = F_\Theta(G_i, G_{j-1}) - G_{i+1} G_{j-1} + D^{-1}(G_{i+1} K_{j-1}).$$

The above two steps can now be applied successively again and again. In general we have for suitable $n \in \mathbb{N}$:

$$D^{-1}(G_{i+n} K_{j-n}) = F_\Theta(G_{i+n}, G_{j-n-1}) - D^{-1}(G_{j-n-1} K_{i+n+1}),$$

as well as

$$D^{-1}(G_{j-n-1} K_{i+n+1}) = G_{i+n+1} G_{j-n-1} - D^{-1}(G_{i+n+1} K_{j-n-1}),$$

from which we obtain the representation

$$D^{-1}(G_{i+n} K_{j-n}) = F_\Theta(G_{i+n}, G_{j-n-1}) - G_{i+n+1} G_{j-n-1} + D^{-1}(G_{i+n+1} K_{j-n-1}).$$

Successive application of the above formulas provides the representation

$$\begin{aligned} D^{-1}(G_i K_j) = \sum_{k=0}^{m-1} F_\Theta(G_{i+k}, G_{j-1-k}) - \sum_{k=0}^{m-1} G_{i+k+1} G_{j-k-1} + \\ D^{-1}(G_{i+m} K_{j-m}), \end{aligned} \tag{3.5.6}$$

where $m = \lfloor \frac{j-i}{2} \rfloor$.

If $j-i$ is an even number, it follows $i+m = \frac{i+j}{2}$ and $j-m = \frac{i+j}{2}$ and we have

$$D^{-1}(G_{i+m}K_{j-m}) = D^{-1}\Big(G_{\frac{i+j}{2}}K_{\frac{i+j}{2}}\Big) = \frac{1}{2}G^2_{\frac{i+j}{2}} \tag{3.5.7}$$

by Lemma 3.39 (ii).

If $j-i$ is an odd number, it follows $i+m = \frac{i+j-1}{2}$ and $j-m = \frac{i+j+1}{2}$ and we have

$$D^{-1}(G_{i+m}K_{j-m}) = D^{-1}\Big(G_{\frac{i+j-1}{2}}K_{\frac{i+j+1}{2}}\Big) = \frac{1}{2}F_\Theta\Big(G_{\frac{i+j-1}{2}}, G_{\frac{i+j-1}{2}}\Big) \tag{3.5.8}$$

by Lemma 3.39 (i).

Summarizing the two cases (3.5.7) and (3.5.8) for the integral $D^{-1}(G_{i+m}K_{j-m})$ in (3.5.6), we can rewrite (3.5.6) in the compact form

$$D^{-1}(G_iK_j) = \frac{1}{2} \sum_{\substack{r+s=i+j-1 \\ i \le r \le j-1}} F_\Theta(G_r, G_s) - \frac{1}{2} \sum_{\substack{r+s=i+j \\ i+1 \le r \le j-1}} G_r\, G_s,$$

which proves (3.5.4).

(iii) Let now $i > j$. Integration by parts provides

$$\begin{aligned} D^{-1}(G_iK_j) &= D^{-1}(G_i(DG_j)) \\ &= G_iG_j - D^{-1}(G_j(DG_i)) \\ &= G_iG_j - D^{-1}(G_jK_i), \end{aligned}$$

which proves (3.5.5). □

Example 3.43. We consider the same situation as in Example 3.41, i.e.

$$\Theta := D^3 + u^2D + Du^2$$

and

$$F_\Theta(A,B) = 2ABu^2 + AB_{xx} + BA_{xx} - A_xB_x.$$

We demonstrate the results of Theorem 3.42 in order to compute $D^{-1}(G_2K_2)$, $D^{-1}(G_2K_3)$, $D^{-1}(G_4K_0)$ and $D^{-1}(G_0K_4)$.

For the results for K_0, G_0, G_1, G_2, G_3 and G_4 we also refer to Example 3.41. According to Theorem 3.19, we obtain for K_1, K_2, K_3 and K_4:

$$\begin{aligned} K_1 &= \Theta(G_0) \\ &= 6\,u_x\,u^3 + 2\,u_{xxx}\,u + 6\,u_x u_{xx}, \end{aligned}$$

$$\begin{aligned} K_2 &= \Theta(G_1) \\ &= 15\,u^5\,u_x + 10\,u_{xxx}\,u^3 + 70\,u_{xx}\,u^2\,u_x + 40\,u\,u_x^3 + 2\,u_{xxxxx}\,u + \\ &\quad 10\,u_{xxxx}\,u_x + 20\,u_{xx}\,u_{xxx}, \end{aligned}$$

$$\begin{aligned} K_3 &= \Theta(G_2) \\ &= 35\,u^7\,u_x + 35\,u_{xxx}\,u^5 + 385\,u^4\,u_x\,u_{xx} + 420\,u^3\,u_x^3 + 14\,u_{xxxxx}\,u^3 + \\ &\quad 154\,u_{xxxx}\,u^2\,u_x + 280\,u_{xxx}\,u^2\,u_{xx} + 476\,u_{xxx}\,u\,u_x^2 + 672\,u\,u_x\,u_{xx}^2 + \\ &\quad 2\,u_{xxxxxxx}\,u + 420\,u_x^3\,u_{xx} + 14\,u_{xxxxxx}\,u_x + 42\,u_{xxxxx}\,u_{xx} + 70\,u_{xxx}\,u_{xxxx}, \end{aligned}$$

$$\begin{aligned} K_4 &= \Theta(G_3) \\ &= \frac{315}{4}\,u^9\,u_x + 105\,u^7\,u_{xxx} + 1575\,u^6\,u_x\,u_{xx} + 2520\,u^5\,u_x^3 + 63\,u_{xxxxx}\,u^5 + \\ &\quad 1071\,u_{xxxx}\,u^4\,u_x + 1890\,u^4\,u_{xx}\,u_{xxx} + 6216\,u^3\,u_x^2\,u_{xxx} + 8652\,u^3\,u_x\,u_{xx}^2 + \\ &\quad 18\,u_{xxxxxxx}\,u^3 + 14784\,u^2\,u_x^3\,u_{xx} + 270\,u_{xxxxxx}\,u^2\,u_x + 714\,u_{xxxxx}\,u^2\,u_{xx} + \\ &\quad 1134\,u_{xxxx}\,u^2\,u_{xxx} + 2604\,u\,u_x^5 + 1200\,u_{xxxxx}\,u\,u_x^2 + 5352\,u_{xxxx}\,u\,u_x\,u_{xx} + \\ &\quad 3456\,u\,u_x\,u_{xxx}^2 + 4872\,u\,u_{xx}^2\,u_{xxx} + 2\,u_{xxxxxxxxx}\,u + 1680\,u_{xxxx}\,u_x^3 + \\ &\quad 9408\,u_x^2\,u_{xx}\,u_{xxx} + 4536\,u_x\,u_{xx}^3 + 18\,u_{xxxxxxxx}\,u_x + 72\,u_{xxxxxxx}\,u_{xx} + \\ &\quad 168\,u_{xxxxxx}\,u_{xxx} + 252\,u_{xxxx}\,u_{xxxxx}. \end{aligned}$$

Application of Theorem 3.42 provides for the case $i = j = 2$:

$$
\begin{aligned}
D^{-1}(G_2K_2) &= \frac{1}{2}G_2^2 \\
&= \frac{25}{8}\,u^{12} + 25\,u^9\,u_{xx} + 50\,u^8\,u_x^2 + 5\,u^7\,u_{xxxx} + 20\,u^6\,u_x\,u_{xxx} + \\
&\quad 200\,u^5\,u_x^2\,u_{xx} + 200\,u^4\,u_x^4 + 20\,u^4\,u_{xx}\,u_{xxxx} + 40\,u^3\,u_x^2\,u_{xxxx} + \\
&\quad 80\,u^3\,u_x\,u_{xx}\,u_{xxx} + 160\,u^2\,u_x^3\,u_{xxx} + 120\,u^2\,u_x^2\,u_{xx}^2 + 2\,u^2\,u_{xxxx}^2 + \\
&\quad 16\,u\,u_x\,u_{xxx}\,u_{xxxx} + 12\,u\,u_{xx}^2\,u_{xxxx} + 32\,u_x^2\,u_{xxx}^2 + 48\,u_x\,u_{xx}^2\,u_{xxx} + \\
&\quad 60\,u^3\,u_{xx}^3 + 65\,u^6\,u_{xx}^2 + 18\,u_{xx}^4.
\end{aligned}
$$

Due to Theorem 3.42, we find for $i = 2$ and $j = 3$:

$$
\begin{aligned}
D^{-1}(G_2K_3) &= \frac{1}{2}F_\Theta(G_2, G_2) \\
&= \frac{25}{4}\,u^{14} + \frac{175}{2}\,u^{11}\,u_{xx} + 175\,u^{10}\,u_x^2 + 35\,u^9\,u_{xxxx} + \\
&\quad 140\,u^8\,u_x\,u_{xxx} + 455\,u^8\,u_{xx}^2 + 1050\,u^7\,u_x^2\,u_{xx} + 5\,u_{xxxxxx}\,u^7 + \\
&\quad 245\,u^6\,u_{xx}\,u_{xxxx} + 280\,u^5\,u_x^2\,u_{xxxx} + 280\,u^5\,u_x\,u_{xx}\,u_{xxx} + \\
&\quad 910\,u^5\,u_{xx}^3 + 2520\,u^4\,u_x^3\,u_{xxx} + 2240\,u^4\,u_x^2\,u_{xx}^2 + 20\,u_{xxxxxx}\,u^4\,u_{xx} - \\
&\quad 20\,u^4\,u_{xxx}\,u_{xxxxx} + 24\,u^4\,u_{xxxx}^2 + 2800\,u^3\,u_x^4\,u_{xx} + 40\,u_{xxxxxx}\,u^3\,u_x^2 - \\
&\quad 20\,u^3\,u_x\,u_{xx}\,u_{xxxxx} + 212\,u^3\,u_x\,u_{xxx}\,u_{xxxx} + 524\,u^3\,u_{xx}^2\,u_{xxxx} + \\
&\quad 160\,u^2\,u_x^3\,u_{xxxxx} + 420\,u^2\,u_x^2\,u_{xx}\,u_{xxxx} + 1264\,u^2\,u_x^2\,u_{xxx}^2 - \\
&\quad 144\,u^2\,u_x\,u_{xx}^2\,u_{xxx} + 456\,u^2\,u_{xx}^4 + 4\,u_{xxxxxx}\,u^2\,u_{xxxx} - 2\,u^2\,u_{xxxxx}^2 - \\
&\quad 320\,u\,u_x^4\,u_{xxxx} + 1280\,u\,u_x^3\,u_{xx}\,u_{xxx} + 1560\,u\,u_x^2\,u_{xx}^3 + \\
&\quad 16\,u_{xxxxxx}\,u\,u_x\,u_{xxx} + 4\,u\,u_x\,u_{xxxx}\,u_{xxxxx} + 12\,u_{xxxxxx}\,u\,u_{xx}^2 - \\
&\quad 40\,u\,u_{xx}\,u_{xxx}\,u_{xxxxx} + 60\,u\,u_{xx}\,u_{xxxx}^2 + 40\,u\,u_{xxx}^2\,u_{xxxx} + \\
&\quad 320\,u_x^5\,u_{xxx} + 240\,u_x^4\,u_{xx}^2 + 96\,u_x^2\,u_{xxx}\,u_{xxxxx} - 50\,u_x^2\,u_{xxxx}^2 + \\
&\quad 72\,u_x\,u_{xx}^2\,u_{xxxxx} + 40\,u_x\,u_{xx}\,u_{xxx}\,u_{xxxx} + 160\,u_x\,u_{xxx}^3 + \\
&\quad 180\,u_{xx}^3\,u_{xxxx} - 80\,u_{xx}^2\,u_{xxx}^2 + 1400\,u^6\,u_x^4.
\end{aligned}
$$

These are exactly those cases, which have already been discussed in (i) and (ii) of Lemma 3.39.

Next we compute $D^{-1}(G_0K_4)$ and $D^{-1}(G_4K_0)$: Formula (3.5.4) in Theorem 3.42 (ii) provides

$$
\begin{aligned}
D^{-1}(G_0K_4) &= F_\Theta(G_0, G_3) + F_\Theta(G_1, G_2) - G_1G_3 - \frac{1}{2}G_2^2 \\
&= \frac{105}{16}\, u^{12} + 105\, u^9\, u_{xx} + 315\, u^8\, u_x^2 + 63\, u^7\, u_{xxxx} + 630\, u^6\, u_x\, u_{xxx} + \\
&\quad 630\, u^6\, u_{xx}^2 + 2436\, u^5\, u_x^2\, u_{xx} + 18\, u_{xxxxxx}\, u^5 + 651\, u^4\, u_x^4 + \\
&\quad 180\, u_{xxxxx}\, u^4\, u_x + 534\, u^4\, u_{xx}\, u_{xxxx} + 300\, u^4\, u_{xxx}^2 + 18\, u_{xx}^4 + \\
&\quad 480\, u^3\, u_x^2\, u_{xxxx} + 2256\, u^3\, u_x\, u_{xx}\, u_{xxx} + 872\, u^3\, u_{xx}^3 + 72\, u_x^2\, u_{xxx}^2 + \\
&\quad 2\, u_{xxxxxxxx}\, u^3 + 240\, u^2\, u_x^3\, u_{xxx} + 960\, u^2\, u_x^2\, u_{xx}^2 - 72\, u_x\, u_{xx}^2\, u_{xxx} + \\
&\quad 12\, u_{xxxxxxx}\, u^2\, u_x + 60\, u_{xxxxxx}\, u^2\, u_{xx} + 108\, u_{xxxxx}\, u^2\, u_{xxx} + \\
&\quad 72\, u^2\, u_{xxxx}^2 - 480\, u\, u_x^4\, u_{xx} - 24\, u_{xxxxxx}\, u\, u_x^2 - 72\, u_{xxxxx}\, u\, u_x\, u_{xx} - \\
&\quad 144\, u\, u_x\, u_{xxx}\, u_{xxxx} + 72\, u\, u_{xx}^2\, u_{xxxx} + 80\, u_x^6 + 24\, u_{xxxxx}\, u_x^3.
\end{aligned}
$$

Finally, due to (iii) of Theorem 3.42, we compute

$$
\begin{aligned}
D^{-1}(G_4K_0) &= G_4G_0 + G_3G_1 + \frac{1}{2}G_2^2 - F_\Theta(G_3, G_0) - F_\Theta(G_2, G_1) \\
&= \frac{21}{16}u^{12} + 105\, u^8\, u_x^2 + 126\, u^6\, u_x\, u_{xxx} - 63\, u^6\, u_{xx}^2 + \\
&\quad 756\, u^5\, u_x^2\, u_{xx} + 651\, u^4\, u_x^4 + 36\, u_{xxxxx}\, u^4\, u_x - 36\, u^4\, u_{xx}\, u_{xxxx} + \\
&\quad 18\, u^4\, u_{xxx}^2 + 288\, u^3\, u_x^2\, u_{xxxx} + 564\, u^3\, u_x\, u_{xx}\, u_{xxx} - \\
&\quad 188\, u^3\, u_{xx}^3 + 672\, u^2\, u_x^3\, u_{xxx} + 966\, u^2\, u_x^2\, u_{xx}^2 + \\
&\quad 4\, u_{xxxxxxx}\, u^2\, u_x - 4\, u_{xxxxxx}\, u^2\, u_{xx} + 4\, u_{xxxxx}\, u^2\, u_{xxx} - \\
&\quad 2\, u^2\, u_{xxxx}^2 + 480\, u\, u_x^4\, u_{xx} + 24\, u_{xxxxxx}\, u\, u_x^2 + \\
&\quad 72\, u_{xxxxx}\, u\, u_x\, u_{xx} + 144\, u\, u_x\, u_{xxx}\, u_{xxxx} - 72\, u\, u_{xx}^2\, u_{xxxx} - \\
&\quad 80\, u_x^6 - 24\, u_{xxxxx}\, u_x^3 - 72\, u_x^2\, u_{xxx}^2 + 72\, u_x\, u_{xx}^2\, u_{xxx} - \\
&\quad 18\, u_{xx}^4.
\end{aligned}
$$

Although the computations involve rather large expressions, the time needed to compute the above results is rather short. We did a rough implementation of the recursion formulas presented in this section in MuPAD, which uses Algorithm 3.36 from page 164 to compute the canonical form of Θ first and, afterwards, proceeds due to Theorem 3.42. On a usual PC[5], each of the above

[5] We performed our computations using MuPAD Pro 4 under Microsoft Windows XP on a PC with a Pentium 4 CPU, 2.40 GHz, 1.0 GB RAM.

expressions is computed in less than 1 second.

The above results may also be obtained using Algorithm 3.15 on page 143 to compute the canonical form of the expressions $D^{-1}(G_2K_2)$, $D^{-1}(G_2K_3)$, $D^{-1}(G_4K_0)$ and $D^{-1}(G_0K_4)$. The results coincide, but in the special situation of the skew symmetric hierarchy under consideration, the recursion formulas of Theorem 3.42 are more efficient. $\diamond$

3.6 Conclusions

3.6.1 Resumé

In this chapter we discussed classes of ODEs arising from the application of skew symmetric operators in the sense of Theorem 3.19. In contrast to the results for computing integrating factors discussed in the framework of Chapter 2, we could present explicit symbolic recursion formulas for the computation of the integrals of the total derivatives arising from the multiplication of an ODE by a suitable integrating factor. These recursion formulas have been presented in the framework of Theorem 3.42. The central point for the efficient computation of integrating factors in such a way was the fact that we could make use of the fundamental form of the corresponding skew symmetric operator to avoid integrations.

Hence, once the fundamental form is computed using Algorithm 3.36, there are no further integrations necessary to compute integrating factors of the ODEs in the skew symmetric hierarchy generated by the corresponding skew symmetric operator. All other integrations to be performed in the setting of ODEs and integrating factors in this chapter could be done basically by using Algorithm 3.15 to compute canonical forms of the elements in the algebra $\mathcal{A}$. Hence, the requirement of a powerful integrator in the computer algebra system, where the methods of this chapter are to implemented, is not as necessary as in the case of the methods discussed in Chapter 2.

Even in the situation, when there is no elaborate algorithm for computing integrals available at all, the methods of this chapter should still work fine. Additionally, even if an elaborate integrator is available in practice, it may still be desirable to avoid its use for computing integrating factors and integrals of total derivatives as far as possible for reasons of efficiency.

What however is still missing is a link between these methods and the approach via symmetry methods. This will be discussed — very briefly — in the next two subsections.

3.6.2 Back to the symmetries

One might ask, if — in case there are not enough integrating factors found in a skew symmetric hierarchy to achieve a suitable reduction of the ODE under consideration — there are other ways for finding integrals with the help of symmetries.

In fact there are two ways, where either symmetries may help to construct additional integrating factors or where a direct integration is possible via a variant of Theorem 1.25 and Theorem 1.26. In both cases, additional research will be needed in order to make clear all structural aspects, to give rigorous results and to explore the wide area of possible applications.

However, the first method, we shall briefly present here, and the second method in the next subsection[6].

The brevity of presentation is chosen, because — to our belief — the final form of presentation of these results has not yet been found due to the necessary additional investigations.

In the algebra $\mathcal{A}$ we introduce the basics of a tensor structure, where the densities $\mathcal{A}/\mathcal{N}$ do play the role of the scalars. However, we only introduce those notions, which are needed to present those aspects we mentioned above.

For $K \in \mathcal{A}$ we define a map

$$\nabla_K : \mathcal{A} \to \mathcal{A}$$

by

$$\nabla_K Q := Q'[K],$$

where Q' denotes the directional derivative as introduced earlier in (1.3.2).

[6]These preliminary results are appended here, because we already use them in some algorithmic implementations and we want to give the reader an idea on how the results of this thesis will be extended by future research.

Remark 3.44. For any $K \in \mathcal{A}$ the operator ∇_K maps $\mathcal{N}$ into $\mathcal{N}$, i.e. if we denote the restriction of ∇_K to $\mathcal{N}$ by $\nabla_{K|\mathcal{N}}$, then we have an operator

$$\nabla_{K|\mathcal{N}} : \mathcal{N} \to \mathcal{N}.$$

Proof. If $Q \in \mathcal{N}$, then $Q = D(P)$ for some $P \in \mathcal{A}$. Hence, $\nabla_K Q = D(\nabla_K P)$ and $\nabla_K Q \in \mathcal{N}$. □

Hence, for $A_1, A_2 \in \mathcal{A}$ with $[A_1]_{\mathcal{N}} = [A_2]_{\mathcal{N}}$, it follows $[\nabla_K A_1]_{\mathcal{N}} = [\nabla_K A_2]_{\mathcal{N}}$. Therefore, the operator ∇_K may be considered as an operator mapping densities from $\mathcal{A}/\mathcal{N}$ into densities:

$$\nabla_K : \mathcal{A}/\mathcal{N} \to \mathcal{A}/\mathcal{N}.$$

Now we introduce two derivatives on $\mathcal{A}$, one being the analogue of the Lie derivative and the other one being the analogue of the exterior derivative. The map

$$L_K : \mathcal{A} \to \mathcal{A}$$

defined by

$$L_K Q := Q'[K] - K'[Q],$$

we call the *Lie derivative* and the map

$$L_K^\star : \mathcal{A} \to \mathcal{A}$$

defined by

$$L_K^\star Q := Q'[K] + K'^\star[Q],$$

where $K'^\star$ is the transposed of K' (with respect to the density valued scalar product), is said to be the *adjoint Lie derivative*.

Now, consider an equivalence class $P = [Q]_{\mathcal{N}}$ and let Q_1, Q_2 be two elements from this class. Then obviously for $K \in \mathcal{A}$ we have

$$\langle 1, \nabla_K Q_1 \rangle = \langle 1, \nabla_K Q_2 \rangle,$$

hence,

$$\langle 1, Q_1'[K] \rangle = \langle 1, Q_2'[K] \rangle$$

or

$$\langle Q_1'^\star(1), K \rangle = \langle 1, Q_2'^\star(1), K \rangle.$$

Since this is valid for all $K \in \mathcal{A}$ and since the scalar product is non–degenerate, we must have $Q_1'^\star(1) = Q_2'^\star(1)$. We call this element the *gradient* of the class P and denote it by $\mathrm{grad}(P)$. Hence, we have found

$$\nabla_K P = \langle \mathrm{grad}(P), K \rangle.$$

We summarize:

Remark 3.45. For each density $P \in \mathcal{A}/\mathcal{N}$ there is a unique element $\mathrm{grad}(P) \in \mathcal{A}$, such that $\nabla_K P = \langle \mathrm{grad}(P), K \rangle$ for all $K \in \mathcal{A}$. This gradient is equal to $Q'^\star(1)$, where any element Q from the class P may be taken.

As an obvious consequence we obtain:

Lemma 3.46. *Let $P \in \mathcal{A}/\mathcal{N}$ be some density. Then for $K \in \mathcal{A}\backslash\mathcal{F}$ the following are equivalent:*

(i) $\nabla_K P = 0$.

(ii) $\mathrm{grad}(P)$ *is an integrating factor for $K = 0$.*

Proof. The proof follows directly from the definition of an integrating factor: $\mathrm{grad}(P)$ is an integrating factor for $K = 0$ if and only if $\nabla_K P = \langle \mathrm{grad}(P), K \rangle = 0$. □

Another consequence of our definitions is:

Lemma 3.47. *For $G, K, Q \in \mathcal{A}$ we have the following product rule:*

$$\nabla_Q \langle G, K \rangle = \langle L_Q^\star(G), K \rangle + \langle G, L_Q(K) \rangle.$$

Proof. The proof is carried out by inserting the quantities as defined:

$$\begin{aligned}
\nabla_Q \langle G, K \rangle &= \nabla_Q [GK]_\mathcal{N} \\
&= \langle \mathrm{grad}(GK), Q \rangle \\
&= \langle (GK)'^\star(1), Q \rangle \\
&= \langle 1, (GK)'[Q] \rangle \\
&= \langle 1, G'[Q]K \rangle + \langle 1, GK'[Q] \rangle \\
&= \langle G'[Q], K \rangle + \langle 1, G(K'[Q] - Q'[K]) \rangle + \langle G, Q'[K] \rangle \\
&= \langle (G'[Q] + Q'^\star(G)), K \rangle + \langle G, L_Q K \rangle \\
&= \langle L_Q^\star G, K \rangle + \langle G, L_Q K \rangle.
\end{aligned}$$

□

We call some $S \in \mathcal{A}$ a *genuine Lie-Bäcklund symmetry generator* for $K = 0$, if $L_S K = 0$. One should observe that in case of an evolution equation this is the same as being a generator of a Lie–Bäcklund symmetry, that, however, in the general case this is a stronger condition as being such a generator.

Now we have the following permanence principle for integrating factors:

Theorem 3.48. (Permanence principle for integrating factors) *If G is an integrating factor for $K = 0$ and S is a genuine Lie–Bäcklund symmetry generator for $K = 0$, then $L_S^\star G$ also is an integrating factor for $K = 0$.*

Proof. We have $\langle G, K\rangle = 0$, since G is an integrating factor. Therefore

$$\nabla_S \langle G, K\rangle = 0.$$

Hence, due to Lemma 3.47:

$$0 = \nabla_S \langle G, K\rangle = \langle L_S^\star G, K\rangle + \langle G, L_S K\rangle,$$

which implies

$$\langle L_S^\star G, K\rangle = 0,$$

since S is a genuine Lie-Bäcklund symmetry generator. □

3.6.3 Integrating factors and symmetry generators: the mixed case

Since we have seen that there may be skew symmetric hierarchies, where the number of integrating factors does not suffice to integrate the equation (premature recurrence), one might ask, if in this case additional invariants, such as symmetry generators, may help to integrate. This, theoretically, is a simple problem, as we have the following theorem, which is a generalization of the results stated within Theorem 1.25 and Theorem 1.26:

Theorem 3.49. (B. Fuchssteiner, 2006) *If for an n-dimensional manifold a flow, $n - k - 1$ commuting symmetry generators and k conserved quantities are given, such that*

(i) the k conserved quantities are invariant under the symmetry generators,

(ii) the gradients of the k conserved quantities are linear independent,

(iii) at each point of the manifold the vector fields given by flow and symmetry generators are linear independent,

then the flow is integrable.

Theoretically, a proof is not really necessary, since for a given initial condition we may determine the values of the conserved quantity for this initial condition, then take the submanifold of phase space having the same values for these quantities. This submanifold is invariant under the given symmetries and there the situation of Theorem 1.25 and Theorem 1.26, respectively, applies. However, this application needs a suitable parametrization of the submanifold, which according to mathematical principles is easily determined, however the formulas coming from that may look horribly complicated. To facilitate formulas here, an intuitively simple and transparent domain for working with implicit functions is needed.

3.6.4 Open problems and perspectives

1. A question still unanswered is: Given an arbitrary n-th order ODE, $n \in \mathbb{N}$, how can one decide, whether the ODE belongs to some skew symmetric hierarchy in the sense of Theorem 3.19. If this question can be answered, the next problem arising, is the fact that a suitable skew symmetric operator has to be determined, such that the given ODE is a member of the skew symmetric hierarchy generated by this operator.

   For recognizing, if an ODE is obtained by the application of a skew symmetric operator, pattern matching methods should be developed, i.e. methods, which allow to deduce the desired information from the algebraic structure of the given ODE. As in the case of the algorithms introduced in Chapter 2, it would be desirable that these methods work without solving any auxiliary ODEs or even partial differential equations.

2. The cases, when premature recurrence phenomena as indicated by Example 3.25 appear, have to be characterized. These are exactly those cases, when the integrating factors G_i in the skew symmetric hierarchy do not suffice to reduce the ODE under consideration to a first order ODE. In the situation of Example 3.25, the conserved quantities computed provided powers of the general second order homogeneous linear ODE

$$u\varphi + u_{xx} = 0,$$

which is not integrable in the sense that there is no closed solution formula for a generic function φ.

There are also other choices for Θ, the skew symmetric operator for generating the skew symmetric hierarchy under consideration, which provide recurrence phenomena. In Example 3.26 we stated that the ODE $K_3 = 0$ generated by

$$\Theta = D^3 + 2Du + 2uD$$

in the sense of the skew symmetric hierarchy introduced in Theorem 3.19 cannot be reduced to a first order ODE. The reason seems to be that it admits symmetries, which also appear as members of the skew symmetric hierarchy.

B. Fuchssteiner[7] has made the following conjecture about this phenomenon: Premature recurrence occurs only in either of the following cases:

(i) K_0 cannot be integrated, or

(ii) if K_0 can be integrated then there are as many additional symmetries for the equation under consideration, such that an integration is possible.

The arguments for this conjecture are claimed to be simple, but they need a completely different algebraic setup in order to be proved rigorously, an algebraic setup, which goes beyond the scope of this work.

3. Besides the characterization of recurrence phenomena, further and more general skew symmetric operators (e.g. operators involving the Hilbert transform, which is structurally similar to the differential operator; see e.g. A. S. Fokas et. al. [22]) should be considered to construct more general classes of skew symmetric hierarchies in the sense of the results presented in Theorem 3.19. We have examples for such hierarchies, however, they do not fit nicely into the theory developed so far.

4. With respect to subsection 3.6.2, when looking at examples, one discovers that application of these methods lead to essentially three cases:

[7] Private communication.

(i) New integrating factors are constructed by the application of the adjoint Lie derivative.

(ii) The application of the adjoint Lie derivative leads to trivial integrating factor (factors equal to zero).

(iii) The application of the adjoint Lie derivative reproduces the given hierarchy of integrating factors.

Cases (ii) and (iii) seem to occur only in situations, where much additional algebraic structure is available. Therefore a complete characterization of these cases seems desirable. Furthermore, these cases need to be connected to the symmetry approach known for integrable partial differential equations (soliton equations).

Chapter 4

Non–local symmetries: A link to differential Galois theory

In Definition 1.8, we introduced various notions of groups leading to different types of symmetries. In most cases, the generators for these notions or their counterparts were already used: Lie point symmetries played a role in Section 1.2, where Cheb-Terrab's methods, in order to generalize these, were described. Lie–Bäcklund symmetries where used, when Lie point symmetries were transformed in Section 1.4 in order to present a unified view including the special notion of Lie point symmetries and more generalized ones. Autonomous groups and time–independent groups arose when, in order to simplify the viewpoints. A new component was introduced to have symmetry generators independent of the independent variable and in order to formally prevent a change of the independent variable by the corresponding transformations. All these notions turned out to be connected to another ingredient for finding solutions of ODEs: conserved quantities, especially those given by integrating factors. So these and their connection to symmetries had to be introduced in this thesis aiming at a unified theory for symbolically solving ODEs.

However, it remains to show that the introduction of the notion of local groups was reasonable: so, are there non–local groups for ODEs? I.e. we are looking for transformation groups (group parameter t) mapping a function $u(x)$ onto functions $u(x,t)$, where $u(x,t)$ not only depends on u and its derivatives at $t=0$, but on what is given as boundary value for u at $t=0$. Such a situation typically arises when partial differential equations are solved. So, are there partial differential equations, which serve as symmetries for ODEs? And if so, do such symmetries make sense?

There is another viewpoint, which suggests that we may look out for a generalized notion of symmetry: why did we not present any symmetry approaches for homogeneous linear differential equations

$$Lu = 0,$$

where L is a linear differential operator? This for the simple reason that, apart from the generator u of homogeneity, any Lie–Bäcklund symmetry generator for such an equation automatically already is a solution. So, looking for help in symmetry generators for finding solutions does not seem to be very promising.

And indeed, in general, symmetry methods do not seem to be of much importance for the development of algorithms for solving homogeneous linear differential equations. This hypothesis is supported by the fact that Cheb-Terrab et. al. do not seem to have been successful to carry their methods from general non–linear ODEs to homogeneous linear ODEs[1]. And in their paper [52] W. R. Oudshoorn and M. Van der Put even prove that there is no obvious connection between the Lie symmetries of a homogeneous linear ODE and its differential Galois group[2].

Because of this situation, differential Galois theory, which has been developed for finding solution formulas for such equations, seems to lead a life apart from symmetry methods, a disturbing fact if one searches for unified theories for symbolically solving ODEs. So, here the additional question arises, whether or not these two areas need to present completely disparate approaches to the same subject.

In this chapter we show — by example — that this need not to be so and that a possible link between these two approaches is given by considering non–local symmetries.

We are far away from presenting a complete theory, this for the simple reason that the promising research in this area is just starting — and, frankly, we

[1] At least, we did not find any hints in the literature or the documentation of the computer algebra system Maple, in which Cheb-Terrab implemented his algorithms, that homogeneous linear ODEs are treated by Lie point symmetry methods.

[2] More precisely, the authors of [52] prove that the structure and the dimension of the Lie algebra of Lie point symmetries of a linear differential equation in general does not help to identify its differential Galois group. So there seems to be no obvious algebraic connection between these objects.

do not know to what new horizons it will lead. However, focussing on such research seems to be promising. Therefore we can only present some aspects in order to show that new perspectives may be opening. Since, when going in even the most elementary details, we need to touch rather different subjects like differential Galois theory and nilpotent flows. We shall present for these areas short overviews in order to facilitate orientation for the reader and to make it easier to follow the course of this chapter without first studying standard textbooks on these areas. Before we present these overviews, we first introduce that kind of non–local symmetry for second order homogeneous linear ODEs, which is of interest for us.

4.1 Non–local symmetries for linear ODEs of second order

We now consider second order homogeneous linear differential equations, i.e. equations of the form

$$y''(x) + a(x)\,y'(x) + b(x)\,y(x) = 0, \tag{4.1.1}$$

where $a(x)$ and $b(x)$ are arbitrary functions. It is well–known that the transformation

$$y(x) \to z(x)\,\exp\Big(-\frac{1}{2}\int^x a(\tau)\,d\tau\Big) \tag{4.1.2}$$

transforms (4.1.1) to the more special form

$$z''(x) = r(x)\,z(x). \tag{4.1.3}$$

See for example [20], page 22. Since solutions of (4.1.1) can easily be recovered from solutions of the corresponding equation (4.1.3), we restrict our attention to equations of the more special form (4.1.3).

For these equations we explicitly introduce a non–local symmetry by relying on the well–known method of variation of constants. Later on, when nilpotent flows are introduced, we shall see that this can be approached more systematically. As in Chapter 3, we use the abbreviation D^{-1} to denote integration with respect to x in the following.

Let a solution $w(x)$ of (4.1.3) be given. Then by using variation of constants for w, a second solution w_1 is found to be

$$w_1 = wD^{-1}\Big(\frac{1}{w^2}\Big). \tag{4.1.4}$$

Now consider an arbitrary solution s_0 of (4.1.3) and the transformation group

$$s_0 \to s = s_0 + t\, s_0\, D^{-1}\Big(\frac{1}{s_0^2}\Big). \tag{4.1.5}$$

Then, by (4.1.4), it becomes clear that this transformation group maps solutions onto solutions. Hence, its infinitesimal generator

$$\frac{d}{dt}s = sD^{-1}\Big(\frac{1}{s^2}\Big) \tag{4.1.6}$$

is a symmetry group generator. Moving the s from the right–hand–side to the left–hand–side of (4.1.6) and performing a differentiation with respect to x, we obtain the partial differential equation

$$s\left(\frac{d}{dt}\frac{d}{dx}s\right) - \left(\frac{d}{dt}s\right)\left(\frac{d}{dx}s\right) = 1, \tag{4.1.7}$$

which obviously defines a non–local symmetry for (4.1.3), where (4.1.6) is its evolutionary form. The fact that this is a non–local group can formally be seen either from its form (4.1.6), because an integral operator is involved, and from (4.1.7), because it is described by a partial differential equation (whereas local symmetry groups for ODEs always were flows in finite dimensional phase space.)

However, there is an important structural point to be observed: if an explicit form $S(t,s)$ of a symmetry group is given (as in (4.1.5)), then, since the underlying equation is linear, *all* t-derivatives of $S(t,s)$ must be solutions of this equation. So, the group $S(t,s)$ defines a flow in a two dimensional space (solution space of the equation (4.1.3)). Hence, the space of functions obtained by taking subsequent t-derivatives of $S(t,s)$ has dimension two, i.e. the third derivative of these derivatives may be written as linear combination of the first two derivatives or may even be equal to zero. Flows with this property are what is introduced as nilpotent or recursive flows in the papers [27], [28] and [29] by B. Fuchssteiner and M. Lo Schiavo. So, before going to applications of these observations, we present our sketched overviews as announced above.

4.2 Overview: Nilpotent and recursive flows

We present a summarized version of the theory of nilpotent and recursive flows developed by B. Fuchssteiner and M. Lo Schiavo. The notation used in this section is chosen in the spirit of [27].

We consider a manifold given by some vector space V. The typical element of V will be denoted by v.

Definition 4.1. A vector field $G(v)$ on V is called *nilpotent (of index N)*, $N \in \mathbb{N}$, if, whenever the equation $\frac{d}{dt}s = G(s)$ defines a flow $(s,t) \mapsto s(t)$, then

$$\frac{d^{N+1}}{dt^{N+1}}s = 0 \quad \text{and} \quad \frac{d^k}{dt^k}s \neq 0$$

for all $0 \leq k \leq N$.

Definition 4.2. A vector field $G(v)$ on V is called *recursive (of order N)* if, whenever $\frac{d}{dt}s = G(s)$ defines a flow $(s,t) \mapsto s(t)$, then there are constants $\alpha_0, \ldots, \alpha_N$, such that (identically)

$$\frac{d^{N+1}}{dt^{N+1}}s = \sum_{n=0}^{N} \alpha_n \frac{d^n}{dt^n}s$$

and such a relation does not hold for any $\frac{d^k}{dt^k}s$, $0 \leq k \leq N$.

Definition 4.3. A family $G(v,t)$ of vector fields on E is called *weakly recursive (of order N)* if, whenever $\frac{d}{dt}s = G(s)$ defines a flow $(s,t) \mapsto s(t)$, there are smooth coefficients $\alpha_0(t), \ldots, \alpha_N(t)$, such that (identically)

$$\frac{d^{N+1}}{dt^{N+1}}s = \sum_{n=0}^{N} \alpha_n(t) \frac{d^n}{dt^n}s$$

and such a relation does not hold for any $\frac{d^k}{dt^k}s$, $0 \leq k \leq N$.

For examples, see [27], Examples 1.2, 1.3, 1.4, 1.6. Further applications are discussed in [28] and [29].

Assume that the vector space V consists of smooth functions in the independent variable x. As usual, by D we denote the differential operator with respect to x.

Definition 4.4. Let $G(s,t)$ be weakly recursive of order N admitting a flow $s = s(t,x)$. We consider coefficients $a_n = a_n(t,x)$, $0 \le n \le N$, such that the operator

$$\Phi = \Big(D^{N+1} + \sum_{n=0}^{N} a_n D^n \Big)$$

satisfies (for each t)

$$\Phi\Big(\frac{d^n}{dt^n} s\Big) = 0$$

for all $0 \le n \le N$. The linear differential operator Φ is called the *characteristic operator* for the flow $s = s(t)$.

In the following, for reasons of abbreviation, we often write $\Phi\frac{d^n}{dt^n}s$ instead of $\Phi(\frac{d^n}{dt^n}s)$.

We will mainly be interested in nilpotent flows in the framework of this chapter. A useful property of nilpotent flows can be characterized as follows: Let $\frac{d}{dt}s = G(s)$, $s = s(t,x)$, be a nilpotent flow of index N and Φ the associated characteristic operator of order $N+1$. Assume that

$$s_{|t=0},\ \Big(\frac{d}{dt}s\Big)_{|t=0},\ \Big(\frac{d^2}{dt^2}s\Big)_{|t=0},\ldots,\ \Big(\frac{d^N}{dt^N}s\Big)_{|t=0}$$

are linear independent functions contained in the kernel of Φ. Then $s(t,x)$ is an element of the kernel of Φ for each $t \in \mathbb{R}$, since the Taylor series expansion at $t = 0$ provides

$$s(t,x) = \sum_{k=0}^{N} \frac{t^k}{k!} \Big(\frac{d^k}{dt^k}s\Big)_{|t=0}$$

and the right–hand–side is a linear combination of elements in the kernel of Φ.

Whenever the Wronskian determinant constructed with the functions $\frac{d^n}{dt^n}s$, $n = 0,\ldots,N$, is non–zero, then the coefficients $a_0(t,x),\ldots,a_N(t,x)$ in Definition 4.4 uniquely exist and satisfy $N+1$ independent equations. The replacement of $\frac{d^{n+1}}{dt^{n+1}}s$ by $\frac{d^n}{dt^n}G(s,t)$ provides a representation of $a_0,\ldots,a_N$ as functions of s and t, i.e. $a_n = a_n(s,t)$, $0 \le n \le N$. This representation may also contain x-derivatives and x-integrals of s. Additionally, if $G = G(s)$ does not explicitly depend on t, then the a_n, $0 \le n \le N$, are independent of t.

The following remark presents a practical approach for the computation of the characteristic operator via Wronskian determinants.

Remark 4.5. The characteristic operator Φ may (formally) be computed as follows: Take[3]

$$\Phi_{\text{op\_}(N+1)} = \det \begin{pmatrix} s & \frac{d}{dt}s & \cdots & \frac{d^N}{dt^N}s & D^0 \\ \frac{d}{dx}s & \frac{d}{dx}\frac{d}{dt}s & \cdots & \frac{d}{dx}\frac{d^N}{dt^N}s & D^1 \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ \frac{d^{N+1}}{dx^{N+1}}s & \frac{d^{N+1}}{dx^{N+1}}\frac{d}{dt}s & \cdots & \frac{d^{N+1}}{dx^{N+1}}\frac{d^N}{dt^N}s & D^{N+1} \end{pmatrix} \tag{4.2.1}$$

and

$$\Phi_{\text{sub\_}(N)} = \det \begin{pmatrix} s & \frac{d}{dt}s & \cdots & \frac{d^N}{dt^N}s \\ \frac{d}{dx}s & \frac{d}{dx}\frac{d}{dt}s & \cdots & \frac{d}{dx}\frac{d^N}{dt^N}s \\ \vdots & \vdots & \cdots & \vdots \\ \frac{d^N}{dx^N}s & \frac{d^N}{dx^N}\frac{d}{dt}s & \cdots & \frac{d^N}{dx^N}\frac{d^N}{dt^N}s \end{pmatrix}, \tag{4.2.2}$$

where the determinants are formally computed using minor expansion (in the case of $\Phi_{\text{op\_}(N+1)}$, minor expansion with respect to the last column is assumed to be used). Then

$$\Phi = \Phi_{(N+1)} := (\Phi_{\text{sub\_}(N)})^{-1}\Phi_{\text{op\_}(N+1)} \tag{4.2.3}$$

and the coefficients a_n are given by

$$a_n(x,t) = (-1)^n(\Phi_{\text{sub\_}(N)})^{-1}\Phi_{\text{sub\_}(n)}, \tag{4.2.4}$$

where $\Phi_{\text{sub\_}(n)}$ is obtained from $\Phi_{\text{op\_}(N+1)}$ by eliminating the last column and the $(n+1)$-st row (i.e. by simply using minor expansion to compute the determinant).

Note that for any $n \in \{0, \ldots, N\}$ we indeed have

$$\begin{aligned} \Phi\frac{d^n}{dt^n}s &= (\Phi_{\text{sub\_}(N)})^{-1}\Phi_{\text{op\_}(N+1)}\frac{d^n}{dt^n}s \\ &= (\Phi_{\text{sub\_}(N)})^{-1}\det \begin{pmatrix} s & \frac{d}{dt}s & \cdots & \frac{d^N}{dt^N}s & \frac{d^n}{dt^n}s \\ \frac{d}{dx}s & \frac{d}{dx}\frac{d}{dt}s & \cdots & \frac{d}{dx}\frac{d^N}{dt^N}s & \frac{d}{dx}\frac{d^n}{dt^n}s \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ \frac{d^{N+1}}{dx^{N+1}}s & \frac{d^{N+1}}{dx^{N+1}}\frac{d}{dt}s & \cdots & \frac{d^{N+1}}{dx^{N+1}}\frac{d^N}{dt^N}s & \frac{d^{N+1}}{dx^{N+1}}\frac{d^n}{dt^n}s \end{pmatrix} \\ &= 0, \end{aligned}$$

[3]We adopt the notation from [27] for the following linear differential operators

since two columns of the matrix coincide and, hence, the determinant must vanish. Here, $\Phi_{\text{op-}(N+1)}\frac{d^n}{dt^n}s$ is evaluated by substituting the last column of $\Phi_{\text{op-}(N+1)}$ by

$$\frac{d^n}{dt^n}s,\ \frac{d}{dx}\frac{d^n}{dt^n}s, \dots,\ \frac{d^{N+1}}{dx^{N+1}}\frac{d^n}{dt^n}s.$$

The following theorem gives exactly that result from [27], which is of central importance for us:

Theorem 4.6. (B. Fuchssteiner, M. Lo Schiavo, 1993) *The following are equivalent:*

(i) $G(s,t)$ is weakly recursive of order $N-1$.

(ii) There exists a linear differential operator of order N of the form

$$\Phi = D^N + \sum_{n=0}^{N-1} a_n(s,t)D^n,$$

such that $\{\frac{d^n}{dt^n}s \mid n \in \mathbb{N}_0\}$ spans the solution space $\{\varphi \mid \Phi\varphi = 0\}$ of Φ.

If either of these conditions is fulfilled, then we obtain in addition: For Φ we find the representation

$$\Phi = w_0 w_1 \cdots w_{N-1} D\frac{1}{w_{N-1}}D\frac{1}{w_{N-2}}\cdots D\frac{1}{w_0},$$

where

$$\begin{aligned}
w_0 &= s,\\
w_1 &= D\frac{1}{w_0}\frac{d}{dt}s,\\
&\vdots\\
w_k &= D\frac{1}{w_{k-1}}D\frac{1}{w_{k-2}}\cdots D\frac{1}{w_1}D\frac{1}{w_0}\frac{d^k}{dt^k}s,\\
&\vdots\\
w_{N-1} &= D\frac{1}{w_{N-2}}D\frac{1}{w_{N-3}}\cdots D\frac{1}{w_1}D\frac{1}{w_0}\frac{d^{N-1}}{dt^{N-1}}s.
\end{aligned}$$

Proof. The implication (i) $\Rightarrow$ (ii) follows by taking the characteristic operator. For the implication (ii) $\Rightarrow$ (i) we refer to [27], proof of Theorem 3.9, pp. 37, 38. $\square$

We apply the result of Theorem 4.6 in Section 4.4.

Finally, we adopt the notion of the lowering of a linear differential operator from [27]. Lowerings of linear differential operators are the proper generalization of the method of variation of constants, thus reducing the order of a differential operator by given elements of its kernel.

Definition 4.7. Let $\Phi = \sum_{n=0}^{N} a_n D^n$ be a linear differential operator, a_n functions of x and, possibly, further parameters. For $w \in V$ we define the *lowering* $\Phi^{(w)}$ *of* Φ *with respect to* w by

$$\begin{aligned}\Phi^{(w)}\alpha(x) &:= \left[\Phi, \int^x \alpha(\xi)\, d\xi\right] \\ &:= \Phi\Big(w \int^x \alpha(\xi)\, d\xi\Big) - \int^x \alpha(\xi)\, d\xi\, \Phi w.\end{aligned}$$

Remark 4.8. In the notation of Definition 4.7 we obtain: If $\Phi w_0 = 0$ for some $w_0 \in V$, then $w_1 \in V$ is an element of the kernel of $\Phi^{(w_0)}$ if and only if $\Phi w_0 D^{-1} w_1 = 0$. For a proof see Example 3.1 of [27].

4.2.1 Nilpotent flows for second order linear ODEs

Here we show that application of the results in the last section for homogeneous linear ODEs of second order leads exactly to the same transformation groups as in Section 4.1.

Let $\Phi = D^2 + a_1 D + a_0$, where $a_i = a_i(x)$, $0 \leq i \leq 1$. Furthermore, let $w_0 \neq 0$, $w_0 = w_0(x)$ be an element of the kernel of Φ. Then the homogeneous linear differential equation $\Phi^{(w_0)} w = 0$ is of order 1. By Definition 4.7, where we introduced the lowering of a linear differential operator, we get

$$\Phi^{(w_0)} = (D^2 + a_1 D + a_0)^{(w_0)} = w_0 D + 2\frac{d}{dx} w_0 + a_1 w_0,$$

i.e. the differential equation $\Phi^{(w_0)} w = 0$ is of the form

$$w_0 \frac{d}{dx} w + \Big(2\frac{d}{dx} w_0 + a_1 w_0\Big) w = 0$$

and can be solved for w. A solution[4] w_1 is given by

$$w_1 = \frac{\exp(-D^{-1}a_1)}{w_0^2}.$$

Let $s = s(t,x)$ and consider the ansatz

$$\begin{aligned} w_0 &= s, \\ w_1 &= D\frac{1}{w_0}\frac{d}{dt}s. \end{aligned}$$

We solve the last equation for $\frac{d}{dt}s$ and express w_0 and w_1 in terms of s. This provides

$$\frac{d}{dt}s = sD^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big).$$

Whenever s is contained in the kernel of Φ, then the same holds for $\frac{d}{dt}s$ (see also Remark 4.8). This flow is indeed nilpotent of index 1, since

$$\begin{aligned} \frac{d^2}{dt^2}s &= \Big(\frac{d}{dt}s\Big)D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big) + sD^{-1}\Big(-\frac{2(\frac{d}{dt}s)\exp(-D^{-1}a_1)}{s^3}\Big) \\ &= s\Big(D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big)\Big)^2 - \\ &\quad 2sD^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big)\Big), \end{aligned}$$

and integration by parts provides

$$2\,D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big)\Big) = \Big(D^{-1}\Big(\frac{\exp(-D^{-1}a_1)}{s^2}\Big)\Big)^2,$$

which means $\frac{d^2}{dt^2}s = 0$.

Since any second order homogeneous linear differential equation $\frac{d^2}{dx^2}w + a_1\frac{d}{dx}w + a_0w = 0$ can be transformed to a second order homogeneous linear differential equation of the form $\frac{d^2}{dx^2}w + rw = 0$ for some $r = r(x)$ (see also (4.1.2)), we may assume without loss of generality that $a_1 = 0$, i.e. the above equation for s is reduced to

$$\frac{d}{dt}s = sD^{-1}\Big(\frac{1}{s^2}\Big),$$

[4] This solution is obtained by simply using the method of variation of constants. For details on the method of variation of constants we refer to Chapter IV, Section 7 of [42], Chapter 2 of [56] or [27]. In [27] a formulation of variation of constants in terms of lowerings of linear differential operators is given.

and, hence, independent of any of the coefficients of the original ODE we started with[5]. This is exactly the same equation we arrived at in Section 4.1.

Let us emphasize again, how the symmetry notion has been changed in this example:

Remark 4.9. The classical view on symmetries of an evolution equation

$$\frac{d}{dx}v = K(v),$$

$K(v)$ a vector field, $v = v(x)$, is that symmetry groups are point transformations mapping solutions to solutions. In the case of n-dimensional phase space, the underlying manifold M is n-dimensional, in which the orbits of $\frac{d}{dx}v = K(v)$ are curves, the curves of solutions. That is, for each fixed $x_0 \in \mathbb{R}$, $v(x_0)$ is a point on M. Now consider the ∞-dimensional manifold $\mathcal{M}$ of all possible differentiable curves in M (not only those arising as solutions of $\frac{d}{dx}v = K(v)$) and denote the manifold points by $s(x)$, where x is the parameter for the curve in M. One should observe that M is finite dimensional, whereas $\mathcal{M}$ is ∞-dimensional. The solutions of $\frac{d}{dx}v = K(v)$ form a submanifold $\mathcal{S}$ of $\mathcal{M}$. One parameter groups on $\mathcal{M}$ now have to be parameterized by a different variable than x, say t. And generators for such groups now correspond to partial differential equations

$$G\Big(s, \frac{d}{dt}s, \frac{d}{dx}s, \frac{d}{dt}\frac{d}{dx}s, \frac{d^2}{dx^2}s, \dots\Big) = 0,$$

where $s = s(t, x)$ and G is some function in its arguments. A one–parameter group of point transformations on $\mathcal{M}$ is now a *non–local symmetry group* for $\frac{d}{dx}v = K(v)$, if it leaves invariant the submanifold $\mathcal{S}$.

Starting with a homogeneous linear ODE, the manifold M is a finite dimensional space (in principle nothing more but the phase space for the ODE) and the manifold $\mathcal{M}$ is the manifold of all possible orbits on M, i.e. a manifold of functions, and $\mathcal{S}$ (the set of solutions of that linear ODE) is the kernel of the linear differential operator corresponding to the homogeneous linear ODE under consideration. I.e. $\mathcal{S}$ is a finite dimensional submanifold of an infinite dimensional manifold. For second order homogeneous linear ODEs of the form considered above, a partial differential equation corresponding to a non–local

[5] Actually, $\exp(-D^{-1}a_1)$ contributes some constant of integration c for $a_1 = 0$. For our purposes it is sufficient to consider the case $c = 1$ in the spirit of [27].

symmetry group generator is just given by the above computed nilpotent flow, which also can be written in the form

$$s\left(\frac{d}{dt}\frac{d}{dx}s\right) - \left(\frac{d}{dt}s\right)\left(\frac{d}{dx}s\right) = 1.$$

4.3 Overview: Basics in differential Galois theory

We give a short overview on some aspects of differential Galois theory and concentrate on one of the important notions therein: symmetric powers of linear differential operators. To be able to summarize some of the useful properties of symmetric powers, we first have to state some elementary notions from differential Galois theory and state the relevant classes of solutions of linear ODEs. We mainly follow [48] and [53]. The reader, who is familiar with basic notions from differential Galois theory (i.e. familiar with the notion of differential Galois groups and classes of solutions of linear ODEs like algebraic, exponential and Liouvillian solutions as well as with the notion of symmetric powers), may directly switch to Section 4.4.

Let F be a field and D_F a *derivation*, i.e. $D_F : F \to F$ is an additive map, such that $D_F(ab) = D_F(a)b + aD_F(b)$ for all elements a, b of the field F. Then the pair (F, D_F) is called a *differential field*. We simply call a field F a differential field, if the derivation on F is clear. In this case we may also write D for reasons of abbreviation instead of D_F, if it is clear, which derivation is meant.

If (F, D_F) and (G, D_G) are differential fields and $f : F \to G$ is a field homomorphism commuting with the derivations, i.e. $D_G(f(r)) = f(D_F(r))$ for all $r \in F$, then f is called a *differential homomorphism*. An invertible differential homomorphism $f : F \to F$ is called a *differential automorphism*.

A differential field E with derivation $D_E : E \to E$ is called a *differential field extension of* F, if $E \supseteq F$ and the restriction of D_E to F equals D_F.

For $a_i \in F$, $0 \leq i \leq n-1$, $n \in \mathbb{N}$, $n \geq 2$, we denote by L the linear differential operator

$$L = D^n + a_{n-1}D^{n-1} + \ldots + a_1 D + a_0, \tag{4.3.1}$$

i.e. $Ly = y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_1 y^{(1)} + a_0 y$, where $y^{(i)} = D^i y$ denotes the i-th derivative of y. Given a linear differential operator L with coefficients in the differential field F, there is always a differential field extension $E \supseteq F$, such that $L = 0$ has n independent solutions in E. Then E is called a *differential field extension of F for L*. For a proof of this statement, see [48], page ix of the Outline of Approach. The field F plays the role of a "splitting field" for the linear ODE: as in classic Galois theory, where one studies the set of solutions of univariate polynomial equations, there is always a field extension containing the roots of the equation under consideration.

An element $a \in F$ with $D_F(a) = 0$ is called a *constant of F*. The subfield of constants of a differential field F will be denoted by C_F or simply by C, if the underlying differential field is clear.

$E \supseteq F$ is called a *Picard-Vessiot extension of F for L* if E is generated over F as a differential field by solutions of $L = 0$ in E, $C_E = C_F$ and $L = 0$ has n solutions in E linear independent over the constants.

Any two Picard-Vessiot extensions of the differential field F for L are isomorphic over F. For a proof see [48], page xi of the Outline of Approach and pages 29 and 30, proof of Theorem 3.13.

Let $E \supseteq F$ a differential field extension of F. The group (with respect to composition) of differential automorphisms $E \to E$, whose restriction to F is the identity map, is denoted by $G(E/F)$, i.e.

$$G(E/F) = \{\sigma : E \to E \mid \sigma \text{ is a differential automorphism}, \sigma(a) = a \, \forall a \in F\}.$$

If E is a Picard-Vessiot extension of F for L, then $G(E/F)$ is called the *differential Galois group of $L = 0$*. We also use the customary notation $G(L)$ instead of $G(E/F)$.

We now deal with second order homogeneous linear differential equations, i.e. equations of the form (4.1.3). Let $\eta(x)$ be a solution of (4.1.3). The following classes of solutions are of interest[6]:

- $\eta(x)$ is called *rational*, if $\eta(x) \in F$.

[6]For more details on the classes of solutions see [53], Chapter 1, and [16], Paragraph 2. The definitions of the classes of solutions are also valid for the case of third and higher order homogeneous linear ODEs.

- $\eta(x)$ is called *algebraic*, if it is the solution of a polynomial equation over F.
- $\eta(x)$ is called *primitive*, if its derivative $\eta'(x)$ is an element of F, i.e.

  $$\eta(x) = \int^x f(\tau)\,d\tau$$

  for some $f(x) \in F$.
- $\eta(x)$ is called *exponential* or *exponential of a primitive solution*, if $\frac{\eta'(x)}{\eta(x)}$ is an element of F, i.e.

  $$\eta(x) = \exp\Big(\int^x f(\tau)\,d\tau\Big)$$

  for some $f(x) \in F$.
- $\eta(x)$ is called *Liouvillian* if there is a tower of differential fields

  $$F = E_0 \subset E_1 \subset E_2 \subset \ldots \subset E_{m-1} \subset E_m = E,$$

  such that $\eta(x) \in E$ and for each $i \in \{1, \ldots, m\}$ we have $E_i = E_{i-1}(\eta_i(x))$, where $\eta_i(x)$ is either algebraic, primitive or exponential over E_{i-1} and $\eta_m(x) = \eta(x)$.

The computation of rational solutions is discussed e.g. in [5] by M. Bronstein and [59] by M. F. Singer. In [24] L. Fuchs and in [58] M. F. Singer present algorithms for the computation of algebraic solutions. Methods for finding primitive and exponential solution are presented by M. Van der Put and M. F. Singer in [53]. The computation of Liouvillian solutions is closely connected to symmetric powers as we will see below.

Note that for an algebraic solution $\eta(x)$ of (4.1.3) one can find a minimal polynomial[7], i.e. a monic univariate polynomial of minimal degree $\mu_\eta(X) = a_0(x) + a_1(x)X + a_2(x)X^2 + \ldots + a_{l-1}(x)X^{l-1} + X^l$, $a_i(x) \in F$ for $0 \le i \le l$, such that $\mu_\eta(\eta(x)) = 0$. In [20] the author discusses ways to compute such minimal polynomials[8].

[7] For details, see also [39], Chapter V.

[8] In [20], W. Fakler gives explicit formulas for the computation of such minimal polynomials according to a classification of the differential Galois groups appearing among homogeneous linear ODEs of a fixed order. See section 3.1 of [20] for details.

To give a rough and compact summary of the use of symmetric powers considered in the next section, the above stated well–known terminology from differential Galois theory will be helpful.

4.3.1 Symmetric powers

Let F be a differential field of functions in the variable x and C its field of constants. Let $D : F \to F$ denote the differential operator performing differentiation with respect to the x. Let $m \in \mathbb{N}$, $m \geq 2$, and $L = \sum_{i=0}^{m} a_i(x)D^i$ be an m-th order linear differential operator, $a_i(x) \in F$, $0 \leq i \leq m$. Let $\{w_1(x), w_2(x), \ldots, w_m(x)\}$ be a basis over C of the solution space of L.

Definition 4.10. For $n \in \mathbb{N}$ the *n-th symmetric power* $L^{\otimes n}$ of L is defined as the monic linear differential operator of smallest degree over F, such that

$$\left\{w_1(x)^{l_1} \cdot w_2(x)^{l_2} \cdot \ldots \cdot w_m(x)^{l_m} \mid l_1, l_2, \ldots, l_m \in \mathbb{N}, \sum_{i=0}^{m} l_i = n\right\}$$

is a C-basis of the solution space of $L^{\otimes n}$.

In other words: $L^{\otimes n}$ is the monic linear differential operator of smallest degree with coefficients in F, such that a C-basis for the kernel of L is given by all monomials in $w_1(x), w_2(x), \ldots, w_m(x)$ of degree n.

Example 4.11. Let $L = D^2 - u(x)$ for some $u(x) \in F$. Then the second symmetric power of L is

$$L^{\otimes 2} = D^3 - 4u(x)D - 2u'(x).$$

Assume that $w(x)$ fulfills $w''(x) - u(x)w(x) = 0$. Then

$$L^{\otimes 2}(w(x)^2) = 2w(x)w'''(x) + 6w'(x)w''(x) - 2w(x)^2u'(x) - 8u(x)w(x)w'(x).$$

Inserting $w''(x) = u(x)w(x)$ and $w'''(x) = u'(x)w(x) + u(x)w'(x)$ provides $L^{\otimes 2}(w(x)^2) = 0$ as desired.

The fifth symmetric power $L^{\otimes 5}$ of L is given by:

$$\begin{aligned} L^{\otimes 5} = & D^6 - 35u(x)D^4 - 70u'(x)D^3 + (259u(x)^2 - 63u''(x))D^2 + \\ & (518u(x)u'(x) - 28u'''(x))D + \\ & 155u(x)u''(x) + 130u''(x) - 225u(x)^3 - 5u''''(x). \end{aligned}$$

As the result for $L^{\otimes 5}$ already suggests, the computation of high symmetric powers even in the case of second order linear differential operators may lead to a large expression swell for the coefficients of the operator to be computed. ⋄

Remark 4.12. The n-th symmetric power $L^{\otimes n}$ of L, $n \in \mathbb{N}$, $n \geq 1$, is a linear differential operator over the same coefficient field as L. To see this, we take an element w of the kernel of L, i.e. w is a solution of the homogeneous linear ODE associated with L. We define $Y := w^n$ and consider the equations

$$\begin{aligned} Y &= w^n, \\ DY &= nw^{n-1}Dw, \\ D^2Y &= n(n-1)w^{n-2}(Dw)^2 + nw^{n-1}D^2w, \\ &\vdots \\ D^kY &= \sum_{\substack{k_1,\dots,k_n=0\\ k_1+\dots+k_n=k}}^{k} \binom{k}{k_1,\dots,k_n} (D^{k_1}w)\cdots(D^{k_n}w), \end{aligned}$$

where $\binom{k}{k_1,\dots,k_n}$ denotes the usual multinomial coefficient. On the right–hand–side of the equations we can always express derivatives D^kw for $k \geq n$ in terms of the derivatives D^iw, $0 \leq i \leq n-1$ by using the fact that $Lw = 0$ (i.e. we simply reduce the right–hand–sides of the equations by the m-th order homogeneous linear ODE associated with L). Hence, there must be a integer l, such that the set $\{Y, DY, D^2Y, \dots, D^lY\}$ is linear dependent over the coefficient field F of L. Assume that

$$a_0Y + a_1DY + a_2D^2Y + \dots + a_lD^lY = 0,$$

$a_k \in F$, $0 \leq k \leq l$, $a_l \neq 0$, is the first non–trivial linear dependence of $Y, DY, D^2Y, \dots$ over F. Then

$$L^{\otimes n} = D^l + \sum_{k=0}^{l-1} \frac{a_k}{a_l} D^k$$

is the n-th symmetric power of L.

For second order linear differential operators L, i.e. in the case $m = 2$, the degree of $L^{\otimes n}$ is $n+1$ (see also Example 4.11; $L^{\otimes 5}$ has order 6). For linear differential operators of order $m \geq 3$, one can prove the upper bound $\binom{m+n-1}{m-1}$ for the degree of the n-th symmetric power, which may not be reached by every n-th symmetric power of an m-th order linear differential operator, but which cannot be improved (i.e. there are m-th order linear differential operators, $m \geq 3$, such that the n-th symmetric symmetric power has degree $\binom{m+n-1}{m-1}$). For details we refer to [60].

4.3.2 Applications of symmetric powers

For second and third order linear differential operators L the symmetric powers $L^{\otimes n}$, $n \in \mathbb{N}$, as defined above are used by M. F. Singer et. al. in [60] to characterize necessary and sufficient conditions for the existence of Liouvillian solutions of the homogeneous linear ODE $L(y) = 0$. By inspecting factors of $L^{\otimes n}$, M. F. Singer et. al. present results, which allow to determine the structure of the differential Galois group of $L = 0$.

Let $L(y(x)) = y''(x) + r(x)y(x) = 0$ be a second order homogeneous linear ODE[9] over F with unimodular differential Galois group[10]. Then $L(y(x)) = 0$ has Liouvillian solutions if and only if the sixth symmetric power $L^{\otimes 6}$ of L is reducible over F, i.e. $L^{\otimes 6}$ can be written as a product[11] of linear differential operators over F. For a proof and further details we refer to [60].

Furthermore, differential Galois groups appearing among second order homogeneous linear ODEs can be classified by considering symmetric powers of the associated linear differential operator. For the second order case, Proposition 4.3 in [60] gives such a classification, where the differential Galois group can be determined explicitly with the help of properties of certain symmetric powers.

In his paper [6], M. Bronstein states that the Liouvillian solutions of second and third order homogeneous linear ODEs are closely connected with the algebraic solutions of symmetric powers of the linear differential operator associated (see also [61]). Indeed, first of all, certain differential semi–invariants[12] of

[9]Note that the special form of the homogeneous linear ODE is not a restriction of the generality, since any homogeneous linear differential equation can be transformed to such a form.

[10]The fact that the differential Galois group of L has been assumed to be *unimodular* can be viewed as a technical detail not restricting the generality of the statement. By Theorem 3.2 of [60] or Theorem 1.2 of [61] one knows that any homogeneous linear ODE can be transformed to a homogeneous linear ODE, such that the differential Galois group of this new equation is unimodular. The explicit transformation for a monic linear ODE in y is $y = z \cdot \exp(-\frac{\int a_{m-1}}{m})$, where m is the order of the linear differential operator associated with the homogeneous linear ODE and a_{m-1} the coefficient of the $(m-1)$-st power of D in the operator. In fact, if a second order homogeneous linear ODE is of the form $y''(x)+r(x)y(x) = 0$, its differential Galois group is automatically a unimodular group.

[11]An introduction to the factorization of linear differential operators can be found in the framework of chapter 4 of [53]. State of the art methods for computing factors of linear differential operators are discussed e.g. in [40], [41] and [23]. Tests for the reducibility of linear differential operators are presented by M. F. Singer in [62].

[12]For a precise definition we refer to [65], Definition 4. We do not need the details in the

homogeneous linear differential equations $L(y(x)) = 0$ with finite differential Galois group can be computed as non–trivial rational solutions of symmetric powers of L. Furthermore, the main role of the symmetric powers in the context of the work presented in [61] is in fact that the coefficients of the minimal polynomials of certain classes of solutions of a irreducible homogeneous linear ODEs can be determined as solutions of symmetric powers[13].

F. Ulmer et. al. note in their paper [65] that differential invariants of the differential Galois group are rational solutions of certain symmetric powers of the linear differential operator associated. In the case of second order homogeneous linear ODEs $L(y(x)) = y''(x) + a_1(x)y'(x) + a_0(x)$ the authors investigate solutions of the associated Riccati equation given by $Ri(u(x)) = u'(x) - a_0(x) - a_1(x)u(x) + u(x)^2 = 0$. Although Riccati equations cannot be solved in general in closed form, there are methods to compute special solutions, e.g. rational solutions, of this type of ODEs (see for example [53], Chapter 4 on Algorithmic Considerations).

Furthermore, the authors of [65] state that to compute Liouvillian solutions of $L(y(x)) = 0$, one can compute the minimal polynomial $P(u)$ of an algebraic solution of $Ri(u(x)) = 0$. One of the main results presented in this context in [65] is: Let $L(y(x)) = y''(x) + a_1(x)y(x) + a_0(x)y(x)$, $a_i(x) \in F$, F some differential field. Then all zeroes of the polynomial $P(u(x)) = u(x)^m + \sum_{i=0}^{m-1} b_i(x)u(x)^i$ with $b_i(x) \in F$ are solutions of the associated Riccati equation $Ri(u(x)) = 0$ if and only if $b_{m-1}(x)$ is the logarithmic derivative of an exponential solution (over F) of $L^{\otimes m}(y(x)) = 0$. This means that there is a bijection between the set of monic polynomials of degree m over F, whose roots are solutions of the associated Riccati equation, and exponential solutions of $L^{\otimes m}(y(x)) = 0$. For a proof of this result see [65], proof of Theorem 5.

Additionally, in [65] F. Ulmer et. al. present an algorithm, which uses rational solutions of symmetric powers of second order linear differential operators to find Liouvillian solutions of second order homogeneous linear ODEs. This algorithm is presented and discussed in detail in the framework of section 3 of [65].

---

framework of this chapter.

[13]Section 5 of [61] summarizes the algorithmic ideas to determine solutions out of minimal polynomials, which themselves are computed with the help of symmetric powers.

4.3.3 Computation of symmetric powers

In Remark 4.12 on page 200, we proved the existence of symmetric powers of linear differential operators with coefficients in some differential field. The proof is constructive and, hence, can be used as a recipe for the computation of symmetric powers (at least, if one knows an element of the kernel of the linear differential operator associated). Recall that — in the terminology of Remark 4.12 — the desired symmetric power was found with the help of the first linear dependence over the differential field under consideration among the elements $Y, DY, D^2Y, \ldots$. The test for linear dependence can be done in principle using Wronskian determinants. When the first linear dependence is found, the non–trivial linear combination of the elements $Y, DY, D^2Y, \ldots$ over the considered differential field gives a representation for the symmetric power to be computed.

In their paper [6], M. Bronstein, T. Mulders and J.-A. Weil presented algorithms for the determination of symmetric powers of linear differential operators. M. Bronstein et. al. even treat a more general situation, where the coefficients of the linear differential operator are elements of some differential ring R, which is an integral domain of characteristic 0. In this situation, a special treatment for non–monic linear differential operators becomes necessary, since the leading coefficient might not be a unit (i.e. not invertible with respect to the multiplication in R) of the ring R. Since we are only interested in the case, where the underlying coefficient domain is a differential field, only Theorem 1 of [6] is of interest for us. This theorem presents an explicit iteration formula to compute the n-th symmetric power of a linear differential operator $L = D^2 + a(x)D + b(x)$, where $a(x), b(x) \in F$, F some differential field. We skip the details here, but we note that the recursion formula by M. Bronstein et. al. provides an algorithm, which allows the direct computation of symmetric powers without having to check for the first appearance of linear dependencies as mentioned above. The number of steps needed to compute the desired symmetric power is known a priori.

For third and higher order linear differential operators the algorithmic approaches for the computation of symmetric powers presented in [6] mainly base on the idea of Remark 4.12, i.e. they base on the search for a first appearance of a linear dependency[14]. The main advantage of the methods for computing symmetric powers of operators of order higher than 2 discussed in [6] can be

[14] Of course, there are some more technical details involved in the algorithms presented in [6], which we do not mention here, since this would overstep the framework of this thesis.

seen in the fact that they are fraction free methods, i.e. work in a more general coefficient domain.

Another algorithmic approach for the computation of symmetric powers of second order linear differential operators is sketched by W. R. Oudshoorn and M. Van der Put in [52]. But this approach is very similar to the one proposed by M. Bronstein et. al. in [6].

In the following we discuss an alternative approach to compute symmetric powers of second order linear differential operators based on nilpotent and recursive flows as well as on non–local symmetries for homogeneous linear ODEs.

4.4 An alternative algorithm for computing symmetric powers in the second order case

We now describe how to compute symmetric powers of second order linear differential operators using non–local symmetries, i.e. using those facts, which we know from the theory of recursive and nilpotent flows given by these non–local symmetries. We consider only second order linear differential operators of the form $L = D^2 + u$, $u = u(x)$.

Recall that any second order linear differential operator $D^2 + a_1D + a_0$ can be transformed to an operator of the form $L = D^2 + u$. Of course, the kernels of $D^2 + a_1D + a_0$ and $L = D^2 + u$ differ, but it is easy to transform any solution of the homogeneous linear differential equation $(D^2 + u)w = 0$ to a solution of $(D^2+a_1D+a_0)w = 0$ (see also the footnotes 9 and 10 on page 201).

In the former part above, we saw that the flow in $s = s(t, x)$ with characteristic operator $L = D^2 + u$ is given by

$$\frac{d}{dt}s = sD^{-1}\Big(\frac{1}{s^2}\Big).$$

Now we make use of the flow for L to compute the m-th symmetric power $L^{\otimes m}$ of L. Therefore we consider the ansatz

$$\sigma := \sigma(t, x) := s(t, x)^m.$$

It follows:

$$\frac{d}{dt}\sigma = m\, s^{m-1}\, \frac{d}{dt}s = m\, s^{m-1}\, s\, D^{-1}\Big(\frac{1}{s^2}\Big) = m\, \sigma\, D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big),$$

i.e.

$$\frac{d}{dt}\sigma = m\,\sigma\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)$$

is a nilpotent flow (as we will see below) associated with a characteristic operator denoted by $\widehat{L}$, whose kernel contains the m-th powers of the elements of the kernel of L[15]. In the following, we show that $\widehat{L} = L^{\otimes m}$.

We claim that

$$\frac{d^k}{dt^k}\sigma = m\,(m-1)\cdots(m-k+1)\,\sigma\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^k \tag{4.4.1}$$

for $k = 1, \ldots, m$ and $\frac{d^{m+1}}{dt^{m+1}}\sigma = 0$.

We prove the assertion by induction on k: Assume that the assertion is true for some $k \in \{1, \ldots, m-1\}$. Then

$$\begin{aligned}
\frac{d^{k+1}}{dt^{k+1}}\sigma = m\cdots(m-k+1)\,\Big(\frac{d}{dt}\sigma\Big)\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^k + \\
m\cdots(m-k+1)\,\sigma\,D^{-1}\Big(\Big(\frac{d}{dt}\sigma\Big)\,\Big(-\frac{2}{m}\,\frac{1}{\sigma}\,\frac{1}{\sigma^{2/m}}\Big)\Big)\,k\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^{k-1} \\
= m\cdots(m-k+1)\,m\,\sigma\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^{k+1} + \\
m\cdots(m-k+1)\,\sigma\,\underbrace{D^{-1}\Big(-2\,\frac{1}{\sigma^{2/m}}\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)\Big)}_{=-D^{-1}(1/\sigma^{2/m})^2}\,k\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^{k-1} \\
= m\cdots(m-k+1)\,(m-k)\,\sigma\,D^{-1}\Big(\frac{1}{\sigma^{2/m}}\Big)^{k+1}.
\end{aligned}$$

Hence, it also follows that $\frac{d^{m+1}}{dt^{m+1}}\sigma = 0$.

Now we assume that $s(0,x)$ and $(\frac{d}{dt}s(t,x))_{|t=0}$ are linear independent elements of the kernel of L (in fact, $(\frac{d}{dt}s(t,x))_{|t=0}$ is already contained in the kernel of L if this is true for $s(0,x)$, because the flow associated with L is nothing more but variation of constants providing a linear independent solution to a given solution). If $s(0,x)$ is an element of the kernel of L, then $\sigma(0,x) = s(0,x)^m$ is

[15]One way to compute $\widehat{L}$ explicitly is to use the formulas presented in Remark 4.5 for computing characteristic operators due to [27]. We choose an alternative way for the computation of $\widehat{L}$ in the following.

an element of the kernel of $L^{\otimes m}$. Hence, it follows that $L^{\otimes m}(\frac{d^k}{dt^k}\sigma)_{|t=0} = 0$ for all $0 \leq k \leq m$, since

$$\begin{aligned}\Big(\frac{d^k}{dt^k}\sigma\Big)_{|t=0} &= \Big(m\cdots(m-k+2)(m-k+1)s^{m-k}\Big(sD^{-1}\Big(\frac{1}{s^2}\Big)\Big)^k\Big)_{|t=0}\\ &= m\cdots(m-k+2)(m-k+1)s(0,x)^{m-k}\Big(\Big(\frac{d}{dt}s(t,x)\Big)_{|t=0}\Big)^k.\end{aligned}$$

Since the order of $L^{\otimes m}$ is $m+1$ (see for example [60], Section 3.2.2 on symmetric powers of a differential equation), $L^{\otimes m}$ is completely determined by the $m+1$ independent solutions $(\frac{d^k}{dt^k}\sigma)_{|t=0}$, $0 \leq k \leq m$.

Using the formulas of Theorem 4.6 to construct the characteristic operator associated with a recursive flow, we can compute a representation for the characteristic operator $\widehat{L}$ in terms of $\frac{d^k}{dt^k}\sigma$, $0 \leq k \leq m$. It follows that

$$\widehat{L} = w_0 w_1 \cdots w_m D\frac{1}{w_{m-1}}D\frac{1}{w_m}\cdots D\frac{1}{w_0},$$

where

$$\begin{aligned} w_0 &= \sigma,\\ w_1 &= D\frac{1}{w_0}\frac{d}{dt}\sigma,\\ &\vdots\\ w_k &= D\frac{1}{w_{k-1}}D\frac{1}{w_{k-2}}\cdots D\frac{1}{w_1}D\frac{1}{w_0}\frac{d^k}{dt^k}\sigma,\\ &\vdots\\ w_m &= D\frac{1}{w_{m-1}}D\frac{1}{w_{m-2}}\cdots D\frac{1}{w_1}D\frac{1}{w_0}\frac{d^m}{dt^m}\sigma.\end{aligned}$$

Now we insert the representations (4.4.1) for $\frac{d^k}{dt^k}\sigma$, $0 \leq k \leq m$. Using integration by parts, we find that

$$w_0 = \sigma \quad \text{and} \quad w_k = c_k(m)\frac{1}{\sigma^{2/m}}$$

for $1 \leq k \leq m$, where the $c_k(m)$ are constants, which can be ignored, since they cancel in the below computation. Hence, it follows that

$$\begin{aligned}\widehat{L} &= \sigma\Big(c_1(m)\frac{1}{\sigma^{2/m}}\Big)\cdots\Big(c_m(m)\frac{1}{\sigma^{2/m}}\Big)D\Big(\frac{1}{c_m(m)}\sigma^{2/m}\Big)\cdots D\Big(\frac{1}{c_1(m)}\sigma^{2/m}\Big)D\frac{1}{\sigma}\\ &= \frac{1}{\sigma}\underbrace{D\sigma^{2/m}\cdots D\sigma^{2/m}}_{m\text{-times}}D\frac{1}{\sigma}\\ &= \frac{1}{s^m}\underbrace{Ds^2\cdots Ds^2}_{m\text{-times}}D\frac{1}{s^m}.\end{aligned}$$

Thus, the "expanded form" of $\widehat{L}$ is given by

$$\widehat{L} = D^{m+1} + \sum_{k=0}^{m}\varphi_k\Big(s, \frac{d}{dx}s, \dots\Big)D^k$$

for some polynomial expressions $\varphi_k(s, \frac{d}{dx}s, \dots)$ depending only on s and derivatives of s with respect to x. Since the original linear differential operator we started with was $L = D^2 + u$, we may substitute all derivatives $\frac{d^k}{dx^k}s$, $2 \le k \le m$, by $-\frac{d^{k-2}}{dx^{k-2}}(su)$ successively, until only s, $\frac{d}{dx}s$, u and derivatives of u with respect to x are contained in the representation of $\widehat{L}$. Then we can write

$$\widehat{L} = \widehat{L}_0 + \Big(\frac{\frac{d}{dx}s}{s}\Big)\widehat{L}_1 + \Big(\frac{\frac{d}{dx}s}{s}\Big)^2\widehat{L}_2 + \dots,$$

where $\widehat{L}_0, \widehat{L}_1, \widehat{L}_2, \dots$ are linear differential operators, whose coefficients only contain u and derivatives of u with respect to x and $\widehat{L}_0$ is of order $m+1$, whereas $\widehat{L}_1, \widehat{L}_2, \dots$ are of order at most m.

Note that the evolution equation $\frac{d}{dt}\sigma = m\sigma D^{-1}(\frac{1}{\sigma^{2/m}})$ constructed for σ is invariant under the automorphisms of the differential Galois group $G(L)$ of the homogeneous linear differential equation $Lw = (D^2 + u)w = 0$. For $T \in G(L)$ we obtain that, whenever $s(0, x)$ is an element of the kernel of L, then the same holds for $T(s(0, x))$, since any automorphism of $G(L)$ maps solutions of $Lw = 0$ to solutions[16]. The construction of the nilpotent flow for s and σ is independent of the fact, which elements of the kernel of L are chosen as initial values $s(0, x)$.

To see this, we consider an arbitrary $T \in G(L)$. Then we construct the above operator $\widehat{L}$ using the flow for s and in the same manner compute the

[16]See also page 197 for details on the differential Galois group.

linear differential operator $\widetilde{L}$ using the flow for $T(s)$ instead of s. Then $\widehat{L}$ and $\widetilde{L}$ are both linear differential operators of order $m+1$ and the kernels of both of them consist of all m-th powers of the elements of the kernel of L. Hence, we can conclude $\widehat{L} = \widetilde{L}$ and the application of any $T \in G(L)$ to an element s of the kernel of L does not change the nilpotent flows for s and σ.

The application of T to $\frac{\frac{d}{dx}s}{s}, (\frac{\frac{d}{dx}s}{s})^2, \ldots$ does not change the operator $\widehat{L}$. In $G(L)$ there is at least one automorphism, for which

$$T\left(\frac{\frac{d}{dx}s}{s}\right) \neq \frac{\frac{d}{dx}s}{s},$$

since otherwise

$$\frac{\frac{d}{dx}T(s)}{T(s)} = \frac{T(\frac{d}{dx}s)}{T(s)} = T\left(\frac{\frac{d}{dx}s}{s}\right) = \frac{\frac{d}{dx}s}{s}$$

provides that $T(s) = cs$ for some constant c. But there is at least one transformation (variation of constants), which provides a linear independent solution to a given solution and which is not the case for transformations with $T(\frac{\frac{d}{dx}s}{s}) = \frac{\frac{d}{dx}s}{s}$.

Now choose an element $T \in G(L)$ for which $T(\frac{\frac{d}{dx}s}{s}) \neq \frac{\frac{d}{dx}s}{s}$ holds. Then T leaves invariant the operator $\widehat{L}$ if and only if $\widehat{L}_1, \widehat{L}_2, \ldots$ are equal to zero. Thus, it follows $\widehat{L} = \widehat{L}_0$. Since the kernel of $\widehat{L}$ is spanned by all m-th powers of the elements of the kernel of L, it follows

$$\widehat{L} = \widehat{L}_0 = L^{\otimes m}$$

and we are done.

We sum up the results and close this section with the following algorithm to compute symmetric powers of second order linear differential operators:

Algorithm 4.13. Let $L = D^2 + u(x)$ and $m \in \mathbb{N}$. We compute $L^{\otimes m} f(x)$ for some arbitrary symbolic elements $f(x)$ as follows:

1. Define $s(x) := w(x)^m$ and $r(x) := \frac{d}{dx}(\frac{f(x)}{s(x)})$.
2. For i from 1 to m repeat the following steps:
   - Compute
   $$r(x) := \frac{d}{dx}(s(x)^{2/m} r(x)).$$

- Substitute $\frac{d^2}{dx^2}w(x)$ by $-u(x)w(x)$ in $r(x)$, i.e. reduce $r(x)$ by the homogeneous linear ODE $\frac{d^2}{dx^2}w(x) = -u(x)w(x)$.

3. Replace $\frac{d}{dx}w(x)$ by 0 and $w(x)$ by 1 in $r(x)$.

4. Return $r(x) = L^{\otimes m}f(x)$. $\triangle$

Algorithm 4.13 in the above form can be used to obtain the symmetric powers given in Example 4.11.

4.5 Conclusions

4.5.1 Resumé

We did a rough implementation of Algorithm 4.13 in the computer algebra system MuPAD. We compared our implementation with the algorithm to compute symmetric powers of second order linear differential operators already implemented in MuPAD, which is based on the algorithms in [6].

In practice, even the rough implementation of Algorithm 4.13 is slightly more efficient than the one already implemented in MuPAD. For example, the computation of $L^{\otimes 14}$, where $L = D^2 + u(x)$ for some arbitrary symbolic function $u(x)$, using Algorithm 4.13 takes about 4 seconds[17]. The same computation using the already implemented version of the algorithm in MuPAD to compute symmetric powers takes with about 9 seconds more than twice as long.

4.5.2 Open problems and perspectives

We applied the results from [27] for nilpotent and recursive flows to give an alternative approach to compute symmetric powers. For third and higher order homogeneous linear ODEs, the computation of symmetric powers is more difficult for at least two reasons: On the one hand, the computation of symmetric powers for higher order linear differential operators involves a large expression swell[18]. On the other hand, the degree of the symmetric powers of

[17] We performed our computations using MuPAD Pro 4 under Microsoft Windows XP on a PC with a Pentium 4 CPU, 2.40 GHz, 1.0 GB RAM.

[18] For further remarks on the problem of the expression swell involved with the computation of symmetric powers and approaches to avoid it see [6].

linear differential operators of order at least 3 is not known a priori[19].

The question arising is, whether the approach for the computation of symmetric powers of second order linear differential operators discussed in this chapter can be generalized to higher order linear differential operators. The central role in our approach plays a nilpotent flow associated with a general second order homogeneous linear ODE and which serves as a non–local symmetry for the ODE.

In fact, in a similar way as discussed in Subsection 4.2.1 on pp. 193 one can associate nilpotent flows with general higher order homogeneous linear ODEs.

The nilpotent flow for general third order linear homogeneous ODEs with unimodular differential Galois group reads[20]

$$\frac{d^2}{dt^2}s = sD^{-1}\Big(\Big(D\frac{\frac{ds}{dt}}{s}\Big)D^{-1}\frac{1}{s^3(D\frac{\frac{ds}{dt}}{s})^2}\Big).$$

The nilpotent flow for general fourth order linear homogeneous ODEs with unimodular differential Galois group reads

$$\frac{d^3}{dt^3}s = sD^{-1}\Big(\Big(D\frac{\frac{ds}{dt}}{s}\Big)D^{-1}\Big(\Big(D\frac{1}{D\frac{\frac{ds}{dt}}{s}}D\frac{\frac{d^2s}{dt^2}}{s}\Big)D^{-1}\Big(\frac{1}{s^4(D\frac{\frac{ds}{dt}}{s})^3(D\frac{1}{D\frac{\frac{ds}{dt}}{s}}D\frac{\frac{d^2s}{dt^2}}{s})^2}\Big)\Big)\Big).$$

As in the case of the nilpotent flow associated with a general second order homogeneous linear ODE, these nilpotent flows are independent of the coefficients of the corresponding linear ODEs[21].

The main problem one is faced with in the situation of these flows is the fact that the right–hand–sides involve derivatives of s with respect to t. This makes the situation more complicated and the question, whether these flows can be used to give alternative approaches for the computation of symmetric powers of

[19]The degree of certain symmetric powers of linear differential operators of order at least 3 gives information concerning the existence of Liouvillian solutions of the associated homogeneous linear ODE (see [60] for details).

[20]The following two equations become nilpotent flows in the strict sense of Definition 4.1 when a suitable phase space is introduced.

[21]We verified this also for general n-th order homogeneous linear ODEs, $n \in \mathbb{N}$, $n \geq 3$, as long as the ODEs under consideration have a unimodular differential Galois group.

linear differential operators of order at least 3 remains unanswered in this work, but gives a perspective for further research in this area.

Bibliography

[1] M. Adler, P. Van Moerbeke, P. Vanhaecke: *Algebraic Integrability, Painlevé Geometry and Lie Algebras*, Springer Verlag, Berlin · Heidelberg · New York, Ergebnisse der Mathematik und ihrer Grenzgebiete (A Series of Modern Surveys in Mathematics), Volume 47, 2004

[2] K. E. Avrachenkov, M. Haviv, P. G. Howlett: *Inversion of analytic matrix functions that are singular at the origin*, Siam J. Mmatrix Anal. Appl., Society for Industrial and Applied Mathematics, Volume 22, Number 4, pp. 1175-1189, 2001

[3] G. Baumann: *Symmetry Analysis of Differential Equations with Mathematica*, Springer Verlag, New York · Berlin · Heidelberg, 2000

[4] G. W. Bluman, S. Kumei: *Symmetries and Differential Equations*, Springer Verlag, New York · Berlin · Heidelberg · Tokyo · Hong Kong, Applied Mathematical Sciences, Volume 81, 1989

[5] M. Bronstein: *Solutions of linear differential equations in their coefficient field*, Journal of Symbolic Computation, Volume 13, pp. 413-439, 1992

[6] M. Bronstein, T. Mulders, J.-A. Weil: *On Symmetric Powers of Differential Operators*, ACM Press, Proceedings of the 1997 international symposium on Symbolic and Algebraic Computation, pp. 156-163, 1997

[7] E. S. Cheb-Terrab, L. G. S. Duarte, L. A. C. P. da Mota: *Computer Algebra Solving of First Order ODEs Using Symmetry Methods*, Computer Physics Communications, 101 (1997) 254., 1997

[8] E. S. Cheb-Terrab: *Introduction to the ODEtools package version 3.91*, Documentation website for the package ODEtools included in the computer

algebra system Maple, `http://www.scg.uwaterloo.ca/~ecterrab/help/odetools1.html`, 1997

[9] E. S. Cheb-Terrab, A. D. Roche: *Computer Algebra Solving of Second Order ODEs Using Symmetry Methods*, Computer Physics Communications, 108 (1998) 90, 1998

[10] E. S. Cheb-Terrab, A. D. Roche: *Symmetries and First Order ODE Patterns*, Computer Physics Communications, 113 (1998) 239, 1998

[11] E. S. Cheb-Terrab, A. D. Roche: *Integrating factors for second order ODEs*, Journal of Symbolic Computation, Volume 27, pp. 501-519, 1999

[12] E. S. Cheb-Terrab: *ODE trends in computer algebra: four linear and nonlinear challenges*, Proceedings of the Maple Summer Workshop — Waterloo, 2002

[13] E. S. Cheb-Terrab, T. Kolokolnikov: *First order ODEs, Symmetries and Linear Transformations*, European Journal of Applied Mathematics, Volume 14, Number 2, pp. 231-246, 2003

[14] E. S. Cheb-Terrab: *The ODEtools package*, Documentation website for the package ODEtools included in the computer algebra system Maple, `http://www.scg.uwaterloo.ca/~ecterrab/odetools.html`, 2005

[15] E. S. Cheb-Terrab: *Homepage of E. S. Cheb-Terrab*, Associate member of the Symbolic Computation Group, University of Waterloo, `http://www.scg.uwaterloo.ca/~ecterrab`, 2005

[16] R. C. Churchill: *Introduction to the Galois Theory of Linear Ordinary Differential Equations*, Hunter College and the Graduate Center of CUNY and the University of Calgary, Prepared for the Kolchin Seminar on Differential Algebra, Graduate Center, City University of New York, 2005

[17] E. A. Coddington, N. Levison: *Theory of Ordinary Differential Equations*, E. Krieger, Malabar, Florida, 1984

[18] C. Creutzig, K. Gehrs, W. Oevel: *Das MuPAD–Tutorium*, Springer Verlag, Berlin · Heidelberg · New York · Hong Kong · London · Mailand · Paris · Tokio, 2004

[19] L. Dresner: *Applications of Lie's Theory of Ordinary and Partial Differential Equations*, Institute of Physics Publishing, Bristol · Philadelphia, 1999

[20] W. Fakler: *Algebraische Algorithmen zur Lösung von linearen Differentialgleichungen*, B. G. Teubner, Stuttgart · Leipzig, MuPAD Reports, 1999

[21] A. S. Fokas: *Invariants, Lie–Bäcklund Operators and Bäcklund Transformations*, PhD thesis in Applied Mathematics, California Institute of Technology, Pasadena, California, 1979

[22] A. S. Fokas, B. Fuchssteiner: *The hierarchy of the Benjamin–Ono Equation*, Physics Letters, Volume 86A, Number 6,7, 1981

[23] A. Fredet: *Factorization of Linear Differential Operators in Exponential Extensions*, ISSAC'03, August 3-6, Philadelphia, Pennsylvania, USA, 2003

[24] L. Fuchs: *Über die linearen Differentialgleichungen zweiter Ordnung, welche algebraische Integrale besitzen*, zweite Abhandlung, Journal für Mathematik 85, 1878

[25] B. Fuchssteiner: *Application of Hereditary Symmetries to Nonlinear Evolution Equations*, Nonlinear Analysis 3, pp. 849-862, 1979

[26] B. Fuchssteiner: *The Lie Algebra Structure of Nonlinear Evolution Equations admitting Infinite Dimensional Abelian Symmetry Groups*, Progress of Theoretical Physics 65, pp. 861-876, 1981

[27] B. Fuchssteiner, M. Lo Schiavo: *Nilpotent and Recursive Flows*, Manuscripta Mathematica 79, Springer Verlag, pp. 27-48, 1993

[28] B. Fuchssteiner, M. Lo Schiavo: *Nonlinear PDEs and Recursive Flows: Theory*, Applied Mathematics Letters, 6, pp. 97-100, 1993

[29] B. Fuchssteiner, M. Lo Schiavo: *Nonlinear PDEs and Recursive Flows: Applications*, Applied Mathematics Letters, 6, pp. 101-104, 1993

[30] B. Fuchssteiner: *Symbolische Behandlung von Differentialgleichungen*, appeared in *Computeralgebra in Deutschland — Bestandsaufnahme, Möglichkeiten, Perspektiven*, published by Fachgruppe Computeralgebra Deutschland (of the GI, DMV and GAMM), Passau · Heidelberg, 1993

[31] B. Fuchssteiner, K. Gehrs: *Neue Methoden zur algorithmischen Lösung nicht–linearer ODEs über Symmetrien und integrierende Faktoren*, Computeralgebra–Tagung at the university of Kassel, Fachgruppe Computeralgebra Deutschland (of the GI, DMV and GAMM), 2005

[32] B. Fuchssteiner: *Algorithmische Behandlung von Differenzialgleichungen*, Lectures held at the University of Paderborn, 2005/2006

[33] B. Fuchssteiner, K. Gehrs, W. Oevel: *Constellation problems and integrability for linear ODEs with constant coefficients*, Preprint, 2006

[34] F. R. Gantmacher: *The Theory of Matrices*, Chelsea, New York, 1959.

[35] K. Gehrs: *Some practical algorithms for the solution of ODEs via symmetry methods and integrating factors*, talk at the ICMS workshop "Algebraic Theory of Differential Equations", Heriot-Watt University, Edinburgh, 2006

[36] Ü. Göktas, W. Hereman: *Symbolic Computation of Conserved Densities for Systems of Nonlinear Evolution Equations*, Journal of Symbolic Computation, Volume 11, pp. 1-31, 2004

[37] K. Gottheil: *Axioms, categories and domains*, mathPAD, 4(1), pp. 24-29, 1994

[38] W. Hackbusch, H. R. Schwarz, E. Zeidler: *Teubner Taschenbuch der Mathematik*, Teil 1, B. G. Teubner, Leipzig, 1996

[39] R. Hermann: *The Geometry of Non–Linear Differential Equations, Bäcklund Transformations, and Solitons, Part A*, Math Sci Press, Interdisciplinary Mathematics Volume XII, 1976

[40] M. van Hoeij: *Factorization of Linear Differential Operators*, PhD thesis, University of Nijmegen, 1996

[41] M. van Hoeij: *Factorization of Differential Operators with Rational Functions Coefficients*, Journal of Symbolic Computation, Volume 24, pp. 537-561, 1997

[42] P.-F. Hsieh, Y. Sibuya: *Basic Theory of Ordinary Differential Equations*, Springer Verlag, New York · Berlin · Heidelberg · Barcelona · Hong Kong · London · Milan · Paris · Singapore · Tokyo, Universitext, 1999

[43] N. H. Ibragimov: *Elementary Lie Group Analysis and Ordinary Differential Equations*, John Wiley & Sons, Chichester · New York · Weinheim · Brisbane · Singapore · Toronto, Mathematical Methods in Practice, 1999

[44] E. L. Ince: *Ordinary Differential Equations*, Dover Publications, New-York, 1956

[45] E. Kamke: *Differentialgleichungen*, Akademische Verlagsgesellschaft, Leipzig 1959

[46] H. W. Knobloch, F. Kappel: *Gewöhnliche Differentialgleichungen*, B. G. Teubner, Stuttgart, 1974

[47] J. J. Kovacic: *An algorithm for solving second order linear homogeneous differential equations*, Notes on a talk at the City College of The City University of New York, `http://mysite.verizon.net/jkovacic/`, 2005

[48] A. R. Magid: *Lectures on Differential Galois Theory*, American Mathematical Society, University Lecture Series, Volume 7, 1994

[49] M. Majewski: *MuPAD Pro Computing Essentials*, Springer Verlag, Berlin · Heidelberg · New York · Hong Kong · London · Milan · Paris · Tokyo, 2004

[50] H. Naundorf: *Ein denotationelles Modell für parallele objektbasierte Systeme*, B. G. Teubner, Stuttgart · Leipzig, MuPAD Reports, 1997

[51] P. J. Olver: *Applications of Lie Groups to Differential Equations*, Springer Verlag, New York · Berlin · Heidelberg · Tokyo, 1986

[52] W. R. Oudshoorn, M. Van der Put: *Lie Symmetries And Differential Galois Groups of Linear Equations*, Mathematics of Computation, Volume 71, Number 237, pp. 349-361, 2001

[53] M. Van der Put, M. F. Singer: *Galois Theory of Linear Differential Equations*, Springer Verlag, Berlin · Heidelberg · New York · Hong Kong · London · Milan · Paris · Tokyo, Grundlehren der mathematischen Wissenschaften (A series of comprehensive studies in mathematics), Volume 328, 2003

[54] J. C. Robbinson: *An Introduction to Ordinary Differential Equations*, Cambridge University Press, 2004

[55] T. Sasaki, H. Murao: *Efficient Gaussian Elimination Method for Symbolic Determinants and Linear Systems*, ACM Transactions on Mathematic Software, Volume 8, Number 3, pp. 277-289, 1982

[56] L. Schlesinger: *Handbuch der Theorie der linearen Differentialgleichungen*, Band I, Teubner, Leipzig, 1895

[57] M. Schreiber: *Differential Forms — A heuristic introduction*, Springer Verlag, New York Heidelberg Berlin, Universitext, 1977

[58] M. F. Singer: *Algebraic solutions of n^{th} order linear differential equations*, Proceedings of the 1979 Queens Conference on Number Theory, Queens Papers in Pure and Applied Mathematics 54, 1980

[59] M. F. Singer: *Liouvillian solutions of linear differential equations with Liouvillian coefficients*, Journal of Symbolic Computation, Volume 11, pp. 251-273, 1991

[60] M. F. Singer, F. Ulmer: *Galois Groups for second and third order linear differential equations*, Journal of Symbolic Computation, Volume 16, pp. 9-36, 1993

[61] M. F. Singer, F. Ulmer: *Liouvillian and algebraic solutions of second and third order linear differential equations*, Journal of Symbolic Computation, Volume 16, pp. 37-73, 1993

[62] M. F. Singer: *Testing Reducibility of Linear Differential Operators: a Group Theoretic Perspective*, Applicable Algebra in Engineering, Communication and Computing, Volume 7, pp. 77-104, 1996

[63] M. F. Singer, F. Ulmer: *Linear differential equations and products of linear forms*, Journal of Pure and Applied Algebra, 117 & 118, pp. 549-563, 1997

[64] H. Stephani: *Differential Equations — Their solutions using symmetries*, Cambridge University Press, Cambridge · New York · Port Chester · Melbourne · Sidney, 1989

[65] F. Ulmer, J.-A. Weil: *Note on Kovacic's algorithm*, Journal of Symbolic Computation, Volume 22, pp. 179-200, 1996

[66] W. Walter: *Gewöhnliche Differentiagleichungen*, Springer Verlag, New York · Heidelberg · Berlin, 1996

[67] J. Weidmann: *Lineare Operatoren in Hilberträumen*, B. G. Teubner, Stuttgart, 1976

[68] D. Zwillinger: *Handbook of Differential Equations*, 2nd Edition, Academic Press, San Diego, 1992

List of some notation

Chapter 1

$y^{(k)}(x)$	k-th derivative of $y(x)$ with respect to x
$\frac{\partial}{\partial\epsilon}_{\vert\epsilon=0}$	Derivative with respect to ϵ at $\epsilon = 0$
R	A one–parameter group of transformations
$\eta(\Gamma, x(t), y(t)), \xi(\Gamma, x(t), y(t))$	Curve functions forming the infinitesimal generator of a one–parameter group of transformations
$H(x, y, y', \ldots, y^{(n)}) = 0$	An ODE of the form $y^{(n)} - \Phi(x, y, y', \ldots, y^{(n-1)}) = 0$
$\eta^{[k]}(\Gamma, x(t), y(t))$	k-th prolongation of the curve function $\eta(\Gamma, x(t), y(t))$
X	The infinitesimal generator of a symmetry
$XH = 0 \mod H = 0$	Way of saying that the expression XH has to be zero on the submanifold given by $H = 0$
M	Some C^∞-manifold
u	The manifold variable
$K(u)$	Some vector field on the manifold M
$[\![K(u), G(u)]\!]$	Commutator of the vector fields $K(u)$ and $G(u)$.
V	A vector space
$\mathcal{L}(V)$	The set of all vector fields on the vector space V

$\widehat{\mathcal{L}}(V)$	The set of all covector fields on the vector space V
$\mathcal{F}(V)$	The set of all scalar fields on the vector space V
$\langle \Gamma, K \rangle$	Application of the covector field Γ to the vector field K
$\mathcal{T}_{(n,m)}(V)$	Set of all n-times-covariant and m-times-contraviant tensors on V
d	Exterior derivative
grad	Gradient of a scalar field

Chapter 2

$\Phi(x, y, y', \ldots, y^{(n-1)})$	Right–hand–side of an n-th order ODE of the form $y^{(n)} = \Phi(x, y, y', \ldots, y^{(n-1)})$
μ	An integrating factor
$y^{(k)}(x)$	k-th derivative of $y(x)$ with respect to x
D^{-1}	Integral of a function; formal integration with respect to x
$\xi(x, y), \eta(x, y)$	Infinitesimals of a Lie point symmetry generator

Chapter 3

D	Differential operator; differentiation with respect to x
D^{-1}	Integral of a function; formal integration with respect to x
$\mathcal{A}$	Algebra containing x, $u = u(x)$ and a fixed set of infinitely often differentiable functions being closed against the application of D to its elements
$\mathcal{F}$	Subset of $\mathcal{A}$, whose elements do not contain u or any of its formal derivatives
ord_u	Order of the highest derivative of u with respect to x of an element of $\mathcal{A}$

$\mathcal{N}$	Set of all total derivatives in $\mathcal{A}$; also referred to as the integrable part of $\mathcal{A}$
$A \sim_{\mathcal{N}} B$	Equivalence relation on $\mathcal{A}$; holds if $A - B \in \mathcal{N}$
$[A]_{\mathcal{N}}$	Equivalence class of $A \in \mathcal{A}$ with respect to $\sim_{\mathcal{N}}$
$\mathcal{A}/\mathcal{N}$	Set of all equivalence classes of elements of $\mathcal{A}$ with respect to $\mathcal{N}$
$\langle A, B\rangle$	Density valued scalar product on $\mathcal{A} \times \mathcal{A}$; the equivalence class $[AB]_{\mathcal{N}}$
Θ	Skew symmetric operator $\mathcal{A} \to \mathcal{A}$
$\mathcal{J}_0$	Elements $A = A(x, u) \in \mathcal{A}$ with the property that in their formal Taylor series expansion with respect to u the term of polynomial order 0 does vanish
$\mathcal{J}_n$, $n \in \mathbb{N}$	Elements $A = A(x, u, u_x, \ldots, u^{(n)}) \in \mathcal{A}$ with the property that in their formal Taylor series expansion with respect to $u^{(n)}$ the terms of polynomial order 0 and 1 do vanish
$\mathcal{J}$	The direct sum of all $\mathcal{J}_k$, $k \in \mathcal{N}_0$; also referred to as the non–integrable part of $\mathcal{A}$
F_Θ	Fundamental form of the skew symmetric operator Θ; a map $\mathcal{A} \times \mathcal{A} \to \mathcal{A}$
Q_Θ	Quadratic form of the skew symmetric operator Θ; a map $\mathcal{A} \to \mathcal{A}$
$A^{(n)}$	The n-th derivative of an element $A \in \mathcal{A}$ with respect to x

Chapter 4

F	A differential field
D_F	A derivation on F
L	A linear differential operator
$G(E/F)$	Differential Galois group of the Picard-Vessiot extension $E \supseteq F$; E and F are differential fields

$L^{\otimes m}$	m-th symmetric power of the linear differential operator L
$\binom{k}{k_1,\ldots,k_n}$	Multinomial coefficient Φ
	A linear differential operator
$\Phi_{(N+1)}$	The characteristic operator for a weakly recursive flow
$\Phi^{(w)}$	Lowering of the linear differential operator Φ with respect to w
det	The Determinant of a square matrix

Glossary of Algorithms

Algorithm 2.7	Computing integrating factors of the form $\mu = \mu(x, y)$ of third order ODEs, **82**
Algorithm 2.9	Computing integrating factors of the form $\mu = \mu(x, y')$ of third order ODEs, **85**
Algorithm 2.11	Computing integrating factors of the form $\mu = \mu(y, y')$ of third order ODEs, **88**
Algorithm 2.15	Computing integrating factors of the form $\mu = \mu(y'')$, $\mu(0) \neq 0$, of third order ODEs, **93**
Algorithm 2.19	Computing integrating factors of the form $\mu = f(x, y, y')(y'')^m$, $m \geq 1$, of third order ODEs, **99**
Algorithm 2.22	Computing integrating factors of the form $\mu = f(x, y, y')\frac{1}{(y'')^m}$, $m \geq 3$, of third order ODEs, **103**
Algorithm 2.26	Computing integrating factors of the form $\mu = f(x, y, y')\frac{1}{y''}$ of third order ODEs, **108**
Algorithm 2.29	Computing integrating factors of the form $\mu = \mu(y^{(i)}, y^{(n-2)})$, $0 \leq i \leq n-3$, of n-th order ODEs, **113**
Algorithm 2.30	Computing integrating factors of the form $\mu = \mu(x, y^{(n-2)})$ of n-th order ODEs, **115**
Algorithm 2.31	Computing integrating factors of the form $\mu = \mu(y^{(i)}, y^{(j)})$, $0 \leq i < j \leq n-3$, $i+1 < j$, of n-th order ODEs, **118**
Algorithm 2.32	Computing integrating factors of the form $\mu = \mu(y^{(n-1)})$ of n-th order ODEs, **119**

Algorithm 2.33	Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})(y^{(n-1)})^m, m \in \mathbb{N}$, of n-th order ODEs, **122**
Algorithm 2.34	Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{(y^{(n-1)})^m}, m \in \mathbb{N}, m \geq 3$, of n-th order ODEs, **125**
Algorithm 2.35	Computing integrating factors of the form $\mu = f(x, y, y', \ldots, y^{(n-2)})\frac{1}{y^{(n-1)}})$ of n-th order ODEs, **127**
Algorithm 3.15	Computation of the canonical form in the algebra $\mathcal{A}$, **143**
Algorithm 3.34	Computation of the quadratic form in the algebra $\mathcal{A}$, **143**
Algorithm 3.36	Computation of the fundamental form in the algebra $\mathcal{A}$, **164**
Algorithm 4.13	Computation of symmetric powers of second order linear differential operators, **208**

Index