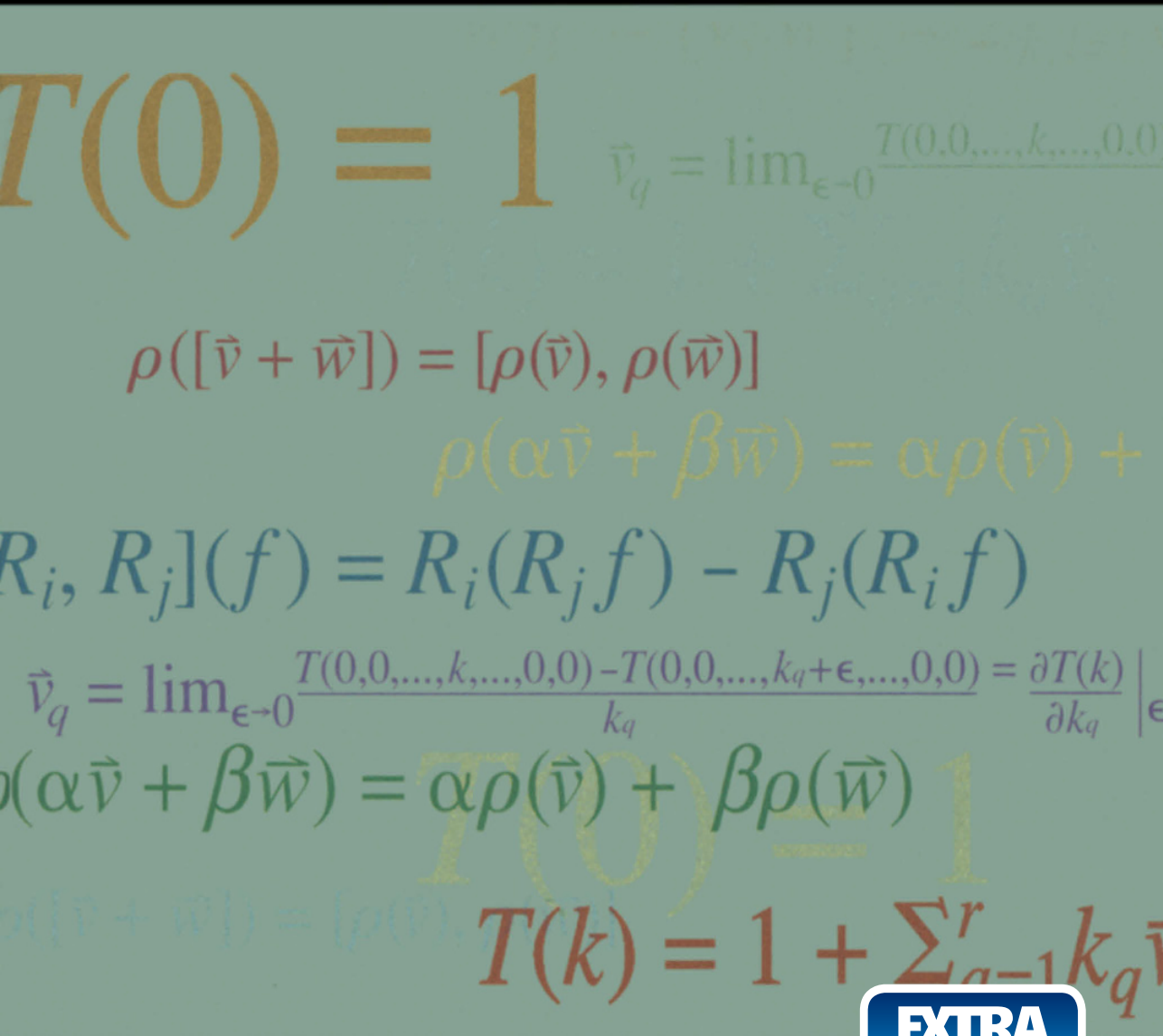
Symmetry Analysis of Differential Equations with *Mathematica*®



Gerd Baumann

Symmetry Analysis of Differential Equations with *Mathematica*®

Gerd Baumann

Symmetry Analysis of Differential Equations with *Mathematica*®

Springer Science+Business Media, LLC

Gerd Baumann
Department of Mathematical Physics
University of Ulm
Ulm D-89069
Germany

Library of Congress Cataloging-in-Publication Data
Baumann, Gerd.
Symmetry analysis of differential equations with Mathematica /
Gerd Baumann.
p. cm.
Includes bibliographical references and indexes.
Additional material to this book can be downloaded from http://extras.springer.com.

ISBN 978-1-4612-7418-6 ISBN 978-1-4612-2110-4 (eBook)
DOI 10.1007/978-1-4612-2110-4

1. Differential equations—Numerical solutions—Computer programs.
2. Symmetry (Physics) 3. Mathematica (Computer program language)
I. Title
QA371.B36 1998
515'.35—dc21 98-26975

Printed on acid-free paper.

Originally published by Springer-Verlag New York, Inc.
Softcover reprint of the hardcover 1st edition 2000

TELOS®, The Electronic Library of Science, is an imprint of Springer-Verlag New York, Inc.

Production managed by Steven Pisano; manufacturing supervised by Jacqui Ashri.
Typeset by Integre Technical Publishing Co., Inc., Albuquerque, NM.

9 8 7 6 5 4 3 2 1

Dedicated to my beloved one and only

Preface

The purpose of this book is to provide the reader with a comprehensive introduction to the applications of symmetry analysis to ordinary and partial differential equations. The theoretical background of physics is illustrated by modern methods of computer algebra. The presentation of the material in the book is based on *Mathematica* 3.0 notebooks. The entire printed version of this book is available on the accompanying CD. The text is presented in such a way that the reader can interact with the calculations and experiment with the models and methods. Also contained on the CD is a package called *MathLie*—in honor of Sophus Lie—carrying out the calculations automatically. The application of symmetry analysis to problems from physics, mathematics, and engineering is demonstrated by many examples.

The study of symmetries of differential equations is an old subject. Thanks to Sophus Lie we today have available to us important information on the behavior of differential equations. Symmetries can be used to find exact solutions. Symmetries can be applied to verify and to develop numerical schemes. They can provide conservation laws for differential equations. The theory presented here is based on Lie, containing improvements and generalizations made by later mathematicians who rediscovered and used Lie's work. The presentation of Lie's theory in connection with *Mathematica* is novel and vitalizes an old theory. The extensive symbolic calculations necessary under Lie's theory are supported by *MathLie*, a package written in *Mathematica*.

Each chapter of the present book includes theoretical considerations and practical applications of *MathLie* and *Mathematica*. The *Mathematica* examples range from simple definitions to complete notebooks discussing specific problems. The examples include definitions of general derivatives, derivations and solutions of determining equations, drop formations in liquids, and the first atomic explosion.

The end of a definition and a theorem in the text is indicated by ○. The end of an example is indicated by □. On the CD, *MathLie* and *Mathematica* notations in the text are denoted by the color dark red. *Mathematica* input is given in red while the output is in blue.

I wish to express my gratitude to Peter Olver, Willy Hereman, and Mike Mezzino for reading the manuscript. My appreciation goes to Gerda Göler and Joachim Engelmann for proofreading the text. I also acknowledge contributions by Gernot Haager, Gerald Landhäußer, and Ronald Schmid.

Any suggestions and comments related to the book or to *MathLie* are most appreciated. Please send your e-mail to Gerd.Baumann@physik.uni-ulm.de or visit my home page at http://www.physik.uni-ulm.de/math/gbaumann/bau.html.

Gerd Baumann

Contents

1

Introduction

Symmetry principles play an important role in the laws of nature. They summarize the regularities of the laws that are independent of the specific dynamics. Thus, invariance principles provide a structure and coherence to the laws of nature, just as the laws of nature provide a structure and coherence to the set of events. In fact, it is hard to imagine that progress could have been made in deducing the laws of nature without the existence of certain symmetries. The ability to represent experiments in different places at different times is based on the invariance of the laws of nature under space-time translations. Without regularities embodied in the laws of physics, we would be unable to make sense of physical events; without regularities in the laws of nature, we would be unable to discover the laws themselves. Today we realize that symmetry principles are even more powerful—they dictate the form of the laws of nature.

An important implication of symmetry in physics and in mathematics is the existence of conservation laws. For every global continuous symmetry (i.e., a transformation of a physical system that acts the same way everywhere and at all times), there exists an associated time-independent quantity. This connection went unnoticed until 1918, when Emmy Noether [1918] proved her famous theorem relating symmetry and conservation laws. Thus, due to the invariance of the laws of physics under spatial transformations, momentum is conserved; due to time-translational invariance, energy is conserved; and due to the invariance under a change in phase of the wave function of charged particles, electric charge is conserved. It is essential that the symmetry be continuous; namely that it is specified by a set of parameters that can be varied continuously, and that the symmetry transformation can be arbitrary close to the identity transformation. The discrete symmetries of nature, such as time-reversal invariance or mirror reflection, do not lead to new conserved quantities.

Until the 20th century, principles of symmetry played only a small role in theoretical physics. The Greeks and others were fascinated by the symmetries of objects and believed that these were mirrored in the structure of nature. Even Kepler attempted to impose his notions of symmetry on the motion of the planets. Newton's laws of mechanics embodied symmetry principles realized in the equivalence of inertial frames, or Galilean invariance. These symmetries implied conservation laws. In the 19th century, this ancient situation changed dramatically beginning with Lie. His great advance in 1873 was to put symmetry first, to regard the symmetry principle as the primary feature of nature that constrains the allowable dynamical laws. Lie applied his theory to different models given by differential equations. In this way, he created the symmetry analysis of differential equations.

Thus, symmetry analysis of differential equations is an old theme in the field of applied mathematics and physics. The subject of the present book started in the late 19th century with the work of Marius Sophus Lie. The theory in its basic form was developed and applied by Lie during the period 1872–1899. Until now there have been extensions of the theory and a continuous application in physics, especially in hydrodynamics, mechanics, electrodynamics, quantum theory, statistical mechanics, field theory, particle physics, etc. Today, symmetry analysis is one of the rare theories which allows one to derive solutions of differential equations in a completely algorithmic way. Among other solution procedures like the inverse scattering theory and the Hirota technique, Lie's theory takes an outstanding position. Although Lie's theory is applicable to any sort of differential equations, the other theories are commonly useful in the solution of so-called completely integrable equations or underlie some other restrictions. However, we will present here an overview of Lie's procedure and its application to some examples which are either of practical or theoretical interest. During the last few decades, there has been a revival of interest in Lie's theory and significant progress has been made due to the efforts of several mathematicians and physicists.

Lie's theory is powerful, versatile, and fundamental to the development of systematic procedures that lead to invariant solutions of boundary value problems. As this theory is not based on linear operators, superposition or other requirements of linear solution techniques, they are applicable to both linear and non-linear differential models.

A central problem in physics, mathematics, and engineering is to find solutions of a given system of differential equations. These equations may be linear or nonlinear. The generic case of practical problems which handle ordinary as well as partial differential equations are nonlinear models. Let us summarize all these equations by the notation

$$\Delta^i(x, u_{(k)}) = 0, \qquad i = 1, 2, \ldots, m \tag{1.1}$$

where x is a p-dimensional vector of independent variables and $u_{(k)}$ denotes the derivatives up to order $k = 0, 1, \ldots$ of a q-dimensional vector of dependent variables u. The central question for such a general system of nonlinear partial or ordinary differential equations is: Can we find a universal procedure which gives us solutions for this system of equations? We do not try to find the general solution but simply a solution. That this is not a trivial task has been known for a long time. In the last century, Lie pointed out this central problem in a foreword to his lecture Differentialgleichungen as follows:

> Die älteren Untersuchungen über gewöhnliche Differentialgleichungen, wie man sie in den gebräuchlichen Lehrbüchern findet, bilden kein systematisches Ganzes. Man entwickelte specielle Integrationstheorien z.B. für die homogenen Differentialgleichungen, für die linearen Differentialgleichungen und andere specielle integrable Formen von Differentialgleichungen. Es war aber den Mathematikern entgangen, daß diese speciellen Theorien sich unter eine allgemeine Methode unterordnen lassen. Das Fundament dieser Methoden ist der Begriff der **infinitisemalen Transformation** und der damit auf das engste zusammenhängende Begriff der **eingliedrigen Gruppe**.
>
> —Auszug aus *Differentialgleichungen* von Sophus Lie, Leipzig 1891

The translation of these comments is:

> The older examinations on ordinary differential equations as found in standard books are not systematic. The writers developed special integration theories for homogeneous differential equations, for linear differential equations, and other special integrable forms of differential equations. However, the mathematicians did not realize that these special theories are all contained in the term **infinitesimal transformations**, which is closely connected with the term of a **one parametric group**.
>
> —Quotation from *Differentialgleichungen* by Sophus Lie, Leipzig 1891

One of the main deficiencies of Lie's theory is the tremendous amount of work necessary to derive a solution of a given differential equation. This work of algebraic manipulation increases if the differential equation depends not only on one but on several independent variables. It increases even more if we study a system of equations. For such general situations, it may happen that we have to handle hundreds of equations to find a single solution. In the past, this large amount of work was a severe barrier for using Lie's theory. Today, we are able to overcome the problems of algebraic manipulation of this great number of expressions. Using computer algebra systems like *Mathematica*, *Maple*, *Macsyma*, or *Axiom*, to name the more powerful systems, we can manage the laborious work in an up-to-date fashion.

In this book, we prefer *Mathematica* to carry out the calculations. An overview of programs written in other programming languages is given in recent articles by Hereman [1994,1996]. Hereman shows that there exists a large number of programs with different capabilities in different programming languages. Our choice of using *Mathematica* as a programming language has been motivated by several reasons. First, *Mathematica* is a language which is easy to use. Second, *Mathematica* allows a direct formulation of the problem. Third, *Mathematica* is a very powerful high-level programming language designed for pattern matching, which is needed in Lie's theory to find structures of a certain type. Finally, *Mathematica* allows a very simple formulation of the theory of Lie. These four points were considered in our decision process to choose the programming language.

To appreciate the present text, the reader should have a moderate understanding of *Mathematica*. You will find the explanations for the commands used in the examples in Appendix B.

Lie's classical theory is a source for various generalizations. Among these generalizations is the *non-classical method* of Bluman and Cole [1974], which was the focus of some research in the last few years uncovering the connection with the direct reduction method of Clarkson and Kruskal [1989]. A recent development in Lie's theory by Baikov, Gazizov, and Ibragimov [1989] is the introduction of approximate symmetries, allowing the asymptotic solutions for a range of parameters to be derived. Another adornment of Lie's classical theory is the introduction of generalized symmetries, which is extensively discussed by Olver [1986]. Generalized symmetries are symmetries which are a generalization of contact symmetries. The generalization of Lie's theory releases one or more of the basic properties obeyed by the classical theory.

The fundamentals of Lie's theory of symmetry analysis of differential equations are based on the invariance of the equation under a transformation of independent and dependent variables. This transformation forms a local group of point transformations which establishes a diffeomorphism on the space of independent and dependent variables, mapping solutions of the equations to solutions.

The description of the fundamentals of Lie's theory, Lie groups, and Lie algebras is the starting point for our discussions in Chapter 2. Chapter 3 presents fundamental aspects of derivatives and their definitions in *Mathematica*. Chapter 4 on ordinary differential equations discusses the application of Lie's integration theory in connection with point symmetries. Chapter 5 deals with point symmetries in connection with partial differential equations. Several examples demonstrate the broad application of Lie's theory. Chapter 6 extends the classical point symmetries to non-classical symmetries. In Chapter 7, potential symmetries of partial differential equations are examined. The recent development of approximate symmetries is

contained in Chapter 8. The generalized symmetries of PDEs and second-order ODEs is presented in Chapter 9. The last chapter contains a special topic of symmetry analysis, i.e., the automatic solution of a system of overdetermined equations.

The material contained in the chapters is based on theoretical considerations necessary to understand what is going on in the related functions of *MathLie*. *MathLie* is a *Mathematica* package supporting the calculations in the book and more. A full version of *MathLie* accompanies the book on CD-ROM. A great number of examples contained in each chapter demonstrate the broad application of Lie's theory in connection with *MathLie*. The examples are designed in such a way that the reader can take an active part by calculating the results interactively. This opens the way to experimentation with the calculations. Thus, the present book is not a book just for reading but a book for experimental mathematics and physics.

2

Elements of Symmetry Analysis

At the beginning we will introduce some basic concepts which will be important throughout the whole book. First, we define the general properties of a group. These group properties are extended to Lie groups in the next step. The related Lie algebra connected with the Lie group is then introduced. We also introduce the notion of a vector field which is closely related to Lie algebras. We present all these highly abstract terms in connection with *Mathematica*. Different examples serve to vitalize the mathematical expressions. This chapter serves also to describe the first steps in *Mathematica* and introduce its notation. The elementary representation of mathematical expressions in *Mathematica* provides the connection between mathematics and computer algebra.

2.1. Groups and Lie Groups

It is the purpose of this section to record, for later reference, some of the results from group theory which will be needed in the text. The notation of a group is introduced in this chapter and the most important properties of group elements are deduced. Illustrations are given from a few very simple groups. Proofs will be minimal or omitted.

2.1.1 Groups

Although we shall soon come to some illustrative examples, it is worth beginning with the abstract definition of a group which is very simple and yet leads to many important consequences.

Definition: Group

A set $\mathfrak{G}$ of elements $\{G_1, G_2, G_3, \ldots\}$ is said to form a group if a rule of composition is defined for the elements satisfying certain conditions. The result of a multiplication involving two elements G_i and G_k is called the product or composition of the two elements and is written $G_i \oplus G_k$. The conditions which such a product has to satisfy are as follows:

(i) Closure relation:

The product $G_i \oplus G_k$ of any two elements is itself an element in the set, i.e.,

$$G_i \oplus G_k = G_j \text{ with } G_j \in \mathfrak{G}$$

(ii) Associativity:

If three elements G_i, G_k, and G_j are multiplied, it does not matter which product is carried out first, i.e.,

$$G_i \oplus (G_k \oplus G_j) = (G_i \oplus G_k) \oplus G_j = G_i \oplus G_k \oplus G_j .$$

This equality shows that the use of brackets is not necessary.

(iii) Identity element:

One element of the set $\mathfrak{G}$, denoted by E and called the identity element, must have the properties

$$E \oplus G_i = G_i \text{ and } G_i \oplus E = G_i$$

for all $G_i \in \mathfrak{G}$.

(iv) Inverse:

To each element G_i in the set $\mathfrak{G}$, there corresponds another element in the set, denoted by G_i^{-1} and called the inverse, which has the properties

$$G_i \oplus G_i^{-1} = E = G_i^{-1} \oplus G_i . \quad \bigcirc$$

In general, it is not permissible to change the order of multiplication of group elements; i.e., $G_i \oplus G_j$ is not, in general, the same element as $G_j \oplus G_i$. A group which satisfies this exception, $G_i \oplus G_j = G_j \oplus G_i$, is called an Abelian group. Its elements are said to commute. The axioms (i)–(iv) stated above are the main ingredients of group theory. However, these properties are abstract entities which need a practical realization. A convenient method of recording the multiplication, $G_i \oplus G_j = G_k$, of elements of a particular group $\mathfrak{G}$ is to build the multiplication table in which the rows and columns are labeled by the group elements and the result

G_k of the multiplication is entered at the intersection of the row G_i and the column G_j. The definition of a group implies that every group element must appear once and only once in each row and in each column.

Deliberately, we did not specify the number of elements in the group. In fact, the number may be finite or it may be infinite. Correspondingly, the group is called a finite or an infinite group. In this book, we shall encounter both groups since they both are of importance in symmetry analysis. If we find a finite group order, we will denote the number of elements n as the order of the group.

The simplest examples of group elements are natural numbers with ordinary multiplication. We will discuss two examples.

Example 1

Let us assume we only know the two numbers 1 and –1 and the ordinary multiplication as group operation. The identity of this group is clearly 1. The inverse of the identity is again the identity. The inverse of –1 is –1 itself. The properties of this group are contained in the group multiplication table *tab1* below. We can create the group table in *Mathematica* by defining the group G as the set

```
G = {-1, 1};
```

Using all combinations of the elements G_i in the group table, we get

```
tab1 = MatrixForm[Table[G[[i]] G[[j]], {i, 1, 2}, {j, 1, 2}]]
```

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

representing the core of the multiplication table. □

Example 2

A slightly larger group of the same kind is the set of numbers

```
G2 = {-1, 1, -I, I};
```

which possesses the multiplication table

```
tab2 = MatrixForm[Table[G2[[i]] G2[[j]], {i, 1, 4},
   {j, 1, 4}]]
```

$$\begin{pmatrix} 1 & -1 & I & -I \\ -1 & 1 & -I & I \\ I & -I & -1 & 1 \\ -I & I & 1 & -1 \end{pmatrix}$$

Because ordinary multiplication is used in both examples, these groups must be Abelian since it does not matter in which order the elements of the sets are used. To demonstrate this, let us exchange the *j*th element by the *i*th element in the table above.

```
MatrixForm[Table[G2[[j]] G2[[i]], {i, 1, 4}, {j, 1, 4}]]
```

$$\begin{pmatrix} 1 & -1 & I & -I \\ -1 & 1 & -I & I \\ I & -I & -1 & 1 \\ -I & I & 1 & -1 \end{pmatrix}$$

Comparing both group tables of *G2* demonstrates that the order of the group elements in the multiplication does not matter. □

For physical systems, rotations are of considerable importance. It is well known that various sets of rotations form groups. The rotations were one of the favorite groups used by Lie to demonstrate the action of his examinations. In reminiscence of this historical note, let us examine a few examples related to that topic. The law of multiplication in this case is defined by transition from one location to another—if a rotation R_1 carries a system from position A to position B and if R_2 carries it from B to C, then the product $R_1 \oplus R_2$ carries it from A to C. It is obvious that this definition of multiplication can, in general, not create an Abelian group. Of course, rotations about a common axis are Abelian. However, rotations in higher-dimensional spaces in general do not commute. Let us demonstrate these two statements by simple examples.

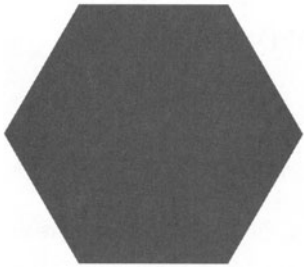
Example 3

To illustrate the commutative and non-commutative properties of rotations, let us define a function which carries out a two- and a three-dimensional rotation of an object. The function Rotation[] uses the standard package Geometry`Rotation` to represent the rotation matrices in two and three dimensions. The function Rotation[] will take a polygon and an angle as input parameters. This function generates the geometrical shape of the object and carries out a rotation. First, let us define the geometrical object by a polygon

```
hexagon = Polygon[Table[{Cos[i], Sin[i]},
    {i, 0, 2 π, 2 π/6}]];
```

Our favorite object is a hexagon which can be graphically displayed by the following lines:

```
Show[Graphics[{RGBColor[1, 0, 0],
   hexagon}], AspectRatio → Automatic]
```

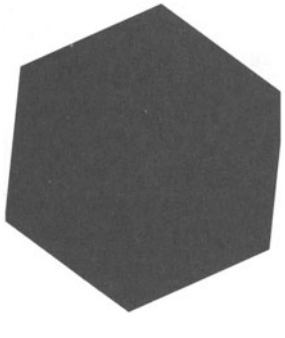


This object will be rotated by our function Rotation[]. In a two-dimensional space, the function is defined in *Mathematica* terms by

```
<< "Geometry`Rotations`"

Clear[Rotation];
Rotation[polygon_Polygon, angle_] :=
 Block[{points},
  points = polygon /. Polygon[x___] → x;
  Polygon[(Rotate2D[#1, angle]&) /@points]]
```

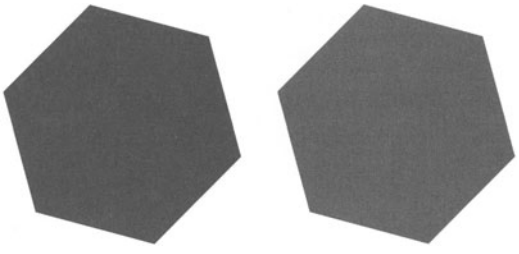
The function Rotation[] needs a polygon and the angle of rotation as input quantities. It returns a clockwise-rotated polygon. The application of the function Rotation[] on our hexagon gives us

```
Show[Graphics[{RGBColor[0.000, 0.000, 1.000],
   Rotation[hexagon, π/5]}], AspectRatio → Automatic]
```



The result shows that the original hexagon is rotated by an angle of $\pi/5$. Having the function Rotation[] available, we are able to check the group properties of a group. Let us start by testing if the two rotations commute. If we assume that the first rotation R_1 rotates the hexagon through an angle $-\pi/3$ and the second R_2 through an angle $-\pi/4$, we can combine the rotations either by $R_1 \oplus R_2$ or $R_2 \oplus R_1$. These two mathematical relations are realized in *Mathematica* by the following lines. The result of the two different sequences of rotations is shown in the following:

```
Show[GraphicsArray[
  {Graphics[{RGBColor[0, 0, 1],
     Rotation[Rotation[hexagon, π/3], -π/4]},
    AspectRatio → Automatic],
   Graphics[{RGBColor[1, 0, 0],
     Rotation[Rotation[hexagon, -π/4], π/3]},
    AspectRatio → Automatic]}]]
```

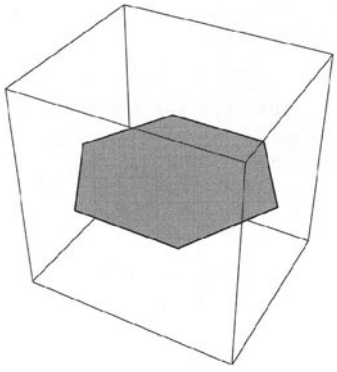


The result of the two sequences of rotations is the same. Thus, we conclude from this graphical experiment that these two rotations in the plane commute. The net effect of the two rotations is a total rotation through an angle of $\pi/12$. To illustrate the non-commutative property of rotations in higher dimensions, let us examine rotations in $\mathbb{R}^3$. For example, let R_Z be a rotation through an angle $\pi/5$ about the z-axis and R_X a rotation through $\pi/7$ about the x-axis. The geometrical object we will rotate is again a hexagon located in the (x, y)-plane. The polygon in *Mathematica* is represented in three dimensions by

```
hexagon = Polygon[Table[{Cos[i], Sin[i], 0},
    {i, 0, 2π, 2π/6}]];
```

with its z coordinate set equal to zero. The hexagon is displayed in three dimensions by

```
Show[Graphics3D[{RGBColor[1.000, 0.000, 0.000],
   hexagon}], AspectRatio → Automatic]
```



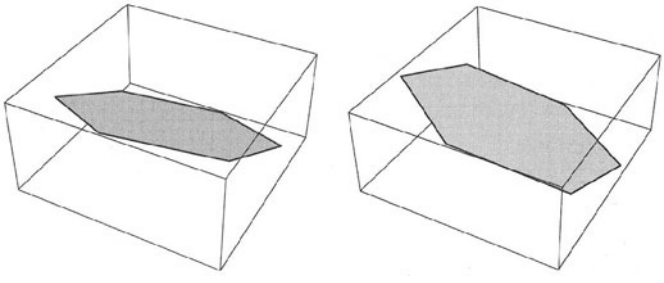
To carry out the rotations about the three coordinate axes, we define three functions, RotationX[], RotationY[], and RotationZ[], in *Mathematica*. The arguments are again the geometrical object and the angle of rotation with respect to the denoted axis.

```
Clear[RotationX, RotationY, RotationZ];
RotationZ[polygon_Polygon, angle_] :=
 Block[{points, mat1, mat2},
  points = polygon /. Polygon[x___] → x;
  mat1 = RotationMatrix2D[angle];
  mat2 = IdentityMatrix[3];
  mat2[[1, 1]] = mat1[[1, 1]];
  mat2[[2, 1]] = mat1[[2, 1]];
  mat2[[1, 2]] = mat1[[1, 2]];
  mat2[[2, 2]] = mat1[[2, 2]];
  Polygon[(mat2 . #1&) /@points]];

RotationX[polygon_Polygon, angle_] :=
 Block[{points, mat1, mat2},
  points = polygon /. Polygon[x___] → x;
  mat1 = RotationMatrix2D[angle];
  mat2 = IdentityMatrix[3];
  mat2[[2, 2]] = mat1[[1, 1]];
  mat2[[2, 3]] = mat1[[1, 2]];
  mat2[[3, 2]] = mat1[[2, 1]];
  mat2[[3, 3]] = mat1[[2, 2]];
  Polygon[(mat2 . #1&) /@points]];
```

```
RotationY[polygon_Polygon, angle_] :=
 Block[{points, mat1, mat2},
  points = polygon /. Polygon[x___] → x;
  mat1 = RotationMatrix2D[angle];
  mat2 = IdentityMatrix[3];
  mat2[[1, 1]] = mat1[[1, 1]];
  mat2[[1, 3]] = mat1[[1, 2]];
  mat2[[3, 1]] = mat1[[2, 1]];
  mat2[[3, 3]] = mat1[[2, 2]];
  Polygon[(mat2 . #1&) /@points]]
```

The two rotations about the z- and x-axes mentioned above are graphically represented by the following lines:

```
Show[GraphicsArray[
  {Graphics3D[
    RotationX[RotationZ[hexagon, π/5], π/7],
    AspectRatio → Automatic,
    ViewPoint → {1.300, -2.400, 2.000}],
   Graphics3D[
    RotationZ[RotationX[hexagon, π/7], π/5],
    AspectRatio → Automatic,
    ViewPoint → {1.300, -2.400, 2.000}]}]]
```



The graphic shows that the two rotations applied to the same object in a different order results in two different states of the hexagon. Thus, by a simple example, we graphically verified that two rotations in a three-dimensional space are non-commutative. The reader may check this result by different rotations about different axes using different angles of rotation. □

Another important term in group theory governing the relations between two groups is the notion of isomorphism.

2.1.2 Isomorphism

The given definition of a group is very abstract, yet general. With respect to this generality, it sometimes happens that two groups whose elements are defined in very different ways may nevertheless be related so closely that they may be regarded as the same group. This fact is expressed in the following definition.

Definition: Isomorphic groups

We say that two groups G and H are isomorphic if a one-to-one correspondence $G_i \leftrightarrow H_i$ may be set up between the elements G_i of the group G and the elements H_i of H, in such a way that if $G_i \oplus G_k = G_j$, then $H_i \oplus H_k = H_j$. ○

Closely related to the term isomorphism is the subject of homomorphism. The word homomorphism is used for such a relationship if the one-to-one correspondence is absent. Due to the definition of isomorphism, two isomorphic groups have the same group multiplication table with possible re-ordering of the group elements. Thus, the knowledge of the isomorphism of two groups helps to avoid repetitions and to draw useful analogies between the groups.

2.1.3 Lie Groups

Lie groups are special groups which have an additional property apart from the group properties. In addition to the basic group properties, a Lie group carries the structure of a manifold, where a manifold is a topological space which resembles Euclidean space locally. A differentiable manifold is a manifold for which this resemblance is sharp enough to allow partial differentiation and, consequently, all the features of differential calculus on the manifold. In studying Lie groups, we may, therefore, combine calculus, algebra, and topology. The present section aims at showing the sense in which the global study of a Lie group may be reduced to its local study. In the next section, we shall go even further, showing that the study of the local structure can be reduced to the study of the infinitesimal structure. Lie groups are extremely useful in the theory of transformation and in the examination of differential equations. The notion of a Lie group was introduced by Weyl [1928] at the beginning of this century. Weyl used the following definition to distinguish Lie groups from classical groups.

Definition: Lie group

A Lie group is a group which, in addition to the group properties, carries the structure of a differentiable manifold. More precisely, we require that a Lie group $\mathbb{G}$ be C^∞ manifold endowed with a group structure in which multiplication and the inversion are C^∞ operations. ○

The essential feature of a Lie group is that it satisfies the properties (i)–(iv) and carries the structure of a smooth manifold. This means that the group elements G_i can be continuously varied. Thus, a Lie group is a group $\mathbb{G}$ which also carries the structure of a manifold in such a way that both the group operation $\mathbb{G} \oplus \mathbb{G} \to \mathbb{G}$ and the inversion are smooth maps between manifolds. In the following, we will demonstrate these descriptions by a few examples.

Example 1

The first simple example of a Lie group is the real line $\mathbb{R}^1$ with ordinary addition as the group multiplication. Let us denote this group by $\mathbb{A}$. If we add two real numbers, we get a real number as a result. We all know that we can add three real numbers in any order to get the same result. The identity element of this group in $\mathbb{R}^1$ is zero and the inverse are all the negative real numbers. Thus, we can map $\mathbb{R}^1 \times \mathbb{R}^1 \to \mathbb{R}^1$, and the inversion as a smooth map also exists. These properties of addition for the real numbers are actually implemented in *Mathematica* and are accessible by the function N[] converting rational numbers to real numbers. The + sign represents the multiplication of the group $\mathbb{A}$. The manifold on which all these operations are possible are the set of real numbers $\mathbb{R}^1$. □

Example 2

A more sophisticated example for a Lie group is given by continuous matrix groups, or, more generally, continuous groups of linear transformations of a vector space, called linear Lie groups. The set of all non-singular $n \times n$ matrices form the group known as general linear group $GL(n,\mathbb{R})$. A subset of all $n \times n$ matrices with determinant 1 form a group called the unimodular group which is denoted by $SL(n,\mathbb{R})$. The orthogonal group $O(n)$ is the group of $n \times n$ matrices that satisfy $M \oplus M^T = 1$. A special orthogonal group $SO(3)$ is connected with rotations.

Studying the properties of continuous matrix groups, we start with the two-dimensional matrices

$$M_a = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}.$$

This representation of a linear group $SL(2,\mathbb{R})$ immediately shows the property

$$M_a \oplus M_b = M_{a+b},$$

from which we can conclude that the group of two-dimensional matrices is isomorphic to the group $\mathbb{A}$ of our first example. To support this conclusion, let us examine the properties of the two-dimensional matrix group by using *Mathematica*. First, let us define a function allowing us the representation of M_a. Afterward, we use this function to check the group axioms (i)–(iv) for this representation of a group. Since M_a depends on one continuous parameter a, we define the matrix M_a by

```
M[a_] := {{1, a}, {0, 1}}
```

The matrix function M[] uses two nested lists to represent the two-dimensional matrices needed. We check the axioms by using the matrix product for the group multiplication. This type of product is denoted by a lower dot in *Mathematica*. In the sequel, we use the function MatrixForm[] to represent the resulting matrices in a two-dimensional table. Multiplying two different matrices M[a] and M[b], we find

```
MatrixForm[M[a] . M[b]]
```

$$\begin{pmatrix} 1 & a+b \\ 0 & 1 \end{pmatrix}$$

We immediately verified that axiom (i) is satisfied. The property of associativity is checked by interchanging the multiplication order

```
MatrixForm[(M[a] . M[b]) . M[c]]
```

$$\begin{pmatrix} 1 & a+b+c \\ 0 & 1 \end{pmatrix}$$

```
MatrixForm[M[a] . (M[c] . M[b])]
```

$$\begin{pmatrix} 1 & a+b+c \\ 0 & 1 \end{pmatrix}$$

The inverse element of the group corresponds to the inversion of the matrix. Matrices are inverted in *Mathematica* with the help of the function Inverse[]. Inverse[] returns the inverse of a square matrix. The application of this function shows us

```
mi = Inverse[M[a]]; mi // MatrixForm
```

$$\begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}$$

which is the inverse of M_a stored in the variable *mi*. Next, we can check axiom (iv) by

```
ident = mi.M[a]; ident // MatrixForm
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Thus, we find the identity element of the two-dimensional matrices M_a as the identity matrix in two dimensions. It is obvious from the calculations that the set $\mathfrak{M} = \{ M_a \mid a \in \mathbb{R} \}$ is a representation of addition in $\mathbb{R}^1$, thus the two Lie groups $\mathbb{A}$ and $\mathfrak{M}$ are isomorphic. □

Generally speaking, a representation of a group $\mathfrak{G}$ on a vector space V is a homomorphism from the group $\mathfrak{G}$ into an invertible linear transformation of V. These representations must not be matrix representations but can be defined on the infinite dimensional vector space $C^\infty(\mathbb{R})$, which represents the space of infinitely differentiable functions in one dimension.

Example 3

The addition in $\mathbb{R}^1$ can also be represented by a translation in the space of $C^\infty(\mathbb{R})$ functions. Let us assume that we can define an operator T_a, which acts on a function defined on $\mathbb{R}^1$ in the following way:

$$T_a\, f(x) = f(x + a).$$

The shift a translates the argument of the function f by a step a to the left. The definition of a shift operator T_a in *Mathematica* looks like

```
T_a_[f_] := f /. x → x + a
```

This simple definition assumes that the function f depends exclusively on x. The sequence of instructions on the right-hand side of the definition sign (:=) means that the argument x of f, if any, is replaced by $x + a$. The properties of the group are now checked by applying the operator T_a on functions $f(x)$. The identity element of the Lie group is given by $a = 0$,

```
T0[f[x]]
```

```
f[x]
```

The inverse element is represented by a negative shift $-a$. We check the inverse behavior of the transformation by applying the inverse element to a regular element of the group. The expected result is the identity

```
T-a [Ta [f[x]]]
f[x]
```

The property of associativity is expressed by the iterated application of the translation operator $T_a[]$ for different translations a, b, and c and the interchange of two parameters

```
Tc [Tb [Ta [f[x]]]]
f[a + b + c + x]
Tb [Tc [Ta [f[x]]]]
f[a + b + c + x]
```

Again, we observe that the Lie group of translations in $\mathbb{R}^1$ is isomorphic to the group of addition $\mathbb{A}$. In conclusion, we can say that the same Lie group can be represented by different tools like addition of real numbers, matrix multiplication, and translations of functions. These different tools for representing the behavior are known as representations of the group. The idea of a representation of a Lie group helps to clarify the subtle distinction between an abstract group and a variety of its realizations. Thus, $\mathbb{A} = \mathbb{R}^1$, the set of matrices M_a, and the translations T_a are all distinct but isomorphic representations of the same abstract group. □

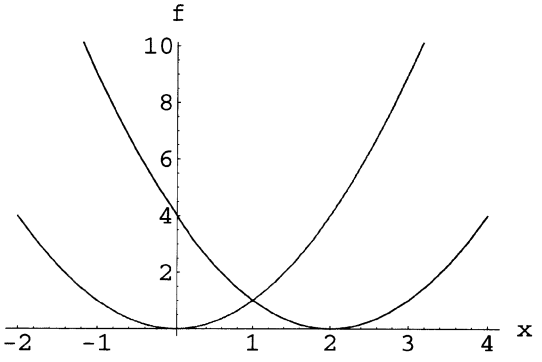
Apart from an isomorphism which is invertible, the term homomorphism is an important quantity in group theory. A Lie group homomorphism is a smooth map $\Phi : \mathsf{G} \to \mathsf{H}$ between two Lie groups respecting the group operations. If Φ has a smooth inverse, it determines an isomorphism between G and H, otherwise it is a homomorphism.

Understanding the action of the group clearly, we discuss the example of a translation a second time. This kind of symmetry is a symmetry frequently recognized in the analysis of differential equations. To illustrate the action of such a group, we will realize it by a graphical representation.

Example 4

For example, let us study the group properties of a parabola $f(x) = x^2$ under the action of our function T_a. Applying T_a to x^2, we mathematically get the expression $x^2 + 2ax + a^2$. The expanded result represents the translation T_a in a more or less mixed form containing products of a and x. However, the action of the group is much simpler to grasp if we present it graphically. The shift a along the x-axis is clearly shown in the figure below.

```
Plot[
 Evaluate[{x^2, T_-2[x^2]}], {x, -2, 4},
 PlotStyle → {RGBColor[1.000, 0.000, 0.000],
   RGBColor[0.000, 0.000, 1.000]}, AxesLabel → {"x", "f"}]
```



The command Evaluate[] used in the function Plot[] forces *Mathematica* to do the calculations first and then display the results. The shift of translation in the example is set to $a = -2$. We clearly observe that the parabola is translated to the right by the length a. □

Example 5

Another example frequently encountered in symmetry analysis of differential equations is the scaling group. A scaling transformation reduces or enlarges an object depending on the amount of the scaling factor. A scaling of a geometrical object is carried out practically by multiplying the coordinates of the object by the scaling factor. A function allowing this operation can be defined in *Mathematica* by

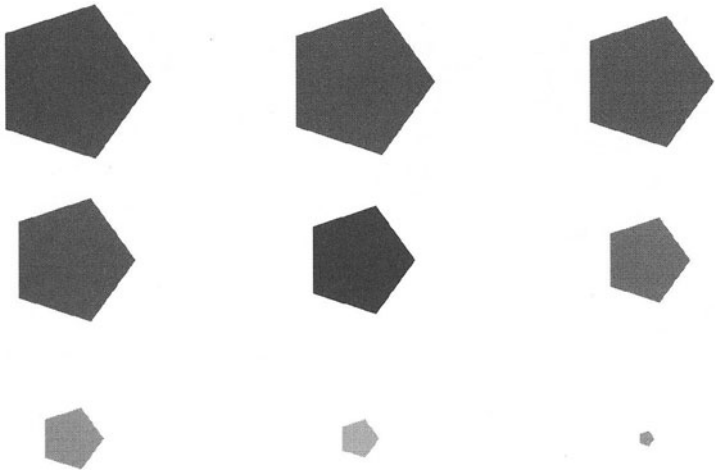
```
Clear[Scaling];
Scaling[object_Polygon, factor_] :=
  points = object /. Polygon[X___] → Polygon[factor X]
```

This definition assumes that the geometric object is given by a polygon. Applying this function to an object, for example, a pentagon, we can reduce or enlarge the figure depending on the second argument of the function. Choosing the factor greater than 1, we stretch the object; taking a factor smaller than 1, we shrink it. The pentagon is generated by

```
pentagon = Polygon[Table[{Cos[i], Sin[i]},
    {i, 0, 2π, 2π/5}]];
```

Let us now examine how the shape of the pentagon changes when the scaling factor is changed. An animation of this change of scaling factors helps us to recognize the meaning of scaling. The following *Mathematica* code contains a Do[] loop which allows the decrease of the scaling factor. For the animation, we choose the scaling factor in the range from 1 to 0.1 in steps of 0.05. The action of the scaling is boosted by changing the color of the pentagon.

```
Do[
 Show[Graphics[{Hue[i], Scaling[pentagon, i]},
   AspectRatio → Automatic],
  PlotRange → {{-1, 1}, {-1, 1}}], {i, 1, .1, -.05}]
```



The animation of the scaling transformation shows us the action of the group. Every time we reduce the scaling factor, the pentagon is reduced but keeps its shape, meaning that we create self-similar objects of the same type. This kind of symmetry transformation satisfies all the properties stated in properties (i)–(iv) of a group. The reader may verify this easily. □

So far we discussed the essentials of group theory including Lie groups. The discussed topics are all relevant for the examination of differential equations. In the following section, we will discuss a related term, the so-called Lie algebra, representing the algebraic properties of a Lie group.

2.2. Lie Algebras

In this section, we show how the study of a Lie group G may be greatly simplified by considering the so-called tangent space V of G around the identity of the group. We shall show how a multiplication may be introduced in V and that the resulting algebraic structure—called a Lie algebra—determines the local structure of a group. Thus, two groups will be locally isomorphic if and only if their Lie algebras are isomorphic. The Lie algebra is a finite dimensional algebra. Therefore, the local study of Lie groups is entirely equivalent to the study of certain finite dimensional linear algebraic structures.

We defined a Lie group as a group connected with an analytic manifold. In this connection, it makes sense to talk about the tangent space V to that manifold and, in particular, the tangent space at the identity of a group. The tangent space itself is called a Lie algebra. To be more precise, let us consider a Lie group depending on r parameters $k_1, k_2, \ldots, k_r$. The group under consideration is also known as a continuous group with an infinite number of group elements. However, the properties of the group may be deduced from a finite number r of operators, called the infinitesimal operators. It will be convenient to use the symbol k for the set of parameters $k_1, k_2, \ldots, k_r$. Consider a representation $T(k)$ of the group G. By convention, the parameters are chosen such that the identity element has all $k_q = 0$, so that

$$T(0) = 1 \tag{2.1}$$

If all parameters k_q are small, then to first order in these parameters,

$$T(k) = 1 + \sum_{q=1}^{r} k_q \vec{v}_q \,, \tag{2.2}$$

where the $\vec{v}_q$ are some fixed linear operators, independent of the parameters k_q. These operators are called infinitesimal operators of the representation T and are given explicitly as partial derivatives

$$\vec{v}_q = \lim_{\epsilon \to 0} \frac{T(0, 0, \ldots, k_q, \ldots, 0, 0) - T(0, 0, \ldots, k_q + \epsilon, \ldots, 0, 0)}{k_q} = \frac{\partial T(k)}{\partial k_q}\bigg|_{\epsilon=0}. \tag{2.3}$$

These linear operators form the basis of a Lie algebra defined as follows.

Definition: Lie algebra

Let us consider a finite dimensional vector space V over a field K of real or complex numbers. The vector space V is called a Lie algebra over K if there is a rule of composition $(\vec{v}, \vec{w}) \to [\vec{v}, \vec{w}]$ in V which satisfies the following axioms:

(i) Antisymmetry:

$$[\vec{v}, \vec{w}] = -[\vec{w}, \vec{v}] \text{ for all } \vec{v}, \vec{w} \in V.$$

(ii) Linearity:

$$[\alpha \vec{v} + \beta \vec{w}, \vec{u}] = \alpha [\vec{v}, \vec{u}] + \beta [\vec{w}, \vec{u}] \text{ for } \alpha, \beta \in K \text{ and } \forall\, \vec{u}, \vec{v}, \vec{w} \in V.$$

(iii) Jacobi identity:

$$[\vec{v}, [\vec{w}, \vec{u}]] + [\vec{w}, [\vec{u}, \vec{v}]] + [\vec{u}, [\vec{v}, \vec{w}]] = 0 \text{ for all } \vec{v}, \vec{u}, \vec{w} \in V. \quad \bigcirc$$

The operator $[\,,\,]$ is the multiplication relation of the algebra and is known as Lie product or Lie bracket. From axiom (iii) it follows that the Lie product is, in general, non-associative. If K is the field of real numbers, then V is called a real Lie algebra; otherwise, if K is complex, V is a complex Lie algebra. A Lie algebra is said to be Abelian or commutative if for any $\vec{v}, \vec{w} \in V$ we have $[\vec{v}, \vec{w}] = 0$.

A subspace N of a Lie algebra V is a subalgebra, if $[N, N] \bigcup N$, and is an ideal if $[V, N] \bigcup N$. Clearly, an ideal is automatically a subalgebra. A maximal ideal N, which satisfies the condition $[V, N] = 0$, is called the center of V, and because $[N, N] = 0$, the center is always commutative.

Let $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n$ be a basis of the vector space V. Then the commutator $\vec{u} = [\vec{v}, \vec{w}]$, when expressed in terms of the coordinates via $\vec{v} = \sum_{i=1}^{n} v^i \vec{e}_i$, $\vec{w} = \sum_{i=1}^{n} w^i \vec{e}_i$, takes the form

$$u^i = \sum_{j,k=1}^{n} c^i_{jk} v^j w^k, \qquad i = 1, 2, \ldots, n,$$

where $[\vec{e}_j, \vec{e}_k] = \sum_{i=1}^{n} c^i_{jk} \vec{e}_i$. The numbers c^i_{jk} are called the structure constants, and n denotes the dimension of the Lie algebra.

Taking axioms (i) and (iii) into account, it is clear that the structure constants c^i_{jk} satisfy the conditions

$$c^i_{jk} = -c^i_{kj},$$

$$\sum_{m=1}^{n} (c_{im}^{l} c_{jk}^{m} + c_{jm}^{l} c_{ki}^{m} + c_{km}^{l} c_{ij}^{m}) = 0.$$

The existence of subalgebras or ideals of a Lie algebra V is reflected in certain definite restrictions on the structure constants. If $\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_k$ are the basis elements of a subalgebra, then the structure constants must satisfy the relations

$$c_{ij}^{m} = 0 \quad \text{for } i, j \leq k, \; m > k$$

and, if they are the basis elements of an ideal, then

$$c_{ij}^{m} = 0 \text{ for } i \leq k, \; m > k \text{ and an arbitrary } j.$$

So far, we defined some basic properties of a Lie algebra. These relations are useful when applied to physical problems. One of these problems is related to the algebra of Pauli matrices widely used in quantum mechanics. In the following example, we discuss the algebraic properties of the Pauli spin matrices.

Example 1

Let V be the set of all skew-Hermitian 2×2 matrices with vanishing trace. From quantum mechanics, we know that V is three dimensional. We choose the basis in V by the three matrices

$$\vec{e}_1 = \frac{1}{2}\begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}, \; \vec{e}_2 = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \; \vec{e}_3 = \frac{1}{2}\begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}. \tag{2.4}$$

These matrices are represented in *Mathematica* by

```
e1 = 1/2 {{0, -I}, {-I, 0}}; MatrixForm[e1]
```

$$\begin{pmatrix} 0 & -\frac{I}{2} \\ -\frac{I}{2} & 0 \end{pmatrix}$$

```
e2 = 1/2 {{0, -1}, {1, 0}}; MatrixForm[e2]
```

$$\begin{pmatrix} 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$$

```
e3 = 1/2 {{-I, 0}, {0, I}}; MatrixForm[e3]
```

$$\begin{pmatrix} -\frac{I}{2} & 0 \\ 0 & \frac{I}{2} \end{pmatrix}$$

The Lie product $[\vec{v}, \vec{w}]$ of V is defined in quantum mechanics by the commutator

$$[\vec{v}, \vec{w}] = \vec{v}\vec{w} - \vec{w}\vec{v} \text{ with } \vec{v}, \vec{w} \in V.$$

For our specific system of two-dimensional matrices, we replace the right-hand side of this relation by the difference of two matrix products. The Lie bracket in *Mathematica* can be expressed by

```
LieProduct[v_List, w_List] := v . w - w . v
```

Knowing the Lie bracket, it is easy to check axioms (i) to (iii) of a Lie algebra. We first demonstrate the antisymmetric behavior of the Lie product using the relation $[\vec{v}, \vec{u}] + [\vec{u}, \vec{v}] = 0$. In *Mathematica,* we get

```
MatrixForm[LieProduct[e1, e2] +
  LieProduct[e2, e1]]
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The linearity of the Lie brackets is shown by

```
MatrixForm[Simplify[LieProduct[a e1 + b e2, e3] -
   (a LieProduct[e1, e3] + b LieProduct[e2, e3])]]
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The Jacobi identity is checked with

```
MatrixForm[LieProduct[e1, LieProduct[e2, e3]] +
  LieProduct[e2, LieProduct[e3, e1]] +
  LieProduct[e3, LieProduct[e1, e2]]]
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Carrying out some experiments with LieProduct[] by interchanging the basis elements $\vec{e}_k$ in the Lie bracket, we observe that the following relation holds:

$$[\vec{e}_i, \vec{e}_k] = \Sigma_{l=1}^{3} \epsilon_{ik}^{l} \vec{e}_l \quad i, k = 1, 2, 3, \tag{2.5}$$

where ϵ_{ik}^{l} is the totally antisymmetric tensor in $\mathbb{R}^3$ (Levi-Civita density). The elements of V are linear combinations of $\vec{e}_i$ with real coefficients. In physics, the matrices $\sigma_k = 2i\vec{e}_k$ are known as Pauli matrices and satisfy the relations $[\sigma_i, \sigma_k] = 2i \sum_{l=1}^{3} \epsilon_{ik}^{l} \sigma_l$. Hence, V is the three-dimensional, real Lie algebra with structure constants $c_{ik}^{l} = \epsilon_{ik}^{l}$. If we want to check this relation for the structure

constants, we first have to define a representation of the Levi-Civita density in *Mathematica*. The function LeviCivita[] has to satisfy the following properties:

$$\epsilon_{ik}^{l} = \begin{cases} 0 & \text{if any index is equal to any other index} \\ +1 & \text{if } i,\ j,\ k,\ \text{form an even permutation of } 1,\ 2,\ 3. \\ -1 & \text{if } i,\ j,\ k,\ \text{form an odd permutation of } 1,\ 2,\ 3 \end{cases} \tag{2.6}$$

The sequence of instructions in *Mathematica* to simulate this behavior is given by

```
LeviCivita[i_, j_, k_] := Block[{out, list, l1},
  list = {i, j, k};
  l1 = Union[list, {1, 2, 3}];
  If[Length[l1] < 3 || Length[l1] > 3,
   out = 0,
   out = Signature[list]];
  out]
```

This function makes use of the *Mathematica* functions Union[], Length[], and Signature[] to implement the properties of the Levi-Civita density. The first step of the function is the collection of the numeric indices i, j, and k in a list. Then, the condition of uniqueness is checked by using the function Union[], verifying that only the integers 1, 2, and 3 occur. Checking the length of the result allows us to distinguish two cases. First, are there two or three indices equal, and second, are there indices different from the numbers 1, 2, and 3? If this happens, the function is terminated with a return value 0. If the indices i, j, and k are contained in the set $\{1, 2, 3\}$, then the signature of the permutation is calculated. The return value is $+1$ if the permutation of 1, 2, and 3 is even and -1 if the permutation is odd.

The function LeviCivita[] allows us to verify relation (2.5) connecting the structure constants c_{ik}^{l} and the ϵ-tensor. Checking the relations with *Mathematica*, we need to define two additional functions generating the right-hand side of relation (2.5) and verifying the equality of both sides. We call these functions rhs[] and CommutativeQ[]. We also collect the three matrices $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ in a common list.

```
eList = {e1, e2, e3}
```

$$\left\{\left\{\left\{0, -\frac{I}{2}\right\}, \left\{-\frac{I}{2}, 0\right\}\right\}, \left\{\left\{0, -\frac{1}{2}\right\}, \left\{\frac{1}{2}, 0\right\}\right\}, \left\{\left\{-\frac{I}{2}, 0\right\}, \left\{0, \frac{I}{2}\right\}\right\}\right\}$$

```
rhs[i_, j_] := Sum[LeviCivita[i, j, k] eList[[k]], {k, 1, 3}]

CommutativeQ[i_, j_] := LieProduct[eList[[i]], eList[[j]]] ==
  rhs[i, j]
```

The check of the relation (2.5) between the structure constants and the ϵ-tensor is carried out by

```
Table[CommutativeQ[i, k], {i, 1, 3}, {k, 1, 3}]
```

```
{{True, True, True}, {True, True, True}, {True, True, True}}
```

The result shows that the relation is satisfied for all combinations of i and k in the range $i, k = 1, 2, 3$. This example demonstrates that the structure of a Lie algebra can be realized by matrices or tensors. On the other hand, in symmetry analysis Lie algebras are represented by differential operators. These operators are the basic elements of the vector space whose Lie product is defined by the commutator of differential operators. □

2.2.1 Representation of a Lie Algebra

In this section, we briefly discuss the representation of an algebra. Before we discuss the theoretical definition, let us continue with another example. In the previous example, we became familiar with a matrix representation of a Lie algebra. The following example shows how a Lie algebra is represented in connection with differential operators. We will find that the two different representations are isomorphic to each other.

Example 2

Another example for the representation of a Lie algebra are the three differential operators generating the Lie algebra of rotations in $\mathbb{R}^3$. This kind of algebra is connected with the symmetry of rotations. Assume we know the three operators given by

$$R_1 = x^3 \frac{\partial}{\partial x^2} - x^2 \frac{\partial}{\partial x^3},$$

$$R_2 = x^1 \frac{\partial}{\partial x^3} - x^3 \frac{\partial}{\partial x^1},$$

$$R_3 = x^2 \frac{\partial}{\partial x^1} - x^1 \frac{\partial}{\partial x^2}$$

acting in $\mathbb{R}^3$ with Cartesian coordinates x^1, x^2, and x^3. The three operators allow a unique formula if we make use of the Levi-Civita density ϵ_{ik}^{l}:

$$R_i = \sum_{k=1}^{3} \sum_{j=1}^{3} -\epsilon_{ij}^{k} \, x^j \, \partial_{x^k} \, .$$

This formula is used to define the three differential operators in *Mathematica*:

```
R[i_, f_] := Block[{variables = {x1, x2, x3}},
  Sum[Sum[-LeviCivita[i, j, k] variables[[j]] ∂variables[[k]] f, {k, 1, 3}], {j, 1, 3}]]
```

The Lie bracket in the related vector space is defined by

$$[R_i, R_j](f) = R_i(R_j f) - R_j(R_i f), \tag{2.7}$$

where f is an arbitrary function. The definition in *Mathematica* looks quite similar:

```
LieBracket[i_, j_, f_] := R[i, R[j, f]] -
  R[j, R[i, f]]
```

The right-hand side of the commutator (2.7) can be again expressed by the Levi-Civita symbol

$$[R_i, R_k] = \sum_{l=1}^{3} \epsilon_{jk}^{l} R_l \,. \tag{2.8}$$

We define the right-hand side of the commutator (2.8) as

```
rhs[i_, j_, f_] := Sum[LeviCivita[i, j, k] R[k, f], {k, 1, 3}]
```

The check of the commutation relation for arbitrary i is calculated by

```
Clear[CommutativeQ]

CommutativeQ[i_, j_, f_] := Simplify[
  LieBracket[i, j, f]] == Simplify[rhs[i, j, f]]
```

Using these definitions, we can verify that relation (2.8) holds. The use of the function CommutativeQ[] in connection with Table[] allows us to verify this proposition:

```
Table[CommutativeQ[i, j, f[x1, x2, x3]],
 {i, 1, 3}, {j, 1, 3}]
```

```
{{True, True, True}, {True, True, True}, {True, True, True}}
```

The arbitrary function f[x1, x2, x3] in CommutativeQ[] is used as an argument for the three operators R_j. These differential operators act, for example, on the infinite dimensional vector space $C^{\infty}(\mathbb{R}^3)$, thus providing an infinite dimensional representation of the Lie algebra $so(3)$ related to the special orthogonal group $SO(3)$.

The result is that the rotation operators in $\mathbb{R}^3$ possess the same structure constants as the Lie algebra of the Pauli matrices. Thus, we can state that the two representations are isomorphic. The conclusion is that two different representations of a Lie algebra may result into the same structure and especially possess the same structure constants. So we face the problem of representation of a Lie algebra. □

Generally, a representation of a Lie algebra $\mathfrak{g}$ on a vector space V is a mapping ρ from $\mathfrak{g}$ to a linear transformation of V such that

$$\rho(\alpha\,\vec{v} + \beta\,\vec{w}) = \alpha\,\rho(\vec{v}) + \beta\,\rho(\vec{w}) \tag{2.9}$$

and

$$\rho([\vec{v}, \vec{w}]) = [\rho(\vec{v}), \rho(\vec{w})], \tag{2.10}$$

where [,] is the Lie product of the algebra $\mathfrak{g}$. The dimension of the representation is equal to the dimension of V.

In Chapter 5, we shall discuss procedures to find the differential operators that represent the symmetries of differential equations. For example, the symmetries or respectively the basis of the Lie algebra for the heat equation $u_t - u_{xx} = 0$ are given by

$$\vec{v}_1 = \partial_x,\ \vec{v}_2 = \partial_t,\ \vec{v}_3 = u\,\partial_u, \tag{2.11}$$

$$\vec{v}_4 = x\,\partial_x + 2\,t\,\partial_t,\ \vec{v}_5 = 2\,t\,\partial_x - x\,u\,\partial_u, \tag{2.12}$$

$$\vec{v}_6 = 4\,t\,x\,\partial_x + 4\,t^2\,\partial_t - (x^2 + 2\,t)\,u\,\partial_u\ . \tag{2.13}$$

The basis elements of the Lie algebra are also called vector fields. This notion will become clear in Section 3.2 where we discuss tangent vectors. For the moment, let us call the elements in (2.11), (2.12), and (2.13) vector fields. These operators form a six-dimensional Lie algebra V. A convenient way to display the structure of such an assembly of operators is to write it in tabular form. For the six-dimensional Lie algebra $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_6\}$, the commutator table for V will be the 6×6 table whose *(i, j)*th entry expresses the Lie bracket $[\vec{v}_i, \vec{v}_j]$. From axiom (i), it is clear that the table is always skew-symmetric, and the diagonal elements are all zero. The structure constants can be easily read off the commutator table. The coefficient c^i_{jk} is the coefficient of $\vec{v}_i$ in the *(j, k)*th entry of the table. For the above set of operators, we find the commutator table

$[\,,\,]$	$\vec{v}_1$	$\vec{v}_2$	$\vec{v}_3$	$\vec{v}_4$	$\vec{v}_5$	$\vec{v}_6$
$\vec{v}_1$	0	0	0	$-\vec{v}_1$	$\vec{v}_3$	$-2\,\vec{v}_5$
$\vec{v}_2$	0	0	0	$-2\,\vec{v}_2$	$-2\,\vec{v}_1$	$2\,\vec{v}_3 - 4\,\vec{v}_4$
$\vec{v}_3$	0	0	0	0	0	0
$\vec{v}_4$	$\vec{v}_1$	$2\,\vec{v}_2$	0	0	$-\vec{v}_5$	$-2\,\vec{v}_6$
$\vec{v}_5$	$-\vec{v}_3$	$2\,\vec{v}_1$	0	$\vec{v}_5$	0	0
$\vec{v}_6$	$2\,\vec{v}_5$	$-2\,\vec{v}_3 + 4\,\vec{v}_4$	0	$2\,\vec{v}_6$	0	0

Table 2.1

The entry (2,6), for example, is given by

$$[\vec{v}_2, \vec{v}_6] = \vec{v}_2\,\vec{v}_6 - \vec{v}_6\,\vec{v}_2 = 2\,\vec{v}_3 - 4\,\vec{v}_4, \tag{2.14}$$

the other entries can be calculated in a similar way. The related structure constants are

$$c_{16}^5 = c_{24}^2 = c_{25}^1 = c_{46}^6 = c_{62}^3 = -2, \tag{2.15}$$

$$c_{61}^5 = c_{42}^2 = c_{52}^1 = c_{64}^6 = c_{26}^3 = 2, \tag{2.16}$$

$$c_{41}^1 = c_{45}^5 = c_{51}^3 = -1,\ c_{41}^1 = c_{55}^5 = c_{15}^3 = 1, \tag{2.17}$$

$$c_{26}^4 = -4, \text{ and } c_{62}^4 = 4 \tag{2.18}$$

with all other c_{jk}^i's being zero. □

2.2.2 Properties of Lie Algebras

This section discusses a few properties of Lie algebras useful in the classification of the solutions of differential equations. We introduce the notion of a derived algebra, the derivation of an algebra, the adjoint algebra, the Killing form of a Lie algebra, and some definitions related to the solvability of a Lie algebra.

If the commutator table or the structure constants are known, it is straightforward to calculate the so-called derived Lie algebras. These algebras are useful for classifying the Lie algebra. The Lie algebra $V^{(1)} = [V, V]$ is called the first derived algebra of the Lie algebra V. By construction, $V^{(1)}$ is an ideal. The higher-order derived algebras are recursively defined by

$$V^{(n+1)} = [V^{(n)}, V^{(n)}], \quad n = 1, 2, 3, \ldots. \tag{2.19}$$

The derived Lie algebras can be used to classify the original algebra. One of the central terms in connection with derived Lie algebras is solvability. If a Lie algebra can be classified as solvable, we know that the related differential equation can be solved. This observation of Lie is central for the solution procedures discussed in

Chapter 4. A Lie algebra V is called solvable if $V^{(n)} = 0$ for some $n > 0$. The simplest examples are the commutative Lie algebras. With this remark, all one- and two-dimensional Lie algebras are solvable. This observation will be of importance in Section 4.4.2 where we will use this criterion to integrate second-order and higher-order ordinary differential equations. A few examples will illustrate the term solvability for partial differential equations.

Example 3

The six-dimensional Lie algebra of the diffusion equation with its basis $\vec{v}_1, \ldots, \vec{v}_6$ is not solvable because the first derived Lie algebra $V^{(1)}$ contains all operators of the six-dimensional Lie algebra and, thus, cannot vanish (cf. Table 1). □

Example 4

An example of a solvable Lie algebra is given by the vector fields for the Korteweg-de Vries (KdV) equation $u_t + u\,u_x + u_{xxx} = 0$. The basis of the Lie algebra calculated in Chapter 5 reads

$$\vec{v}_1 = \partial_x,\ \vec{v}_2 = \partial_t,\ \vec{v}_3 = x\,\partial_x + 3\,t\,\partial_t - 2\,u\,\partial_u, \tag{2.20}$$

$$\vec{v}_4 = t\,\partial_x + \partial_u\ . \tag{2.21}$$

For this equation the commutator table is given by

$[\ ,\ ]$	$\vec{v}_1$	$\vec{v}_2$	$\vec{v}_3$	$\vec{v}_4$
$\vec{v}_1$	0	0	$\frac{\vec{v}_1}{2}$	0
$\vec{v}_2$	0	0	$\frac{3\,\vec{v}_2}{2}$	$-\vec{v}_1$
$\vec{v}_3$	$-\frac{\vec{v}_1}{2}$	$-\frac{3\,\vec{v}_2}{2}$	0	$\vec{v}_4$
$\vec{v}_4$	0	$\vec{v}_1$	$-\vec{v}_4$	0

Table 2.2

Examining this table, we recognize that the first derived Lie algebra contains only the operators $V^{(1)} = \{\vec{v}_1, \vec{v}_2, \vec{v}_4\}$. $V^{(1)}$ is just given by the entries in the commutator table. The second derived Lie algebra consists only of $\vec{v}_1$, i.e., $V^{(2)} = \{\vec{v}_1\}$, thus, the third step gives $V^{(3)} = \{\}$. Thus, the Lie algebra of the KdV equation is solvable. In fact, it is known that the KdV equation belongs to the equations, which are completely integrable. So far, we manually calculated the Lie algebra and its properties. The package *MathLie* offers a way to do the calculation completely automatically.

The determination of the solvability was based on a representation of the Lie algebra by vector fields. Vector fields in symmetry analysis are, on the other hand, based on infinitesimal symmetries. The infinitesimal symmetry of the KdV equation is given by

```
infKdV = {xi[1] → Function[{x, t, u}, k3 + k2 t + k4 x],
  xi[2] → Function[{x, t, u}, k1 + 3 k4 t],
  phi[1] → Function[{x, t, u}, k2 - 2 k4 u]};
```

where the constants k_i represent the group constants connected with the vector field given above. How these infinitesimal symmetries are calculated is the subject of Chapter 5. For the moment, we assume that the infinitesimal symmetries are known. This information can be used to apply the function

```
SolvableAlgebrasOfOrderN[infKdV, {u}, {x, t}, 4,
 VectorFieldRepresentation → True]
```

```
{{V[1], V[2], V[3], V[4]}}
```

to the infinitesimals. The above *MathLie* function also needs the dependent and independent variables and the number of elements in the algebra or subalgebra. The option VectorFieldRepresentation→True creates the output in the symbols of the vector fields. For the KdV equation, we find that the largest solvable Lie algebra is given by the total Lie algebra. We also can use this function to create all solvable subalgebras for the KdV equation by

```
Map[SolvableAlgebrasOfOrderN
 [infKdV, {u}, {x, t}, #
 VectorFieldRepresentation → True]&,

 {2, 3, 4}]
```

```
{{{V[1], V[3]}, {V[1], V[4]},
  {V[2], V[3]}, {V[2], V[4]}, {V[3], V[4]}},
 {{V[1], V[2], V[3]}, {V[1], V[3], V[4]}, {V[2], V[3], V[4]}},
 {{V[1], V[2], V[3], V[4]}}}
```

The result is a list containing all subalgebras of second, third, and fourth order. The function SolvableAlgebrasOfOrderN[] is extensively applied in connection with the integration of ordinary differential equations. □

Another important property of a Lie algebra useful in the study of differential equations is the derivation $\mathfrak{D}$ or its infinitesimal automorphism. A derivation $\mathfrak{D}$ of a Lie algebra V is a linear mapping of V into itself, satisfying

$$\mathfrak{D}([\vec{v}, \vec{w}]) = [\mathfrak{D}(\vec{v}), \vec{w}] + [\vec{v}, \mathfrak{D}(\vec{w})] \quad \forall\ \vec{v},\ \vec{w} \in V. \tag{2.22}$$

It is evident that for two derivations $\mathfrak{D}_1$ and $\mathfrak{D}_2$ of V, the sum $\alpha\ \mathfrak{D}_1 + \beta\ \mathfrak{D}_2$ is also a derivation. Moreover, if $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are derivations, then

$$\mathfrak{D}_1\,\mathfrak{D}_2([\vec{v},\vec{w}]) = \mathfrak{D}_1([\mathfrak{D}_2(\vec{v}),\vec{w}] + [\vec{v},\mathfrak{D}_2(\vec{w})]) \tag{2.23}$$

$$= [\mathfrak{D}_1\,\mathfrak{D}_2\,\vec{v},\vec{w}] + [\mathfrak{D}_2\,\vec{v},\mathfrak{D}_1\,\vec{w}] + [\mathfrak{D}_1\,\vec{v},\mathfrak{D}_2\,\vec{w}] + [\vec{v},\mathfrak{D}_1\,\mathfrak{D}_2\,\vec{w}]. \tag{2.24}$$

Interchanging the indices 1 and 2 and subtracting both formulas from each other, we get

$$[\mathfrak{D}_1,\mathfrak{D}_2]([\vec{v},\vec{w}]) = [[\mathfrak{D}_1,\mathfrak{D}_2]\vec{v},\vec{w}] + [\vec{v},[\mathfrak{D}_1,\mathfrak{D}_2]\vec{w}], \tag{2.25}$$

meaning that the commutator of two derivations is again a derivation. Let V now be a Lie algebra over the real numbers $\mathbb{R}^1$ or the complex numbers $\mathbb{C}$. Using the above general definitions, we introduce an operation for the classification of Lie algebras.

Consider the linear map $\mathrm{ad}\,\vec{v}$ of V into itself defined by

$$\mathrm{ad}\,\vec{v}\,(\vec{w}) := [\vec{v},\vec{w}] \quad \text{with } \vec{v},\ \vec{w} \in V. \tag{2.26}$$

Using the Jacobi identity (iii) in connection with the definition of the derivation, we can write

$$\mathrm{ad}\,\vec{v}([\vec{w},\vec{u}]) = [\mathrm{ad}\,\vec{v}(\vec{w}),\vec{u}] + [\vec{w},\mathrm{ad}\,\vec{v}(\vec{u})]; \tag{2.27}$$

i.e., the map $\mathrm{ad}\,\vec{v}$ represents a derivation of V. Furthermore, from the Jacobi identity and the definition of $\mathrm{ad}\,\vec{v}$, we obtain

$$\mathrm{ad}[\vec{v},\vec{w}]\,(\vec{u}) = [\mathrm{ad}\,\vec{v},\mathrm{ad}\,\vec{w}]\,(\vec{u}). \tag{2.28}$$

Hence, the set $V_a = \{\mathrm{ad}\,\vec{v} \mid \vec{v} \in V\}$ is a linear Lie algebra and a subset of the Lie algebra of all derivations and is called the adjoint algebra. The map $\Phi : \vec{v} \to \mathrm{ad}\,\vec{v}$ is the homomorphism of V onto V_a. In addition, the kernel of the homomorphism Φ is the center of V.

The representation of $\mathrm{ad}\,\vec{v}$, called the adjoint representation of the Lie algebra, always provides a matrix representation of the algebra. If $\{\vec{v}_i\}$ is a n-dimensional basis for V, then

$$\mathrm{ad}\,\vec{v}_i(\mathrm{ad}\,\vec{v}_j) = \sum_{k=1}^{n} c_{ij}^{k}\,\vec{v}_k\,. \tag{2.29}$$

Therefore, the matrix A associated with the transformation $\mathrm{ad}\,\vec{v}_i$ is given by the structure constants

$$(A_i)_k^j = c_{ik}^j\,. \tag{2.30}$$

where $(A_i)_k^j$ represents the (j,k)th entry for the ith matrix. Note the transposition of the indices j and k. So if we know the structure constants of a Lie algebra, we also know the matrix representation of the adjoint Lie algebra.

Example 5

As an example, let us examine the rotations about the three coordinate axes X, Y, and Z. The group of the rotation can be represented by the three matrices

```
Rx[α_] := {{1, 0, 0},
  {0, Cos[α], -Sin[α]},
  {0, Sin[α], Cos[α]}}

Ry[β_] := {{Cos[β], 0, Sin[β]},
  {0, 1, 0}, {-Sin[β], 0, Cos[β]}}

Rz[γ_] := {{Cos[γ], -Sin[γ], 0},
  {Sin[γ], Cos[γ], 0}, {0, 0, 1}}
```

A representation of the corresponding Lie algebra follows if we calculate the first coefficient of a Taylor expansion around the identity, meaning that the representation of the Lie algebra is given by the first derivatives with respect to the parameter around the identical rotation.

```
e1 = ∂α Rx[α] /. α → 0; MatrixForm[e1]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

```
e2 = ∂β Ry[β] /. β → 0; MatrixForm[e2]
```

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

and

```
e3 = ∂γ Rz[γ] /. γ → 0; MatrixForm[e3]
```

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

These three matrices also build the basis for the Lie algebra *so*(3). We will see in Chapter 4 that the matrices $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ are the infinitesimal generators of the Lie group. At the other hand, we know from our examinations above that the structure constants of *so*(3) are given by the Levi-Civita tensor. Knowing the structure

constants, we also know a representation of the adjoint Lie algebra. The structure constants of $so(3)$ are $c^k_{ij} = \epsilon^k_{ij}$. Applying relation (2.30), we can represent the adjoint Lie algebra by

$$(A_i)^j_k = c^j_{ik} = \epsilon^j_{ik} = -\epsilon^k_{ij} . \tag{2.31}$$

So the matrices

```
ae1 = MatrixForm[Array[-LeviCivita[1, #1, #2]&,
    {3, 3}]]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

```
ae2 = MatrixForm[Array[-LeviCivita[2, #1, #2]&,
    {3, 3}]]
```

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

and

```
ae3 = MatrixForm[Array[-LeviCivita[3, #1, #2]&,
    {3, 3}]]
```

$$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

are the adjoint representation of the Lie algebra $so(3)$. In fact, these matrices are identical with the original matrices of $so(3)$. In the above input lines, we used as argument of Array[] a pure function. A pure function in *Mathematica* is terminated by & and allows so-called slots (#1, #2) as input channels. In the above lines, the function LeviCivita[] with two slots and one fixed argument was used as a pure function. □

With the definition of the adjoint algebra, we introduced the homomorphism $\vec{v} \to \mathrm{ad}\,\vec{v}$. In terms of coordinates, we have

$$\{\mathrm{ad}\,\vec{v}\,(\mathrm{ad}\,\vec{w})\}^i = [\vec{v}, \vec{w}]^i = \mathrm{ad}\,\vec{v}_i\,(\mathrm{ad}\,\vec{v}_j) = \sum_{k,l=1}^{n} c^i_{lk}\, v^l\, w^k , \tag{2.32}$$

i.e.,

$$(\mathrm{ad}\,\vec{v})^i_k = \mathrm{ad}\,\vec{v}_i\,(\mathrm{ad}\,\vec{v}_j) = \sum_{l=1}^{n} c^i_{lk}\, v^l . \tag{2.33}$$

Using these relations, we are able to define a scalar product in a Lie algebra by the following relation,

$$\langle \vec{v},\ \vec{w} \rangle \ = \ \mathrm{Tr}(\mathrm{ad}\,\vec{v}\,\mathrm{ad}\,\vec{w})\,. \tag{2.34}$$

This product satisfies the following properties:

(i) Symmetry

$$\langle \vec{v},\, \vec{w} \rangle \ = \ \langle \vec{w},\, \vec{v} \rangle \tag{2.35}$$

(ii) Bilinearity

$$\langle \alpha\,\vec{v} \ + \ \beta\vec{w},\, \vec{u}\,\rangle \ = \ \alpha\,\langle \vec{v},\, \vec{u}\rangle \ + \ \beta\,\langle \vec{w},\, \vec{u}\rangle \tag{2.36}$$

for all $\vec{v},\ \vec{w}\,,\ \vec{u}\ \in\ V$ and $\alpha, \beta \in \mathbb{R}$ or $\mathbb{C}$. And, the relation

(iii)

$$\langle \mathrm{ad}\,\vec{v}(\vec{w}),\, \vec{u}\rangle \ + \ \langle \vec{w},\, \mathrm{ad}\,\vec{v}(\vec{u})\rangle \ = \ 0 \tag{2.37}$$

or

$$\langle [\vec{v},\, \vec{w}],\, \vec{u}\rangle \ + \ \langle \vec{w},\, [\vec{v},\, \vec{u}]\rangle \ = \ 0. \tag{2.38}$$

These properties are immediately derived from the properties of the trace.

The symmetric bilinear form $\langle \vec{v},\ \vec{w}\rangle$ on $V \times V$ is called the Killing form of the Lie algebra. In terms of the coordinates, this expression is given by

$$\langle \vec{v},\ \vec{w}\rangle \ = \ \mathrm{Tr}\big((\mathrm{ad}\,\vec{v})^{i}_{k}\,(\mathrm{ad}\,\vec{w})^{l}_{i}\big) \ = \ \sum_{l,k=1}^{n} c^{i}_{lk}\ v^{l}\ c^{k}_{mi}\ w^{m}$$

$$= \ \sum_{l,m=1}^{n} g_{lm}\ v^{l}\ w^{m}, \tag{2.39}$$

where the symmetric second-rank tensor

$$g_{lm} \ = \ \sum_{i,k=1}^{n} c^{i}_{lk}\ c^{k}_{mi} \tag{2.40}$$

is called the Cartan metric tensor of the Lie algebra V. Note that for some algebras the Killing form can be degenerate. Especially for commutative algebras we find degeneration; i.e., $\det(g_{ik}) \ = \ 0$.

The Killing form and the Cartan metric tensor play a fundamental role in the theory of Lie algebras and their representations. For example, a simple criterion for the solvability of a Lie algebra in terms of the Killing form is: if $\langle \vec{v},\, \vec{v}\rangle = 0$ for each

$\vec{v} \in V$, then V is called a solvable Lie algebra, or if an algebra V is nilpotent, then $\langle \vec{v}, \vec{v} \rangle = 0$ for all $\vec{v} \in V$.

We have separated the type of solvable and nilpotent algebras from the set of all Lie algebras by the above criteria. However, we do not know much about the terms simple and semisimple Lie algebras. In the following, we define the class of simple and semisimple Lie algebras, which are important in the study of the structure and classification of Lie algebras

Definition: Semisimple Lie algebra

A Lie algebra V is semisimple if it has a non-zero commutative ideal. ○

The criterion for semisimplicity is given by the following theorem:

Theorem: Cartan's theorem

A Lie algebra V is semisimple if and only if its Killing form is non-degenerate. ○

This theorem of Cartan is useful for classifying the algebras obtained in the symmetry analysis as semisimple or not. A simple Lie algebra is defined as follows:

Definition: Simple Lie algebra

A Lie algebra V is simple if it has no ideals other than $\{0\}$ and V and if $V^{(1)} = [V, V] \neq 0$. ○

The discussed terms are useful for expressing some relations of differential equations in the following sections. All are the basis for a theory which is general in its settings and can be used in different applications of physics and mathematics.

3

Derivatives

The symmetry analysis of differential equations is based on several differential operators. Among these operators are the ordinary differentiation, the total differential, the Fréchet derivative, the Euler-Lagrange derivative, and the prolongation, to name the main operators. The basis of the symmetry analysis is the prolongation of a differential equation. Unfortunately, the prolongation as a differential operator is not implemented in *Mathematica*. This chapter will discuss the different types of derivatives used in the calculus of symmetry analysis and will demonstrate their application by several examples. Another subject of the present chapter is the presentation of the theoretical background for the derivatives. One point we will discuss is the connection of the theory with the practical implementation of these operators in *Mathematica*. Application of the defined operators to several examples will demonstrate their use. Throughout the text, we use subscripts to denote a differentiation. The subscripted representation in *Mathematica* is created by the function LieTraditionalForm[]. This *MathLie* function converts the standard form of differentials in *Mathematica* to a traditional form frequently used in mathematics.

3.1. Ordinary and Partial Derivatives

Ordinary and partial derivatives are widely used in calculus. As a matter of fact, this kind of calculation is also applied in the symmetry analysis of differential equations. Gauss, Leibniz, and Newton introduced the notion of derivatives in the 17th century in order to have a measure for the slope of a function. Still today, we continue to use

derivatives to measure the slope of a function at a position x. The definition of a differential is one of the fruitful concepts mankind invented to describe nature in mathematical terms. Newton and Leibniz introduced the calculus of differentials to describe physical and mathematical relations by means of differential equations. The main ingredients of differential equations are derivatives combined in a linear or non-linear way. The definition of a derivative in terms of a limiting process is given by the following:

Definition: Ordinary derivative

Given a smooth function $f: \mathbb{R} \to \mathbb{R}$, the derivative of f is defined by the relation

$$\frac{df}{dx} := \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \quad \circ \tag{3.1}$$

This definition is the mathematical expression of how to manage the calculation of the slope for a known function f. The meaning of this formula is that we have to take two neighboring points separated by a distance h in the x domain and calculate the ratio of the difference of the function at these points. If we assume that the distance h becomes smaller and smaller, we end up with a value describing the slope of the function at the point x. Here we used the representation of the derivative in mathematical terms. The definition of the derivative given in (3.1) is not only a symbolic formula but also of practical relevance. In *Mathematica,* we can demonstrate the practical use by just applying relation (3.1) to a specific function. Let us assume that the function f is given by the trigonometric function

```
f[x_] := Sin[x]
```

Formula (3.1) in terms of *Mathematica* reads

```
Df = Limit[(f[x + h] - f[x])/h, h → 0]
```

```
Cos[x]
```

which provides us with the expected result. We certainly know that the derivative of the sin is given by a cos. The result can be checked by a symbolic calculation using the differentiation

```
∂x f[x]
```

```
Cos[x]
```

which gives the same result. We realize that *Mathematica* provides the same result by different algorithmic procedures. However, the standard way of calculating derivatives of functions f is the application of the operator ∂_x to $f[x]$. The pattern

$\partial_x f[x]$ or $D[F[x], x]$ serves to calculate all the ordinary differentials of a function f with one independent variable.

Another way of looking on relation (3.1) in the definition above is based on a geometrical interpretation. Rewriting the original formula (3.1) helps us to understand the geometrical contents. Let us first replace the limit in equation (3.1) by another representation. The derivative defined on the left-hand side of equation (3.1) can be represented by introducing a condition on the right-hand side. Dropping for the moment the Limit[] and introducing a reference point x_0 on the x-axis, we are able to rewrite the right-hand side. We assume that x_0 is a distance h away from our point of interest x. The resulting value on the right-hand side of (3.1) is an approximation of the derivative at the point x_0. In *Mathematica*, we write

```
Clear[f]
```

```
Df = (f[x + h] - f[x])/h /. h → x0 - x
```

$$\frac{-f[x] + f[x0]}{-x + x0}$$

The left-hand side in equation (3.1) can be represented by the differential operator ∂_x. The calculation is carried out at the location $x = x_0$. This representation of the derivative gives us

```
Dlf = ∂x f[x] /. x → x0
```

```
f'[x0]
```

Combining the two expressions, we get an approximate representation of a derivative for the function f at the location x_0 by

```
df = Dlf == Df
```

$$f'[x0] == \frac{-f[x] + f[x0]}{-x + x0}$$

The geometrical way of reading this equation is to consider df as a parametric definition of the function $f[x]$. The parameter x_0 denotes a specific location in the domain of the independent variables. An explicit representation of the function f follows from df by solving it with respect to $f[x]$:

```
sol1 = Solve[df, f[x]] /. f[x] → w
```

```
{{w → f[x0] + x f'[x0] - x0 f'[x0]}}
```

The replacement of $f[x]$ by an auxiliary variable w is necessary to define the function $f[x]$ in a pure function as

```
fun = f → Function[{x, x0}, w] /. Flatten[sol1]
```

```
f → Function[{x, x0}, f[x0] + x f'[x0] - x0 f'[x0]]
```

The result is a representation of the function $f[x]$ defined at any location x knowing at the same time the same function f at a point x_0. In addition, we also have to know the derivative of f at this location. The relation appears somewhat strange at the first look. However, the geometrical content of this expression is easy to understand if we represent it graphically. To perceive the implications of the relation, let us examine a plot of f. Being specific in the plotting, we set the function f to the trigonometric function sin

```
f[x_] := Sin[x]
```

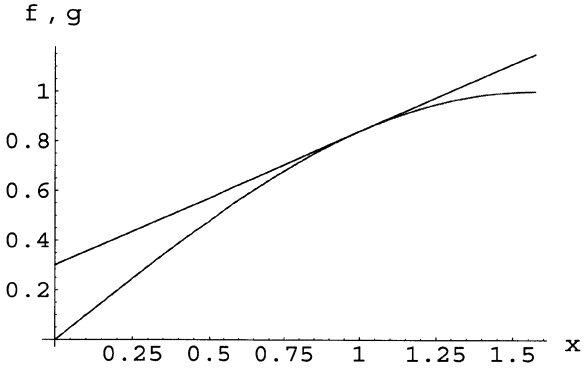
In another step, we define a function $g[x, x_0]$, combining our results for $f[x, x_0]$. The new function $g[x, x_0]$ allows us to represent f at any points x and x_0:

```
Clear[g]
```

```
g[x_, x0_] := f[x, x0] /. fun
```

If we plot both functions $f[x]$ and $g[x, x_0]$ in a common coordinate frame, we get the following picture:

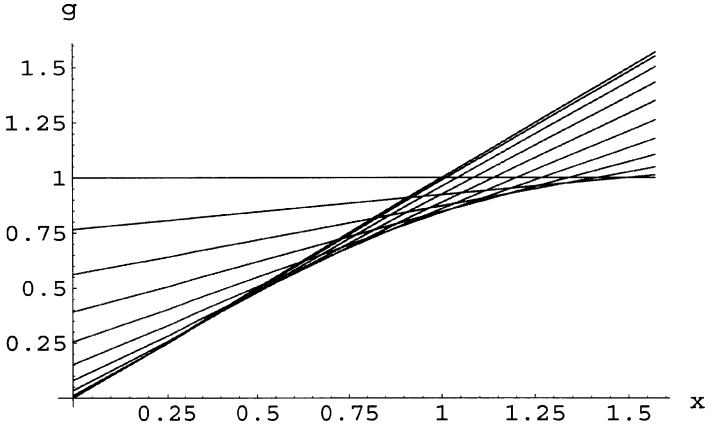
```
Plot[
 Evaluate[{f[x], g[x, 1]}], {x, 0, π/2},
 AxesLabel → {"x", "f,g"},
 PlotStyle → {RGBColor[1, 0, 0],
   RGBColor[0, 0, 1]}]
```



From the above figure, we clearly see the geometrical meaning of g and f. In fact, g is the representation of the tangent of the function f at a certain location x_0. In the figure

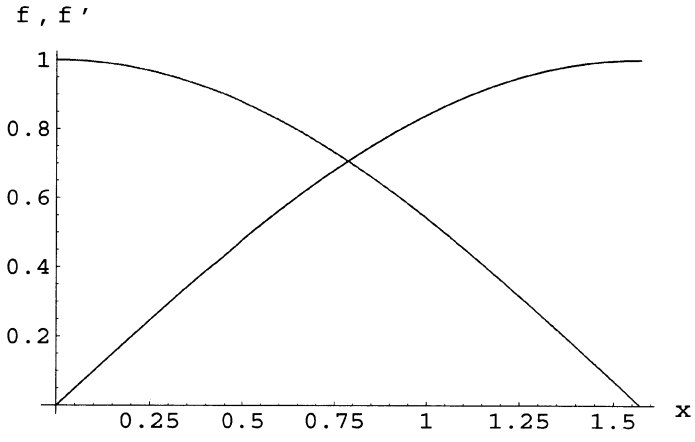
above, we chose $x_0 = 1$. The figure allows us to interpret the derivative of f as a slope. The relation derived in *df* clearly displays a linear dependence in the independent variable x. Examining the slopes of the function $f[x]$ at other points x_0, we get a set of tangents. A graphical representation of this set is given below. This sort of plot represents an envelope of the function $f[x]$. The following lines of *Mathematica* are necessary to create the envelope of f:

```
Plot[Evaluate[
  Table[g[x, x0], {x0, 0, π/2, π/20}]],
 {x, 0, π/2}, AxesLabel → {"x", "g"},
 PlotStyle → Table[RGBColor[1, 0, 0],
   {x0, 0, π/2, π/20}]]
```



The above figure shows that the slope of the function sin[x] starts with a finite value at $x_0 = 0$ and ends up with a vanishing value at $x_0 = \pi/2$. The following figure, showing the function and the derivative of the function, represents another way to examine the behavior of the slope.

```
Plot[Evaluate[
  {f[x], ∂x f[x]}], {x, 0, π/2},
 AxesLabel → {"x", "f,f'"},
 PlotStyle → {RGBColor[0, 0, 1],
   RGBColor[1, 0, 0]}]
```



The figure shows us the derived behavior of the slope in a more compact way. The slope of sin[x] starts with the value 1 at $x = 0$ and finishes with 0 at $x = \pi/2$.

Knowing the geometrical meaning of differentiation, we can ask for additional properties of this operation. In the following, we will discuss some of these properties. We only state a few of these features known by *Mathematica*. One of these properties is the product rule which governs the differentiation of a product of functions, e.g., f and g. The product rule is implemented in *Mathematica* and automatically applied to products of functions:

```
Clear[f, g]
```

```
prule = ∂x (f[x] g[x]) // LieTraditionalForm
```

$$g\, f_x + f\, g_x$$

The result represents the expected relation which is known from standard texts in calculus. Another feature of derivatives is the rule for rational functions. The differentiation of the ratio *f[x]/g[x]* gives

```
qrule = Simplify[∂x f[x]/g[x]] // LieTraditionalForm
```

$$\frac{g\, f_x - f\, g_x}{g^2}$$

which is the standard formula. The chain rule of Leibniz is useful in differentiating nested functions

```
crule = ∂x f[g[x]] // LieTraditionalForm
```

$$f_g\, g_x$$

which indicates that we first differentiate the function f with respect to g followed by a differentiation of g with respect to x. The properties stated above and more are known by *Mathematica* to manage the calculation of differentials.

In symmetry analysis, we frequently have to deal with functions depending on several independent variables. A function of a set of variables can be differentiated with respect to one of these variables at a time. The rest of the independent variables will stay unchanged in this calculation. The slope of a function of several variables is not just a single function, since the independent variables may vary in different ways. All the rates of change for a function of m variables are described by m functions, called its partial derivatives. In the discussion above, we introduced the definition of the derivative known as an ordinary derivative which is defined for functions depending on a single independent variable. The more generic case is that we have functions depending on several independent variables. The partial derivatives of a function of several variables are its ordinary derivatives with respect to each variable separately. We can define this as follows:

Definition: Partial derivative

Given a smooth function $f: \mathbb{R}^m \to \mathbb{R}$ depending on m independent variables x_m, we define the partial derivative of f with respect to the independent variable x_q by

$$\frac{\partial f}{\partial x_q} = \lim_{h \to 0} \frac{(f(x_1, x_2, \ldots, x_q + h, \ldots, x_m) - f(x_1, x_2, \ldots, x_q, \ldots, x_m))}{h} . \bigcirc \tag{3.2}$$

This formula allows us to calculate the variation of f with respect to different coordinates x_q. The partial derivative of a function is an operation known by *Mathematica*. The partial derivative is accessible under the same pattern $\partial_{\square}\, \square$. Although we can access partial derivatives and ordinary derivatives by the same symbol D[], *Mathematica* is capable of distinguishing the different operations. Consider, for example, a function $f = f(x_1, x_2)$ of two variables. If we treat x_2 as a constant, f may be differentiated with respect to x_1. The result is called a partial derivative of f with respect to x_1. In *Mathematica*, we carry out this by

```
∂x1 f[x1, x2] // LieTraditionalForm
```

$$f_{x_1}$$

The resulting symbol for the representation of a partial derivative in *Mathematica* is a superscripted expression of the function f. The superscripts denote the order of differentiation with respect to the independent variables. In our example, we get the first derivative with respect to x_1. The derivative with respect to x_2 follows in the same way by

```
∂x2 f[x1, x2] // LieTraditionalForm
```

$$f_{x_2}$$

The combination of both operations allows us to calculate higher-order derivatives

```
∂x1,x2 f[x1, x2] // LieTraditionalForm
```

$$f_{x_1,x_2}$$

Higher-order derivatives follow by carrying out the differentiation with respect to different variables. The calculation of higher derivatives is done for functions with only one independent variable in a similar way. Since both operations are nearly identical, here we will give only the definition for the case with more than one independent variable. The one-dimensional case is included in this definition. The kth-order derivative is defined as follows:

Definition: kth-Order derivative

Given a smooth function $f: \mathbb{R}^m \to \mathbb{R}$ depending on m independent variables x_q, we call

$$\frac{\partial^k f(x)}{\partial x_{j_1} \partial x_{j_2} \ldots \partial x_{j_k}} := \partial_J f(x), \tag{3.3}$$

the kth-order partial derivative of f with respect to the m independent variables x. The non-sorted multi-index $J = (j_1, j_2, \ldots, j_k)$ denotes the derivative with respect to one of the m coordinates. The integers $1 \le j_k \le m$ of this k-tuple indicate which derivatives are being taken. The order of differentiation k is equivalent to the sum of all indices j_k, which we denote by $|J| = \sum_{i=1}^{k} j_i$. ○

Using this definition, we are able to calculate, for example, the second-order derivative of the function f. The calculation of the partial derivative in *Mathematica* is as simple as the application of the ordinary derivative even for higher-order derivatives. For example, the sixth-order derivative of f is derived by

```
∂{x1,2},{x2,4} f[x1, x2] // LieTraditionalForm
```

$$f_{x1,x1,x2,x2,x2,x2}$$

So far, we discussed simple examples of derivatives already implemented in *Mathematica*. The following sections will illustrate how special types of derivatives are implemented. We will discuss tangent vectors, vector fields, Fréchet derivatives, prolongations of vector fields, and variational derivatives also known as Euler derivatives. The special types of derivatives we are going to discuss are useful in examining symmetries of differential equations.

3.2. Tangent Vector

Sometimes it is important to know how a real-valued function $f: \mathbb{R}^n \to \mathbb{R}$ varies in different directions. The partial derivative discussed above only measures how much f changes in a certain direction. However, it is also possible to measure the variation of f in other directions. Measuring the variation of f at a location $x \in \mathbb{R}^m$ along a straight line $t = x + tx_0$, where $x_0 \in \mathbb{R}^m$, we need the tangent vector $\vec{v}_x$. Since we are dealing with differential operators, we define the tangent vector as an operator acting in the space of functions. Actually, a tangent vector is a vector with a cetrain direction and a finite length. However, in view of the application in symmetry analysis, let us define such an operator.

Definition: Tangent vector

We assume that $f: \mathbb{R}^m \to \mathbb{R}$ is a smooth differentiable function. The tangent vector $\vec{v}_x$ is defined by the relation

$$\vec{v}_x(f) = \frac{d}{dt}(f(\vec{x} + t\,\vec{x}_0)|_{t=0}, \tag{3.4}$$

where $x_0 \in \mathbb{R}^n$ and t is a real parameter. ○

This definition is known as directional derivative in calculus. A more explicit way to write the definition is given by

$$\vec{v}_x(\mathrm{f}) = \lim_{t\to 0} \frac{f(\vec{x} + t\,\vec{x}_0) - f(\vec{x})}{t}. \tag{3.5}$$

Relation (3.5) is more convenient in comparison with the definition of an ordinary derivative. On the other hand, equation (3.4) is more useful in the implementation of the tangent vector in *Mathematica*. Although the second definition (3.5) is based on a complicated mathematical process involving the determination of a limit, the first expression is easier to handle symbolically. The reason is that equation (3.4) contains basic operations like an ordinary differentiation and a substitution. Both of these operations are easily handled by *Mathematica*. Since *Mathematica* does not know how to calculate the tangent vector of a function, we must define an operator which

handles this kind of calculation. Let us now examine equation (3.4) in more detail to see how an implementation can be based on it. In the calculation of the tangent vector for an arbitrary function f, we need to know the function f itself, the independent variables $\vec{x}$, and the support point $\vec{x}_0$. We use these three components as input parameters for our function TangentVector[]. We define the function TangentVector[] in the following way:

```
TangentVector[f_, x_List, x0_List] :=
 Block[{rule, res, t},
  rule = Thread[x → x + t x0];
  res = f /. rule;
  res = ∂t res /. t → 0]
```

These few lines closely follow the definition given in equation (3.4). The lines just state that the original argument is replaced by a new argument and that after the replacement, a differentiation with respect to the parameter t takes place. At the end of the calculation t is replaced by zero. The actual calculation is reduced to an ordinary differentiation with respect to a parameter. All other operations are replacements given as a transformation of the argument and as a side condition. The definition given in *Mathematica* is capable of reproducing the general formula in (3.4) at a certain point $\vec{x}_0$. As an example, we demonstrate here the calculation for a function f depending on four independent variables:

```
TangentVector[f[x1, x2, x3, x4],
   {x1, x2, x3, x4}, {x10, x20, x30, x40}] //
 LieTraditionalForm
```

$$x10\, f_{x1} + x20\, f_{x2} + x30\, f_{x3} + x40\, f_{x4}$$

As expected, the result of our calculation is a sum of four products. Each product consists of a partial derivative with respect to the coordinate x_q and the component x_{0_q} of the related location. Similar to the ordinary differentiation, the function TangentVector[] satisfies some additional properties. Some of these algebraic features of the directional derivative are listed below. Let us assume that we have two real numbers a and b and two independent functions f and g. Then, we can show that the relation

```
TangentVector[a f[x1, x2] + b g[x1, x2], {x1, x2}, {x10, x20}] ==
 a TangentVector[f[x1, x2], {x1, x2}, {x10, x20}] +
  b TangentVector[g[x1, x2], {x1, x2}, {x10, x20}]
```

```
True
```

is satisfied. This behavior of the tangent vector is known as linearity. Thus, we can say that TangentVector[] is a linear operator. The application of TangentVector[] on a product gives us

```
TangentVector[f[x1, x2] g[x1, x2],
 {x1, x2}, {x10, x20}]
```

```
g[x1, x2] (x20 f^(0,1)[x1, x2] + x10 f^(1,0)[x1, x2]) +
 f[x1, x2] (x20 g^(0,1)[x1, x2] + x10 g^(1,0)[x1, x2])
```

which is just the scalar product of the vector *(f, g)* with a vector containing the two tangent vectors of f and g as elements. There is also a chain rule for the tangent vector similar to the case of ordinary differentiation. For example, let g_1 and g_2 be two differentiable functions depending on x_1 and x_2. For a function F given by

```
f = F[g1[x1, x2], g2[x1, x2]]
```

```
F[g1[x1, x2], g2[x1, x2]]
```

we can derive the tangent vector in the form

```
TangentVector[f, {x1, x2}, {x10, x20}] // LieTraditionalForm
```

$$F_{g1}\ (x10\ g1_{x1} + x20\ g1_{x2}) + F_{g2}\ (x10\ g2_{x1} + x20\ g2_{x2})$$

which is a superposition of the vector field of g_1 and g_2 multiplied by the derivatives of F. As we demonstrated, all these properties are immediately available without any additional definitions. This behavior is actually based on the implementation of the derivative in *Mathematica*.

The name used for our function to calculate the tangent vector of a given function is somewhat misleading. Actually, we calculate a scalar product of the tangent vector and a support vector $\vec{x}_0$ using our function. In some calculations, however, it is necessary to have the vector components of the tangent vector available. Such an application, for example, is the calculation of the tangent surface on a hypersurface.

The components of the tangent vector $\vec{v}_x$ become available by altering TangentVector[] in an appropriate way. The following lines generalize the function in such a way that the result of the calculation is a vector of differentials applied to a function. We also assume in our definition that the support point is arbitrary and thus can be created by the operator TangentVector[] itself:

```
TangentVector[f_, x_List] := Block[{rule, res, t},
  x0 = Table[Unique["$aU"],
    {i, 1, Length[x]}];
  rule = Thread[x → x + t x0];
  res = f /. rule;
  res = ∂t res /. t → 0;
  Table[Coefficient[res, x0[[i]]],
   {i, 1, Length[x0]}]]
```

The application of the function TangentVector[] on a function f depending on three independent variables gives us

```
Clear[f];
TangentVector[f[x1, x2, x3], {x1, x2, x3}] // LieTraditionalForm
```

$$\{f_{x1}, f_{x2}, f_{x3}\}$$

which is, in fact, the gradient of the scalar function f. We note that the function TangentVector[] needs only two arguments, the function f and a list of independent variables.

For some applications in geometry and physics, we need to calculate the tangent surface of a given function. Recalling the definition of the tangent of a function given at the beginning of this section, we generalize this one-dimensional definition to a two-dimensional version. Using the vector representation of the tangent vector at a certain point $\vec{x}$ for the two-dimensional case, we can represent the tangent surface by

$$f_s = f(x_0, y_0) + (\vec{x} - \vec{x}_0) \cdot \vec{v}_x(f), \tag{3.6}$$

representing the sum of the function at the support point (x_0, y_0) and the scalar product of $(\vec{x} - \vec{x}_0)$ and the tangent vector. Similar to the definition of a tangent vector, we can implement a tangent surface by

```
Clear[TangentSurface];
TangentSurface[f_, x_List, x0_List] :=
 Block[{rule, tvector, sf, surface},
  rule = Thread[x → x0];
  tvector = TangentVector[f, x];
  sf = f /. rule;
  surface = sf + (x - x0) . tvector]
```

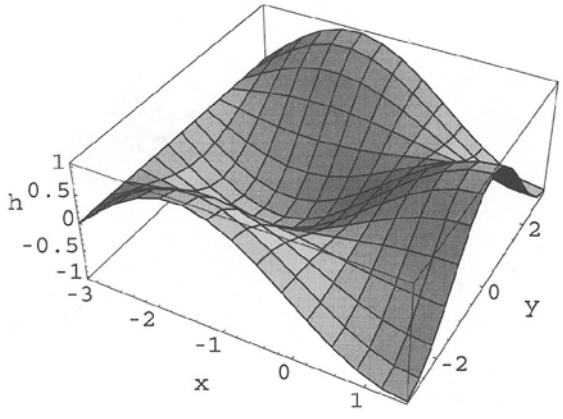
Using the function TangentSurface[], we can determine the tangent space located at x_0 of a given function f. As an example, let us consider a function h in a two-dimensional space with coordinates x and y

```
h = Sin[x] Cos[y]
```

```
Cos[y] Sin[x]
```

The function $h[x, y]$ has the graphical representation

```
pl1 =
 Plot3D[h, {x, -π, π/2}, {y, -π, π}, AxesLabel → {"x", "y", "h"}]
```



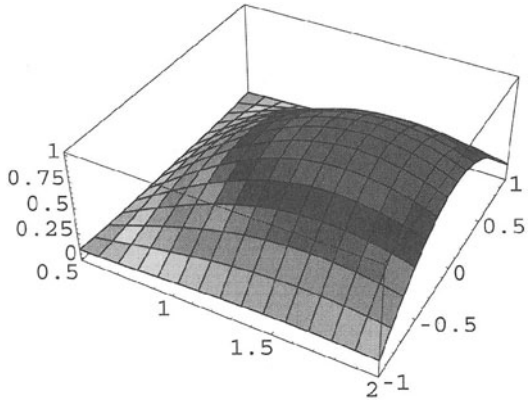
We calculate the tangent surface of this function at the point $(x_0, y_0) = (\pi/2, 0)$ by using our function

```
ht = TangentSurface[h, {x, y}, {π/2, 0}]
```

```
1 + (-π/2 + x) Cos[x] Cos[y] - y Sin[x] Sin[y]
```

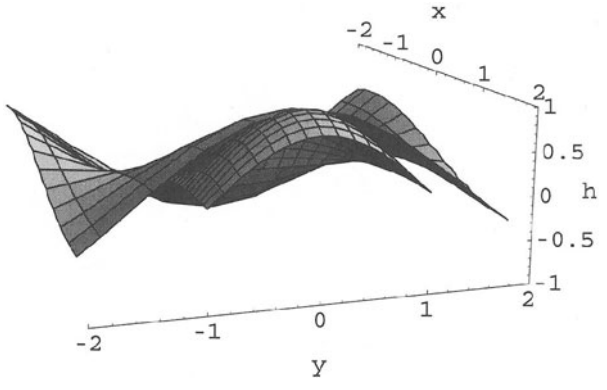
A graphical representation of this relation for the tangent surface follows with

```
pl2 = Plot3D[ht, {x, .5, 2}, {y, -1, 1},
  ColorFunction → Hue]
```



Superimposing both surfaces demonstrates that the two functions have the common point $(x_0, y_0)=(\pi/2, 0)$.

```
Show[pl1, pl2, PlotRange → {{-2, 2}, {-2, 2}, {-1, 1}},
 ViewPoint → {3.130, -1.044, 0.751}, Boxed → False]
```



In the above figure, we observe that the tangent surface is located below the surface h. The tangent surface is plotted for a smaller interval in the x and y directions to prevent intersections of h and the tangent surface. This example shows that the one-dimensional notion of a tangent can be generalized to a two-dimensional version. This generalization is not restricted to two dimensions but can be extended to higher dimensions. Since the higher-dimensional cases cannot be represented easily by graphics, we suppress a further discussion of these tangent surfaces.

This section was intended to show how a *Mathematica* function for a derivative can be defined if we know an appropriate mathematical definition. We also notice that not every mathematical definition is an efficient definition for an implementation in *Mathematica*. An essential point to efficiently implement a mathematical relation in *Mathematica* is a mathematical definition based on structures which are basic elements in *Mathematica*. In the case of the TangentVector[] function, it was essential that we used the pattern matching of *Mathematica* in the replacement rules. The application of such simple operations allows us to write refined functions. In the next section, we will come back to a derivative already known by *Mathematica*, the total derivative.

3.3. The Total Derivative

Let us consider functions f depending on a set of independent variables $x = (x_1, x_2, \ldots, x_n)$ and a set of dependent variables $u_{(k)}$, $k = 0, 1, 2, \ldots$, where $u_{(k)}$ represents all possible derivatives of $u = (u^1, u^2, \ldots, u^\alpha)$ with respect to the independent variables x. We are interested in the derivative of these functions with respect to all independent variables. If we assume that u depends on the vector x, we must consider all derivatives of f with respect to x and $u_{(k)}$. In other words, we obtain the total derivative of f by differentiating f with respect to x, while treating all the u^α's and their derivatives as functions of x.

Definition: Total derivative

The total derivative of a function $f(x, u_{(k)})$ with respect to the independent variable x_i is given by

$$D_i f = \frac{\partial f}{\partial x_i} + \sum_{\alpha=1}^{q} \sum_{J} u^{\alpha}_{J,i} \frac{\partial f}{\partial u^{\alpha}_{J}} \tag{3.7}$$

where for $J = (j_1, j_2, \ldots, j_l)$,

$$u^{\alpha}_{J,j} = \frac{\partial u^{\alpha}_{J}}{\partial x_i} = \frac{\partial^{l+1} u^{\alpha}}{\partial x_i \partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_l}}, \tag{3.8}$$

and the sum in (3.7) runs over all J's of order $0 \leq |J| \leq k$ with $|J| = \sum_{i=1}^{l} j_i$. k is the highest order of the derivatives occurring in f. ○

The *Mathematica* analogue of this definition is available by the function D[], which S. Wolfram [1991] calls a partial derivative. Showing the equivalence of both notions, let us demonstrate the action of the function D[] by considering a simple example.

Let us examine a function f given by $f = x u u_{xy}$, where $u = u(x, y)$ is a function of x and y. We apply the function D[] on this expression in two steps. First, we use x as the variable of differentiation, and in a second step, we differentiate with respect to y:

```
f = x u[x, y] ∂x,y u[x, y]; f // LieTraditionalForm
```

$$u\, x\, u_{x,y}$$

```
D[f, x] // LieTraditionalForm
```

$$u\, u_{x,y} + x\, u_x\, u_{x,y} + u\, x\, u_{x,x,y}$$

```
D[f, y] // LieTraditionalForm
```

$$x\, u_y\, u_{x,y} + u\, x\, u_{x,y,y}$$

Comparing the results obtained by *Mathematica* with the definition given above demonstrates the equivalence of both notions. Higher-order total derivatives are defined by a repeated application of the single operator (3.7) with different variables of differentiation. If $J = (j_1, j_2, \ldots, j_m)$ is a mth-order multi-index, with $1 \leq j_m \leq p$ for each m, then the jth total derivative is denoted by

$$D_J = D_{j_1} D_{j_2} \cdots D_{j_m} . \tag{3.9}$$

For example, we find the $D_x\, D_y$ and $D_y\, D_x$ derivatives by successively applying D_x and D_y. In *Mathematica*, we can realize this by

```
df1 = D[D[f, x], y]; df1 // LieTraditionalForm
```

$$u_y\, u_{x,y} + x\, u_{x,y}^2 + x\, u_y\, u_{x,x,y} + u\, u_{x,y,y} + x\, u_x\, u_{x,y,y} + u\, x\, u_{x,x,y,y}$$

```
df2 = D[D[f, y], x]; df2 // LieTraditionalForm
```

$$u_y\, u_{x,y} + x\, u_{x,y}^2 + x\, u_y\, u_{x,x,y} + u\, u_{x,y,y} + x\, u_x\, u_{x,y,y} + u\, x\, u_{x,x,y,y}$$

Subtracting both representations of the mixed derivatives from each other, we get

```
df1 - df2
```

```
0
```

It is obvious that both expressions contain the same result. This implies that we can commute the D_x and D_y. In general, we can interchange the D's in the calculation in any order.

Another representation of derivatives instrumental in the calculation of symmetries is the prolongation. A prolongation is not a completely new derivative; however, it introduces a geometrical concept in the manifold, allowing a greater flexibility in the use of coordinates.

3.4. Prolongations

In the calculation of symmetries, we frequently have to calculate the prolongation of a given system of differential equations. Here, we first define the term prolongation for a function. In Section 3.7, we will discuss the application of prolongations to vector fields. This definition is extended to differential equations in an additional step in Section 4.2.5 for ODEs and in Section 5.4.1 for PDEs.

The term prolongation actually means an extension of the space of coordinates by their derivatives up to a certain order. As a simple example, we can extend or prolong the space of variables u for a function $u: \mathbb{R} \to \mathbb{R}$ by its first derivative. In classical mechanics, such an extension of the configuration space with coordinates u to a space with u and u' as coordinates is known as an extension of the configuration space to phase space. A more specific example occurring frequently in mathematical physics is given by a vector-valued function u = f(x) = (f_1(x), ..., f_m(x)) with n independent variables $x = (x_1, \ldots, x_n)$ and m dependent variables. For such an $n \times m$ space, the definition of the prolongation reads

Definition: Prolongation

For a given vector-valued function $f: \mathbb{R}^n \to \mathbb{R}^m$, we define the k*th* prolongation of f by

$$\mathrm{pr}^{(k)} f(x) := u^{(k)} \,.\, \circ \tag{3.10}$$

This relation means that we have to determine all derivatives of u up to a certain order k. The result of such a calculation is a set of terms containing all possible derivatives of u up to kth order.

The calculation of the kth prolongation is in some sense equivalent to the calculation of the first k coefficients in a Taylor expansion of f at the point x.

Let us demonstrate the calculation of the prolongation for a single function $f = f(x, y, z)$. Here, $n = 3$ and $m = 1$. We are looking for the second prolongation of f; i.e., $k = 2$. We use *Mathematica* to carry out such a calculation. If we do the calculation by hand, we have to collect the derivatives of f with respect to the independent variables $x = (x, y, z)$ up to order 2

$$\mathrm{pr}^{(2)} f(x, y, z) =$$

$$\left(f,\ \frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y},\ \frac{\partial f}{\partial z},\ \frac{\partial^2 f}{\partial x \partial y},\ \frac{\partial^2 f}{\partial x \partial z},\ \frac{\partial^2 f}{\partial y \partial z},\ \frac{\partial^2 f}{\partial x^2},\ \frac{\partial^2 f}{\partial y^2},\ \frac{\partial^2 f}{\partial z^2} \right). \tag{3.11}$$

This list of terms represents the expansion coefficients of a Taylor series of f around x_0. The first few terms can be read off from the following series expansion:

```
Clear[f, x, y, z, x0, y0, z0];
Normal[Series[f[x, y, z], {x, x0, 1},
  {y, y0, 1}, {z, z0, 1}]] // LieTraditionalForm
```

$$f + (x - x0)\, f_{x0} + (y - y0)\, (f_{y0} + (x - x0)\, f_{x0,y0}) + (z - z0)\, (f_{z0} + (x - x0)\, f_{x0,z0} + (y - y0)\, (f_{y0,z0} + (x - x0)\, f_{x0,y0,z0}))$$

Using *Mathematica*, we do the calculation for the prolongation by applying the function Outer[] in connection with the differentiation D[]. The aim is to reproduce the content of equation (3.11). So we have to define a function called prolongation[] using lists as input variables for the functions f and the independent variables x. The third argument of prolongation[] determines the largest order k of differentiation:

```
prolongation[f_List, x_List, order_] :=
 Block[{aux, dresult},
```

```
result = f;
aux = result;
Do[aux = Outer[D, aux, x];
 AppendTo[result, aux], {i, 1, order}];
Sort[Union[Flatten[result]],
 derivativeOrder[#1, #2]&]]
```

The function prolongation[] is based on the auxiliary function derivativeOrder[]. This function determines the order of a differential expression. The result of derivativeOrder[] influences the sorting of the derivatives in the function Sort[]. derivativeOrder[] allows us to sort the derivatives by an increasing order.

```
derivativeOrder[expr1_, expr2_] :=
 If[FreeQ[expr1, Derivative] ||
   FreeQ[expr2, Derivative], True,
  Plus[expr1 /. _^(x___)[___] → x] <
   Plus[expr2 /. _^(x___)[___] → x]];
```

The function derivativeOrder[] checks the two arguments *expr1* and *expr2* on derivatives. If the expressions are free of derivatives, the function returns *True*. If the expressions contain derivatives, the function only returns True if the order of the derivatives increases. The application of prolongation[] on f[x,y,z] up to second order gives us the coordinates of the extended space

```
prolongation[{f[x, y, z]}, {x, y, z}, 2] // LieTraditionalForm
```

$$\{f,\ f_x,\ f_y,\ f_z,\ f_{x,x},\ f_{x,y},\ f_{x,z},\ f_{y,y},\ f_{y,z},\ f_{z,z}\}$$

From a mathematical point of view, we determined the coordinates of a jet-space of order 2 (cf. Olver [1986]). The kth prolongation $\mathrm{pr}^{(k)}\,(f(x))$ is also known as the k-jet of f. The related space of independent and dependent variables extended by the derivatives is thus called jet-space. Thus, if $u = f(x)$ is a function whose graph lies in the space of dependent and independent variables, the kth prolongation $\mathrm{pr}^{(k)}\,(f(x))$ is a function whose graph lies in the k-jet space.

3.5. The Fréchet Derivative

In the previous sections, we discussed differential operators available in *Mathematica*. This section deals with a differential operator instrumental in the theory of symmetry analysis. Here, we discuss a generalized derivative and its definition in *Mathematica*. The derivative is called a Gateaux or Fréchet derivative. This kind of derivative is very useful in the calculations of symmetries (Olver [1986], Fokas [1980, 1987], Fokas and Fuchssteiner [1981], Baumann [1997]). Such a derivative uses not only the steepest descent of a function but also puts a weight on it.

Definition: Fréchet derivative

Let $f = f(x, u_{(n)})$ be a function in p independent and q dependent variables. $u_{(n)}$ denotes all the derivatives in this function up to order $n = 0, 1, 2, \ldots$. The Fréchet derivative $\mathfrak{D}_f$ of a function f based on $w(x, u_{(n)})$ is defined in such a way that

$$\mathfrak{D}_f(w) = \frac{d}{d\epsilon} f(u + \epsilon w(u))\Big|_{\epsilon=0} \tag{3.12}$$

holds for all auxiliary functions w. We call the function f the support of the Fréchet derivative and w the test function. ○

The algorithmic content of this definition is that $\mathfrak{D}_f(w)$ is calculated by replacing u and all of its derivatives in f by $u + \epsilon w$. If we later differentiate the resulting expression with respect to ϵ and set $\epsilon = 0$, we determined the Fréchet derivative. The result of these two steps is the Fréchet derivative of f based on the test function w. Using the steps in a pencil calculation for one independent and one dependent variable, equation (3.12) can be reduced to an explicit expression like

$$\mathfrak{D}_f = \sum_{k=0}^{\infty} \frac{\partial f}{\partial u_{(k)}} \frac{\partial^k}{\partial x^k}. \tag{3.13}$$

In (3.13) $u_{(k)}$ denotes the kth derivative of u with respect to x. The sum in (3.13) is finite since the order of the largest derivative of the support is finite. This is the case in all practical situations.

Let us consider as a support function $f = u_x\, u_{x,x,x} + u_x^2$. The Fréchet derivative with a test function $w = w(x, u_{x,x,x})$ contained in the class of support functions f is given by applying equation (3.13) to f:

$$\mathfrak{D}_f(w) = (u_{x,x,x} + 2\,u_x)\, D_x\, w + u_x\, D_{x,x,x}\, w, \tag{3.14}$$

where D_x and $D_{x,x,x}$ denote the first- and third-order total derivatives. If we choose the test function as $w = u_x^2/2$, we get from relation (3.14)

$$\mathfrak{D}_f(w) = (u_{x,x,x} + 2\,u_x)\,u_x = u_x\, u_{x,x,x} + 2\,u_x^2 \tag{3.15}$$

representing a differential expression containing only derivatives of u. The definition of the Fréchet derivative given above for one independent and one dependent variable is easily generalized to a vector of r support functions $f = (f_1, \ldots, f_r)$ and q test functions $w = (w_1, w_2, \ldots, w_q)$. Then the Fréchet derivative of such an r-tuple is given by the relation

$$\mathfrak{D}_f(w) = \frac{d}{d\epsilon}\begin{pmatrix} f_1(u_1 + \epsilon w_1, u_2, \ldots, u_r) + \cdots + f_1(u_1, u_2, \ldots, u_r + \epsilon w_r) \\ f_2(u_1 + \epsilon w_1, u_2, \ldots, u_r) + \cdots + f_2(u_1, u_2, \ldots, u_r + \epsilon w_r) \\ \vdots \\ f_r(u_1 + \epsilon w_1, u_2, \ldots, u_r) + \cdots + f_r(u_1, u_2, \ldots, u_r + \epsilon w_r) \end{pmatrix}\Bigg|_{\epsilon=0}. \quad (3.16)$$

Introducing the q test functions as a column vector allows us to define the $q \times r$ matrix differential operator

$$(\mathfrak{D}_f(w))_{\mu\alpha} = \frac{d}{d\epsilon} f_\mu(u_\alpha + \epsilon w_\alpha(u))\Big|_{\epsilon=0}, \qquad \mu = 1, \ldots, r, \quad \alpha = 1, \ldots, q. \quad (3.17)$$

This expression is equivalent to the matrix differential operator

$$(\mathfrak{D}_f)_{\mu\alpha} = \sum_J \frac{\partial f_\mu}{\partial u^\alpha_J} D_J, \quad \mu = 1, \ldots, r; \quad \alpha = 1, \ldots, q. \quad (3.18)$$

The sum in (3.18) extends over all multi-indices J. To define the Fréchet derivative in *Mathematica*, we use relation (3.12) and its matrix version (3.16). We note again that the u^α and all their derivatives are replaced by $u^\alpha + \epsilon w^\alpha$. After the replacement of the arguments, we differentiate with respect to ϵ and set $\epsilon = 0$ in the next step. The result in the general case is a matrix containing the derivatives of the support f based on the test functions w^α. The implementation of the Fréchet derivative in *Mathematica* is

```
FrechetD[support_List, dependVar_List,
 independVar_List, testfunction_List] :=
 Block[{indep, frechet, deriv, ϵ, r0, x1, x2},
  r0 = Function[indep, x1 + ϵ x2];
  frechet = {}; Do[deriv = {};
   Do[AppendTo[deriv, ∂ϵ (support[[j]] /.
         dependVar[[i]] → (r0 /.
             {indep → independVar,
              x1 → dependVar[[i]]@@independVar,
 x2 → testfunction[[i]]@@independVar})) /. ϵ → 0],
    {i, 1, Length[support]}];
   AppendTo[frechet, deriv],
   {j, 1, Length[support]}];
  frechet]
```

The code of the Fréchet derivative follows closely the relation given in equation (3.12). In FrechetD[], we first define a pure function stored in the variable r_0. This function serves as a general pattern to replace the original argument by a varied argument. A loop extending over the number of dependent variables replaces the independent and dependent variables. This step creates an explicit rule for the replacement. After the replacement, a differentiation with respect to the parameter ϵ is performed and, at the end, ϵ is replaced by zero. The resulting expressions are

collected in the list *frechet* which is returned by the function. An example will demonstrate the application of the function.

Example 1

Consider a set of two expressions representing a system of partial differential equations given by

$$\begin{aligned} v_x - u &= 0, \\ v_t - \frac{u_x}{u^2} &= 0, \end{aligned} \tag{3.19}$$

where u and v are functions of x and t. This set of equations is equivalent to a non-linear diffusion equation in v. Our aim is to calculate the Fréchet derivative of the left-hand side of the system (3.19). Let us define a variable *eqsys* containing the left-hand side of the equations:

```
eqsys = {∂x v[x, t] - u[x, t], ∂t v[x, t] - ∂x u[x, t] / u[x, t]^2};
eqsys // LieTraditionalForm
```

$$\left\{-u + v_x,\ -\frac{u_x}{u^2} + v_t\right\}$$

The application of our function FrechetD[] to this expression gives us

```
FrechetD[eqsys, {u, v}, {x, t}, {w1, w2}] //
 MatrixForm // LieTraditionalForm
```

$$\begin{pmatrix} -w1 & w2_x \\ \frac{2\, w1\, u_x}{u^3} - \frac{w1_x}{u^2} & w2_t \end{pmatrix}$$

The calculation of the Fréchet derivative using FrechetD[] is carried out by supplying four arguments containing the equations: the dependent variables of the support, the independent variables, and the test functions w_1 and w_2. The result is a 2×2 matrix containing expressions of w_1 and w_2 and their derivatives. From the result, we can get the corresponding operators if we consider w_1 and w_2 as auxiliary functions. The related matrix operator reads

$$\begin{pmatrix} -1 & \partial_x \\ \frac{2\, u_x}{u^3} - \frac{1}{u^2}\,\partial_x & \partial_t \end{pmatrix}. \tag{3.20}$$

In our symmetry calculations, we sometimes need also the adjoint representation of the Fréchet derivative. In general, the adjoint representation of a differential operator is defined via an integral expression. Assume that we know the differential operator (3.12). We denote the adjoint operator of $\mathfrak{D}_f$ by $\mathfrak{D}_f^*$ satisfying

$$\int_\Omega V\, \mathfrak{D}_f\, W\, d\,x \;=\; \int_\Omega W\, \mathfrak{D}_f^*\, V\, d\,x. \tag{3.21}$$

Equation (3.21) holds for any pair of functions W and V. If we examine definition (3.21) in more detail, we can replace the integral operation by a plain differential representation (cf. Olver [1986]). The corresponding expression to (3.18) is given by

$$(\mathfrak{D}_f^*)_{\mu\alpha} \;=\; \sum_J\, (-1)^J\; D_J\left(\frac{\partial f_\mu}{\partial u_J^\alpha}\right), \qquad \mu = 1,\, \ldots,\, r;\;\; \alpha = 1,\, \ldots,\, q\,. \tag{3.22}$$

In view of an algorithm in *Mathematica,* this means that we convert derivatives of the test functions to derivatives of the support multiplied by some coefficients. This is strictly the definition of an adjoint differential operator. The described procedure is implemented in the following function AdjointFrechetD[]:

```
AdjointFrechetD[support_List, dependVar_List,
 independVar_List, testfunction_List] :=
 Block[{subrule, $testf, frechet, n, b},
  subrule = b_. $testf^(n___) @@independVar :>
    (-1)^Plus@@{n} ∂_Delete[Thread[{independVar, {n}}], 0] (b $testf@@independVar)
  frechet = FrechetD[support, dependVar,
    independVar, testfunction];
  Do[frechet = frechet /.
     (subrule /. $testf → testfunction[[i]]),
   {i, 1, Length[testfunction]}];
  frechet = Transpose[frechet]]
```

The adjoint representation of the Fréchet derivative of the system *eqsys* is thus given by

```
AdjointFrechetD[eqsys, {u, v}, {x, t},
 {w1, w2}] // MatrixForm // LieTraditionalForm
```

$$\begin{pmatrix} -w1 & \frac{w1_x}{u^2} \\ -w2_x & -w2_t \end{pmatrix}$$

3.6. The Euler Derivative

In this section, we will discuss the Euler derivative. The Euler derivative, also known as the functional derivative, has its origin in the *calculus of variations*. The term *calculus of variations* was first coined by Leonhard Euler in 1756. He used it to describe a new method in mechanics which Lagrange had developed 1 year earlier. Thus, the original application of the Euler derivative originates from mechanics. In this context, Euler and Lagrange used this sort of derivative to write down their famous equations, the Euler-Lagrange equations. Up to now, the main application of this derivative in physics has been the formulation of dynamical equations. In Chapter 9, we will show that the Euler derivative is a useful tool in connection with Lie-Bäcklund or generalized symmetries. Before we discuss the Euler derivative and its implementation, we recall briefly the basic properties of the origin in the calculus of variations.

3.6.1 The Problem of Variation

The calculus of variations was first used by Johann Bernoulli in July 1696, when he presented the *brachystochrone problem*. The problem can be formulated as follows. A point mass is moving frictionless in a homogenous force field along a path joining two points. The question is which curve connects the two points for the shortest travel. Johann Bernoulli announced the solution of the problem, but did not present his findings in public. He preferred to first challenge his contemporaries to examine the problem, too. This challenge was particularly aimed at his brother and teacher Jakob Bernoulli, who was his bitter enemy. Jakob found one solution, but did not present it to Johann. It was only upon the intervention of Leibniz, with whom Jakob had a lifelong friendship and a scientific correspondence, that he sent it to his brother in May 1697. The most fascinating event was that this solution was a cycloid, a curve also discovered at this time.

As mentioned above, the main idea in the calculus of variations arose from the work of Euler and Lagrange. Later, Hamilton contributed the term *minimum principle* to the theory, and it is still in use today. The main idea of all these considerations of Euler, Lagrange, and Hamilton is the assumption that there exists a generating functional F. This functional is responsible for the dynamical development of the motion. The key point in the calculus of variations is to find a function which extremizes the functional F. The solution of this issue is to vary the function by introducing a test function. Thus, the variation of F is actually carried out by replacing the function u by a slightly changed new function $u + \epsilon w$, where ϵ is a small parameter and w denotes an arbitrary test function. After replacing u and all of its higher derivatives in the functional F, we have to determine the extreme of F. The

functional in this representation can be considered as a function of the parameter ϵ. The maximum or minimum of F is found if we use the standard procedure of calculus for finding extreme values. In mathematical terms, we need to calculate the derivative of F with respect to ϵ under the condition that ϵ vanishes:

$$\frac{dF(\epsilon)}{d\epsilon}\Big|_{\epsilon=0} = 0. \tag{3.23}$$

The basic problem of the calculus of variations is to determine a function $u(x)$ such that the integral

$$F[u] = \int_{x_1}^{x_2} f(x, u, u_x, \ldots)\, dx = \int_{x_1}^{x_2} f(x, u_{(k)})\, dx, \qquad k = 1, 2, \ldots \tag{3.24}$$

assumes an extreme. An extreme here is either a maximum or a minimum. In equation (3.24), $u_x = \partial u / \partial x$ denotes the partial derivative of u with respect to the independent variables x, where x is a vector of coordinates. Let us assume first that we have only one independent variable x. This assumption will make it easier to represent and discuss the theory. A generalization to more independent variables will be given below.

The expression $F[u]$ given in equation (3.24) is called a functional defined by an integral over a density f which depends on the independent variable x and the unknown function u. In general, this density may also depend on derivatives of u up to a certain order k, denoted by $u_{(k)}$. The limits in the integral (3.24) are assumed to be fixed. We note that fixed limits are not necessary. If they are allowed to vary, the problem increases in such a way that not only $u(x)$ but also x_1 and x_2 are needed to bring F to an extreme value. The question is how to manage the functional F in becoming an extreme. Let us assume that an extreme of F exists if a function $u = u(x)$ makes the functional F a minimum. Then, any neighboring function, no matter how close it approaches $u(x)$, must make F increase. The definition of a neighboring or test function may be as follows. We introduce a parametric representation of $u = u(x; \epsilon)$ in such a way that for $\epsilon = 0$, $u = u(x; \epsilon = 0) = u(x)$, we get the identity and the functional yields an extreme. We write the small perturbation of u as

$$u(x; \epsilon) = u(x; 0) + \epsilon w(x), \tag{3.25}$$

where $w(x)$ is the test function which has continuous derivatives and vanishes at the endpoints x_1 and x_2. We note that the vanishing of $w(x)$ at x_1 and x_2 $w(x_1) = w(x_2) = 0$ is one of the basic assumptions of the calculus of variations.

If functions of the type given in equation (3.25) are considered as variations of u, the functional F becomes a function of ϵ:

$$F[u;\epsilon] = \int_{x_1}^{x_2} f(x, u(x;\epsilon), u_x(x,\epsilon), \ldots)\, dx. \tag{3.26}$$

The condition that the integral has a stationary value (in other words, an extreme) is that F be independent of ϵ in first order. This means that

$$\frac{\partial F}{\partial \epsilon}\Big|_{\epsilon=0} = 0 \tag{3.27}$$

for all functions $w(x)$. This is a necessary condition but not a sufficient one. We will not pursue the details of the sufficient conditions here. They were extensively discussed by Blanchard and Brüning [1992]. To demonstrate how these formulas work in detail, let us consider the simple example of the shortest connection between two points in an Euclidean plane.

Example 1

Let us consider the equation of a curve in a Euclidean space which yields the shortest distance between two points in the plane. The geometrical increment of distance ds in the (u, x)-plane is given by

$$ds = \sqrt{dx^2 + du^2} = \sqrt{1 + \left(\frac{du}{dx}\right)^2}\, dx. \tag{3.28}$$

The total length s of the curve between two points x_1 and x_2 is

$$s = \int_{x_1}^{x_2} \sqrt{1 + u_x^2}\, dx \equiv F[u]. \tag{3.29}$$

We know that the shortest connection between two points in the Euclidean plane is a straight line given by

$$u(x) = \alpha x + \beta, \tag{3.30}$$

where α and β are constants determining the slope and the intersection of the line with the vertical coordinate axis. Now let us consider the line in the range $x \in [0, 2\pi]$. To demonstrate the numerical behavior of the functional F, we choose a special test function $w(x) = \sin(4x)$. Using our representation of u given by equation (3.30) with α=1 and β=0 for example, we get for the derivative of u,

$$u_x = 1 + 4\epsilon\cos(4x). \tag{3.31}$$

Inserting this representation into (3.29) we find

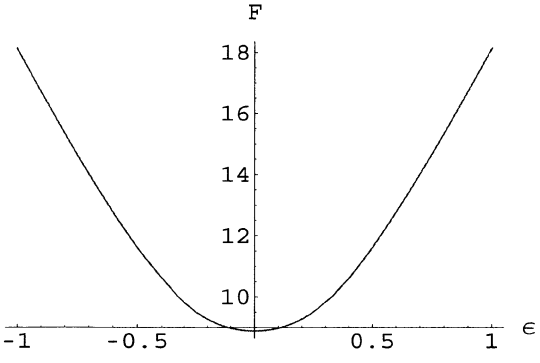
$$F[\epsilon] = \int_0^{2\pi} \sqrt{1 + 4\,\epsilon \cos(4\,x)}\; dx. \tag{3.32}$$

This relation represents our specific functional. We are looking for the minimum of this function to get the extreme of the functional. Considered as a function of ϵ, this relation cannot be solved for ϵ. However, to get an idea of the dependence on the parameter ϵ, we can use *Mathematica*. If we define equation (3.32) as a function depending on ϵ, we can use the numerical capabilities of *Mathematica* to graphically represent the dependence of F on ϵ. First, let us define equation (3.32) by

```
F[ϵ_] := NIntegrate[
  Sqrt[1 + (1 + 4 ϵ Cos[4 x])^2], {x, 0, 2 π}]
```

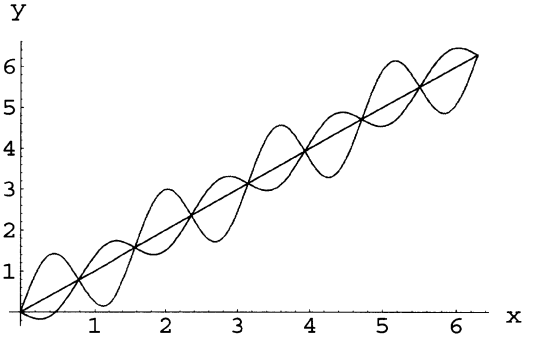
We then use the defined function F[] in connection with Plot[] to represent the value of the functional for certain values of ϵ:

```
Plot[Evaluate[F[ϵ]], {ϵ, -1, 1}, AxesLabel → {"ϵ", "F"},
 PlotStyle → RGBColor[1, 0, 0]]
```



The result of our calculation shows that the value of the functional is minimal for ϵ=0 and increases for all other values of ϵ. Thus, we demonstrated numerically that the minimum of the functional exists. In a second plot, we demonstrate the influence of ϵ on the function $u(x) = x$ for different values of ϵ. This shows us that the value of $F[u; \epsilon]$ is always greater than $F[u; 0]$, no matter which value (positive or negative) is chosen for ϵ.

```
Plot[Evaluate[
  {y[x, 0], y[x, 1], y[x, -1/2]} /.
   y → Function[{x, ϵ}, x + ϵ Sin[4 x]]],
 {x, 0, 2 π},
 AxesLabel → {"x", "y"},
 PlotRange → All,
 PlotStyle → {RGBColor[0, 0, 0.996109],
   RGBColor[1.000, 0.000, 0.000],
   RGBColor[0.000, 0.251, 0.251]}]
```



From this figure, we can conclude that the line $u(x) = x$ is one realization of the shortest connection between two points in the Euclidean plane. □

3.6.2 Euler's Equation

In this section, we derive the analytical representation of the Euler derivative. The construction of this sort of derivative is based on condition (3.27). If we carry out the differentiation with respect to ϵ, equation (3.26) will provide

$$\frac{\partial F}{\partial \epsilon} = \frac{\partial}{\partial \epsilon} \int_{x_1}^{x_2} f(x, u, u_x, \ldots)\, dx. \tag{3.33}$$

Since the limits of the integral are fixed, the differentiation affects only the density of the functional F. Hence,

$$\frac{\partial F}{\partial \epsilon} = \int_{x_1}^{x_2} \left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \epsilon} + \frac{\partial f}{\partial u_x}\frac{\partial u_x}{\partial \epsilon} + \frac{\partial f}{\partial u_{x,x}}\frac{\partial u_{x,x}}{\partial \epsilon} + \cdots \right) dx. \tag{3.34}$$

If we now use the representation of $u = u(x; \epsilon)$ as given in (3.25) to introduce the ϵ dependence for the variable u and the derivatives $u_{(k)}$, we get

$$\frac{\partial u}{\partial e} = w(x), \qquad \frac{\partial u_x}{\partial \epsilon} = w_x, \qquad \frac{\partial u_{x,x}}{\partial \epsilon} = w_{x,x,x}, \quad \ldots \tag{3.35}$$

Using these relations in equation (3.34), we find

$$\frac{\partial F}{\partial \epsilon} = \int_{x_1}^{x_2} \left(\frac{\partial f}{\partial u} w(x) + \frac{\partial f}{\partial u_x} w_x + \frac{\partial f}{\partial u_{x,x}} w_{x,x} + \cdots \right) d x . \tag{3.36}$$

The result so far is that the integrand contains derivatives of the density f and the test function w. Since we do not know anything about the derivatives of w, we need to reduce (3.36) in such a way that it only contains the test function w. The reduction can be obtained by an integration of parts with respect to the test function. Additional use of the conditions $w(x_1) = w(x_2) = 0$ simplifies expression (3.36) to

$$\frac{\partial F}{\partial \epsilon} = \int_{x_1}^{x_2} w(x) \left(\frac{\partial f}{\partial u} - \frac{d}{d x} \left(\frac{\partial f}{\partial u_x} \right) + \frac{d^2}{d x^2} \left(\frac{\partial f}{\partial u_{x,x}} \right) \mp \cdots \right) d x. \tag{3.37}$$

The integral in equation (3.37) seems to be independent of ϵ. However, the function $u = u(x; \epsilon)$ and all derivatives of u are still functions of ϵ. We know from the representation of $u(x; \epsilon)$ that this dependency disappears if we set $\epsilon = 0$. Before we start this calculation, we generalize (3.37) to arbitrary orders in the derivatives:

$$\frac{\partial F}{\partial \epsilon} = \int_{x_1}^{x_2} w(x) \left(\sum_{n=0}^{\infty} (-1)^n \frac{d^n}{d x^n} \left(\frac{\partial f}{\partial u_{(n)}} \right) \right) d x, \tag{3.38}$$

where $u_{(n)} = \frac{\partial^n u}{\partial x^n}$ denotes the nth derivative of u with respect to x. Our aim was to find the extreme of F. A necessary condition for the existence of an extreme is the vanishing of the derivative $\frac{\partial F}{\partial \epsilon}\big|_{\epsilon=0} = 0$. In our calculations, we assumed that w is an arbitrary function. Thus, the derivative of F can only vanish if the integrand vanishes and so we end up with the result

$$\sum_{n=0}^{\infty} (-1)^n \frac{d^n}{d x^n} \left(\frac{\partial f}{\partial u_{(n)}} \right) = 0, \tag{3.39}$$

where u and all the derivatives of u are now independent of ϵ. This result is known as Euler's equation and it is a necessary condition for the functional F to allow an extreme. The Euler equation is reduced to the well-known Euler-Lagrange equation if we restrict the order of the derivatives to 2. Since the Euler equation is needed in the calculation of symmetries, we define a special symbol for this operation and call it the Euler operator.

3.6.3 Euler Operator

The Euler operator is also known as a variational derivative in the field of dynamical formulations or statistical mechanics. In this section, we define this operator as a special type of derivative.

Definition: Euler operator

Let $f = f(x, u, u_x, \ldots)$ be the density of a functional $F[u]$. Then we call

$$\frac{\delta F}{\delta u} := \sum_{n=0}^{\infty} (-1)^n \frac{d^n}{dx^n}\left(\frac{\partial f}{\partial u_{(n)}}\right)$$

the functional derivative of F and

$$\in := \sum_{n=0}^{\infty} (-1)^n D_n \frac{\partial}{\partial u_{(n)}}$$

an Euler operator. $D_n = \frac{d^n}{dx^n}$ denotes the nth-order total derivative. ○

The actual information of this definition is that the functional derivative $\frac{\delta F}{\delta u}$ can be replaced by ordinary and partial derivatives if we know the density of the functional F. Consequently, we can introduce a general derivative, the Euler operator, which is based on known operations. The essential content of the definition above is that knowing the density f of a functional F is sufficient to calculate the corresponding functional derivative. The functional derivative follows just by differentiation of the density f. An additional merit is the knowledge of the Euler equation for this functional F. The definition from above is a result of the calculus of variations. Thus, the Euler derivative can be calculated by an algorithmic procedure.

3.6.4 Algorithm Used in the Calculus of Variations

Our next goal is to define a *Mathematica* function allowing the calculation of the Euler derivative. Before we present the function, we briefly repeat the main steps of the calculus of variations. These steps are intimately related to the definition of the Euler derivative and are thus the basis of the calculation. The four main steps of the algorithm are as follows:

1. Replacement of the dependent function u by its variation $u = u + \epsilon w$.
2. Differentiation of the functional density with respect to the parameter ϵ and replacement of ϵ by zero after the differentiation.
3. Use the boundary conditions for the test function to eliminate the derivatives in w.
4. The coefficient of the test function w delivers the Euler equation.

These four steps define the calculation of the Euler derivative algorithmically. The function defined in *Mathematica* is based on these four steps. When looking at the definition of the Euler derivative $\in$, we realize that we need at least three pieces of information to carry out the calculation. First, we should know the density of the functional F, second the dependent variable, and third the name of the independent variable. From our discussions of the algorithm, we expect that the highest order of differentiation should be determined by the function itself. Thus, we define the function EulerD[] with three necessary arguments. A fourth optional argument allows influencing the representation of the result of the function. The following lines contain the code for EulerD[]:

```
(* --- Euler derivative for ---*)
(* --- one dependent and one independet variable ---*)
Clear[EulerD];
Options[EulerD] = {eXpand → False};

EulerD[density_, depend_, independ_,
 options___] :=
 Block[{f0, rule, fh, ϵ, w, y, expand},
  (*--- check options ---*)
  {expand} = {eXpand} /. {options} /.
    Options[EulerD];
  (*--- rule for the variation of u---*)
  f0 = Function[x, y[x] + ϵ w[x]];
  (*--- rule for the replacement of
    derivatives of w --- *)
  rule = b_. w^(n_) [independ] :→
    (-1)^n Hold[∂{independ,n} b];
  (*--- step of variation ---*)
  fh = density /. depend → f0 /.
    {x → independ, y → depend};
  (*--- differentiation with respect to ϵ ---*)
  fh = Expand[∂ϵ fh /. ϵ → 0];
  (*--- transformation to w ---*)
  fh = fh /. rule /. w[independ] → 1;
  (*---- Euler equations --- *)
  If[expand, fh = ReleaseHold[fh], fh]]
```

Using this function, it is straightforward to calculate the functional derivative of any density containing one dependent and one independent variable. We demonstrate the application of this function by discussing the famous brachistochrone problem already mentioned in the introduction.

Example 1

Let us discuss the classical problem of the brachystochrone solved by Johann Bernoulli in 1696. The physical content of this famous problem is the following: Consider a particle moving in a constant force field. The particle with mass m starts at rest from some higher point in the force field and moves to some lower point. The question is which path is selected by the particle to finish the transit in the least possible time. Let us reduce the problem to the point of deriving the Euler equation. The dimensionless functional density governing the movement of the particle can be derived from the integral $t = \int_{p_1}^{p_2} 1/v\, ds$ where t is time, ds the line element, and v the velocity. Expressing the line element and the velocity in Cartesian coordinates, we can express the density of the functional by

$$f(x, u, u_x) = \left(\frac{1 + u_x^2}{2gx} \right)^{1/2}, \tag{3.40}$$

where u describes the horizontal coordinate and x the vertical one. The application of our function EulerD[] on this functional density

```
f = Sqrt[(1 + (∂x u[x])^2)/(2 g x)]; f // LieTraditionalForm
```

$$\frac{\sqrt{\frac{1 + u_x^2}{g\, x}}}{\sqrt{2}}$$

gives us

```
EulerD[f, u, x, eXpand → True] // PowerExpand // Simplify //
 LieTraditionalForm
```

$$\frac{u_x + u_x^3 - 2\, x\, u_{x,x}}{2\sqrt{2}\,\sqrt{g}\, x^{3/2}\, (1 + u_x^2)^{3/2}}$$

a second-order ordinary differential equation for the variable u. The solution of this equation is a cycloid and can be derived by applying *Mathematica* (cf. Baumann [1996]). □

Example 2

Another example of the application of the function EulerD[] is the derivation of the Euler-Lagrange equation for a mechanical system with one degree of freedom. The functional density for such a problem is generally given by the Lagrange function $\mathcal{L}$

```
ℒ = l[t, q[t], q'[t]]; ℒ // LieTraditionalForm
```

$$l[t, q, q_t]$$

where q denotes the generalized coordinate of the particle and t the time. The Euler-Lagrange equation for this general Lagrangian then follows by

```
EulerD[ℒ, q, t, eXpand → False]
```

$$-\text{Hold}[\partial_{\{t,1\}}\, l^{(0,0,1)}[t, q[t], q'[t]]] + l^{(0,1,0)}[t, q[t], q'[t]]$$

representing the left-hand side of the expression

$$\frac{\partial}{\partial q}\mathcal{L} - \frac{d}{dt}\left(\frac{\partial}{\partial q_t}\mathcal{L}\right) = 0. \tag{3.41}$$

The disadvantage of this representation is the appearance of the function Hold[] in the equation. However, if we are only interested in the explicit form of the equations, we can set the option *eXpand→True*. Then, the result reads

```
EulerD[ℒ, q, t, eXpand → True] == 0 // LieTraditionalForm
```

$$l_q - q_t\, l_{q,q_t} - l_{t,q_t} - l_{q_t,q_t}\, q_{t,t} == 0$$

This equation is the general representation of the Euler-Lagrange equation. □

The Euler operator defined above was the result of the variation of a functional. We demonstrated the calculation for a single dependent variable $u = u(x)$ which was a function of one independent variable x. The generic case in applications is more complex. We rarely find systems with only one dependent variable. Thus, we need a generalization of the formulation considering more than one dependent variable in the functional F. In the following exposition, we assume that a set of q dependent variables u^α exists. The functional F for such a case is represented by

$$F[u^1, u^2, u^3, \ldots] = \int_{x_1}^{x_2} f(x, u^1, \ldots, u^1_x, \ldots)\, dx. \tag{3.42}$$

The variation of the dependent variables is now performed by introducing a set of test functions w^α. Using this set of auxiliary functions, we can represent the variation by

$$u^\alpha(x;\epsilon) = u^\alpha(x;0) + \epsilon w^\alpha(x), \quad \alpha = 1, 2, 3, \ldots, q. \tag{3.43}$$

The derivation of the Euler operator proceeds in exactly the same manner as presented above. We skip the detailed calculations and present only the result:

$$\frac{\partial F}{\partial \epsilon} = \int_{x_1}^{x_2} \sum_{\alpha=1}^{q} \left\{ \sum_{n=1}^{\infty} (-1)^n \, D_{(n)} \, \frac{\partial f}{\partial u^{\alpha}_{(n)}} \right\} w^{\alpha}(x) \, dx. \tag{3.44}$$

Since the individual variations $w^{\alpha}(x)$ are all independent of each other, the vanishing of equation (3.23) when evaluated at ϵ=0 requires the separate vanishing of each expression in curly brackets. Thus, we again can define an Euler operator for each of the q dependent variables u^{α}.

3.6.5 Euler Operator for q Dependent Variables

In this section, we extend the definition of the Euler derivative to a set of q dependent variables. Let $f = f(x, u^1, u^2, \ldots, u^1_x, u^2_x, \ldots)$ be the density of the functional $F[u^1, u^2, \ldots]$. Then, we define the Euler operator $\in_{\alpha}$ as

$$\in_{\alpha} := \sum_{n=0}^{\infty} (-1)^n \, D_{(n)} \, \frac{\partial}{\partial u^{\alpha}_{(n)}}, \quad \alpha = 1, 2, \ldots, q, \tag{3.45}$$

which will give us the αth Euler equation when applied to the density f,

$$\in_{\alpha} f = 0. \tag{3.46}$$

The only difference between this definition and the definition for the single variable is the number of equations contained in (3.46). The occurrence of the q equations in the theoretical formulas must now be incorporated in our *Mathematica* definition for the Euler derivative EulerD[]. The theoretical definition (3.45) only alters our *Mathematica* function in a way that, for several dependent variables, a set of Euler equations results. Thus, we change our *Mathematica* function in such a way that all dependent variables are taken into account in the application of the $\in_{\alpha}$ operator. We realize this by including a loop scanning the input list of the dependent variables. The code of this generalized Euler operator is

```
EulerD[density_, depend_List,
 independ_, options___] :=
 Block[{f0, fh, e, w, y, expand,
   euler = {}, wtable},
  {expand} = {eXpand} /. {options} /.
    Options[EulerD];
  wtable = Table[w[i],
    {i, 1, Length[depend]}];
  f0 = Function[x, y[x] + e w[x]];
```

```
rules[i_] :=
 b_. wtable[[i]]^(n_) [independ] :>
  (-1)^n Hold[∂{independ,n} b];
Do[
 fh = density /. depend[[j]] → f0 /.
   {x → independ, y → depend[[j]],
    w → wtable[[j]]};
 fh = Expand[∂ε fh /. ε → 0];
 fh = fh /. rules[j] /.
   wtable[[j]][independ] → 1;
 AppendTo[euler, fh],
 {j, 1, Length[depend]}];
If[expand,
 euler = ReleaseHold[euler],
 euler]]
```

Let us demonstrate the application of this function by two examples.

Example 1

Assume that we know the functional density of a two-dimensional oscillator system. Let us further assume that the two coordinates of the oscillators are coupled by a product. We expect that the two equations of motion follow by applying the Euler derivative. The Lagrange density of the system reads

```
l = u[t] v[t] + (∂t u[t])^2 + (∂t v[t])^2 - u[t]^2 - v[t]^2;
l // LieTraditionalForm
```

$$-u^2 + u\,v - v^2 + u_t^2 + v_t^2$$

The corresponding system of second-order equations follows by

```
EulerD[l, {u, v}, t, eXpand → True] // LieTraditionalForm
```

$$\{-2\,u + v - 2\,u_{t,t},\ u - 2\,v - 2\,v_{t,t}\}$$

which are the left-hand sides of the Euler-Lagrange equations. □

Note that we used the same name, EulerD[], for the operators $\in$ and $\in_\alpha$. This sort of definition is possible in *Mathematica* and provides a great flexibility in the application of a single symbol for different operations. *Mathematica* is able to distinguish the two different functions by the different arguments.

Example 2

Another example for a two-dimensional Lagrangian is given by the function

```
f = u[t] v[t] + (∂t u[t])^2 + (∂t v[t])^2 + 2 ∂t u[t] ∂t v[t];
f // LieTraditionalForm
```

$$u\,v + u_t^2 + 2\,u_t\,v_t + v_t^2$$

This density is a special model of a Dirac Lagrangian containing the derivatives with respect to time as a binomial. The corresponding Euler-Lagrange equations read

```
EulerD[f, {u, v}, t, eXpand → True] // LieTraditionalForm
```

$$\{v - 2\,u_{t,t} - 2\,v_{t,t},\ u - 2\,u_{t,t} - 2\,v_{t,t}\}$$

representing a coupled system of second-order ordinary differential equations. □

So far, we are able to handle point systems depending on one independent variable. However, equations occurring in real situations depend on more than one variable. Thus, we need a generalization of our Euler derivative to more than one independent variable. In fact, the definitions of an Euler operator can be extended from the $q+1$-dimensional case to the $q+p$-dimensional case. We define this operator in the following section.

3.6.6 Euler Operator for $q+p$ Dimensions

Here, we will discuss the general definition of an Euler operator. This sort of operator, for example, is used to write down field equations such as Maxwell's equations, Schrödinger's equation, Euler's equation in hydrodynamics, and many others.

Definition: (q,p)-Dimensional Euler operator

Let $f = f(x, u_{(n)})$ be the density of the functional $F[u]$ with $x = (x^1, x^2, \ldots, x^p)$, and $u = (u^1, u^2, \ldots, u^q)$ the p- and q-dimensional vectors of the independent and dependent variables. By $u_{(n)}$ we denote all the derivatives with respect to the independent variables. We call

$$\mathrm{E}_\alpha = \sum_J (-D)_J \frac{\partial}{\partial u_J^\alpha} \tag{3.47}$$

the general Euler operator in q dependent and p independent variables. J is a multi-index $J = (j_1, \ldots, j_k)$ with $1 \le j_k \le p, k \ge 0$. ○

Since the functional densities f depend on a finite number of derivatives u_J^α, the infinite sum in (3.47) is terminated at this upper limit. Again, the Euler equations for a given functional $F[u]$ follow from the application of E_α to F:

$$\in_\alpha F = 0\,, \quad \alpha = 1, 2, \ldots, q\,. \tag{3.48}$$

From a theoretical point of view, we know the general Euler operator. Our next step is to make this operation available in *Mathematica*. We define the generalized Euler operator by taking into account the different independent variables. The corresponding definition of EulerD[] for $q+p$ dimensions is given by

```
Clear[EulerD]
EulerD[density_, depend_List,
 independ_List, options___] :=
 Block[{f0, fh, e, w, y, x$m, expand,
   euler = {}, wtable},
  {expand} = {eXpand} /. {options} /.
    Options[EulerD];
  wtable = Table[w[i],
    {i, 1, Length[depend]}];
  f0 = Function[x$m, y + e w];
  ruleg[i_] :=
   b_. wtable[[i]]^(n___) @@independ :>
    (-1)^Plus@@{n} Hold[∂Delete[Thread[{independ,{n}}],0] b];
  Do[
   fh = density /. depend[[j]] → f0 /.
     {x$m → independ,
      y → depend[[j]]@@independ,
      w → wtable[[j]]@@independ};
   fh = Expand[∂e fh /. e → 0];
   fh = fh /. ruleg[j] /.
     wtable[[j]]@@independ → 1;
   AppendTo[euler, fh],
   {j, 1, Length[depend]}];
  If[Not[expand],
   euler = ReleaseHold[euler],
   euler]]
```

We demonstrate the application of the function EulerD[] to the wave equation in 2+1 dimensions and to a system of coupled non-linear diffusion equations.

Example 1

Let us consider a functional in $q = 1$ and $p = 3$ variables and assume that the density is quadratic in the derivatives given by

$$F[u] = \frac{1}{2}\int \left(u_{x_1}^2(x_1, x_2, x_3) - u_{x_2}^2 - u_{x_3}^2\right) \boldsymbol{d}\, x_1\, \boldsymbol{d}\, x_2\, \boldsymbol{d}\, x_3\,. \tag{3.49}$$

Calculating the variational derivative, we immediately find that the Euler equations are given by the Laplace equation:

$$-u_{x_1\,x_1} + u_{x_2,x_2} + u_{x_3,x_3} = 0. \tag{3.50}$$

Using the generalized definition of EulerD[], we can reconstruct the result of our pencil calculation. First, let us define the density by

```
f = 1/2 {(∂x1 u[x1, x2, x3])^2 -
    (∂x2 u[x1, x2, x3])^2 - (∂x3 u[x1, x2, x3])^2};
f // LieTraditionalForm
```

$$\left\{\frac{1}{2}\left(u_{x1}^2 - u_{x2}^2 - u_{x3}^2\right)\right\}$$

The application of EulerD[] to f gives

```
wave = EulerD[f, {u}, {x1, x2, x3}];
Map[# == 0&, Flatten[wave]] // LieTraditionalForm // TableForm
```

$$-u_{x1,x1} + u_{x2,x2} + u_{x3,x3} == 0$$

The resulting equation is known as the wave equation in $2+1$ dimensions. □

Example 2

In this example, we will consider a system in two field variables ($q = 2$) and two independent variables ($p = 2$). The physical background of this model is the diffusion of two components in a non-linear medium. The Lagrange density of this field model has the representation

```
l = v[x, t] ∂t u[x, t] + ∂x u[x, t] ∂x v[x, t] + u[x, t]^2 v[x, t]^2;
l // LieTraditionalForm
```

$$u^2\,v^2 + v\,u_t + u_x\,v_x$$

The related equations of motion follow by

```
cnondiffu = EulerD[l, {u, v}, {x, t}];
Map[# == 0&, cnondiffu] // LieTraditionalForm // TableForm
```

$$2\,u\,v^2 - v_t - v_{x,x} == 0$$
$$2\,u^2\,v + u_t - u_{x,x} == 0$$

representing two coupled non-linear diffusion equations for the variables u and v. □

So far, we discussed differential operators like tangent vectors, prolongations, Fréchet derivatives, and several versions of Euler derivatives. All these differentials are non-standard operators with special applications. One of these applications is symmetry analysis of differential equations. In the following section, we discuss the central usage of these differential operators. We will show that the Fréchet derivative is the main link between differential equations and their symmetries.

3.7. Prolongation of Vector Fields

In Lie's theory, a vector field takes a central role in analyzing symmetries of differential equations. A vector field is closely related to the term of the tangent on a curve. The following will show the connection between the tangent of a curve and the related vector field. We will also calculate the prolongation or extension of a vector field which is instrumental in symmetry analysis. The calculation of the extension of a vector field is one of the central terms in Lie's theory.

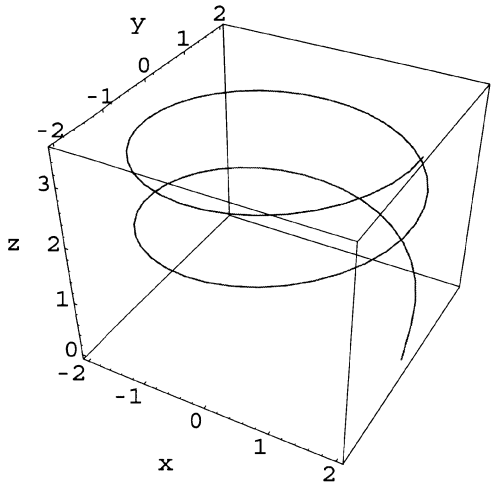
As discussed in Section 3.2, we define a tangent vector as a measure to determine the variation of a function in all its independent variables. Closely related to the tangent vector is a vector field. Let us assume that we have a smooth curve $\mathbf{C}$ on a manifold $\mathfrak{M}$ given in a parametric form by $\Phi : I \to M$, where I is a subinterval of $\mathbb{R}$. The local representation of the curve is thus given by the m coordinates of $\mathfrak{M}$ by $\Phi = (\Phi^1(t), \Phi^2(t), \ldots, \Phi^m(t))$, where t is a parameter. The tangent vector of the curve is given by the derivative with respect to the parameter t and is calculated by $\frac{d\Phi}{dt}$. An example for this notation is the three-dimensional spiral defined by the function

```
Φ = {2 Cos[t], 2 Sin[t], √t}
```

```
{2 Cos[t], 2 Sin[t], √t}
```

A graphical representation of this curve is created by

```
pl1 = ParametricPlot3D[Flatten[{Φ,
    RGBColor[1, 0, 0]}], {t, 0, 4 π},
  AxesLabel → {"x", "y", "z"}]
```



The tangent vector of the curve is just calculated with the definition by differentiating Φ with respect to the parameter t:

```
tang = ∂t Φ
```

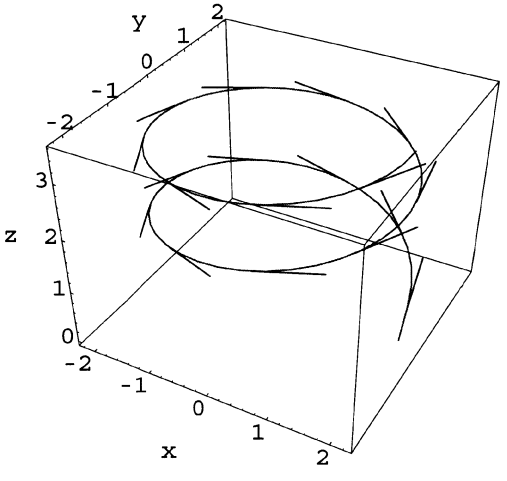
$$\left\{-2\,\mathrm{Sin}[t],\ 2\,\mathrm{Cos}[t],\ \frac{1}{2\sqrt{t}}\right\}$$

The derived tangent vector depends on the location on the curve. In the parametric representation, we also observe that the tangent vector is infinite in the origin and becomes smaller and smaller if t increases. If we plot the tangents on different locations along the curve, we get a field of vectors defining the vector field $\vec{v}$. The function Line[] allows us to graphically represent the vector field in connection with the curve:

```
vfield = Table[
  {RGBColor[0.000, 0.000, 1.000],
   Line[{Φ, Φ + tang/Sqrt[tang . tang]}]},
  {t, 0.1, 4 π, .7}];
```

The vector field and the curve are shown below. We combine the plots and the graphic primitives as follows:

```
Show[pl1, Graphics3D[vfield],
 PlotRange → All]
```



We observe that the vector field $\vec{v}$ of our spiral assigns a tangent vector to each point (x, y, z). The vector field itself varies smoothly from point to point. In Cartesian coordinates, the vector field of our spiral is given by

$$\vec{v} = \begin{pmatrix} -y \\ x \\ \frac{1}{2z} \end{pmatrix}. \tag{3.51}$$

This representation is based on the basis vectors of $\mathbb{R}^3$. In symmetry analysis of differential equations, it is convenient to replace the Cartesian basis by a representation using the partial differentiations with respect to the Cartesian coordinates. The partial derivatives with respect to the coordinates can be interpreted as placeholders for the Cartesian basis. Thus, we define a vector field as a differential operator in local coordinates as follows.

Definition: Vector field

A vector field $\vec{v}$ on a manifold $\mathfrak{M}$ is a tangent vector $\vec{v}_x$ to each point $x \in \mathfrak{M}$ varying smoothly from point to point. In local coordinates, a vector field has the representation

$$\vec{v}_x = \xi^1 \frac{\partial}{\partial x^1} + \xi^2 \frac{\partial}{\partial x^2} + \cdots + \xi^m \frac{\partial}{\partial x^m}, \tag{3.52}$$

where the ξ^i are smooth functions of the coordinates x. ○

The related *Mathematica* definition is thus given by

```
Clear[VectorField];
VectorField[coef_List,
 vars_List] := Block[{F, k},
  F = k@@vars; Sum[coef[[i]] ∂vars[[i]] F, {i, 1, Length[coef]}]]
```

Application in a three-dimensional manifold $\mathfrak{M}$ with coordinates (x, y, z) gives

```
var = {x, y, z};

coefficients = {v1@@var, v2@@var, v3@@var}

{v1[x, y, z], v2[x, y, z], v3[x, y, z]}

VectorField[coefficients, var] // LieTraditionalForm

v1 k_x + v2 k_y + v3 k_z
```

The differential operators are represented in our result by derivatives of an arbitrary function k with respect to the coordinates contained in the variable *var*. Let us look at this example from a more physical point of view, which was the original view of Lie. Assume that the components of the vector field are the components of a velocity field of a laminar fluid flow. Then, at each point (x, y, z), the vector $\vec{v}_x$ describes the velocity of the fluid particles passing through the point x. Thus, we are able to describe the velocity field of a fluid by using the mathematical term of a vector field.

Now, let us look at our example of the spiral in a different way. Knowing that the vector field describes the velocity field of the fluid, we may ask for the stream lines or potential representation of the flow. From a physical point of view, the vector field is connected with the flow if we consider the laminar behavior. To derive the potential representation of the flow of our example, we have to consider the components of the vector field as the defining components of the flow of the coordinates. The defining equations read

```
flowEquations =
 Thread[{∂ϵ x[ϵ], ∂ϵ y[ϵ], ∂ϵ z[ϵ]} == {-y[ϵ], x[ϵ], 1/(2 z[ϵ])}]

{x'[ϵ] == -y[ϵ], y'[ϵ] == x[ϵ], z'[ϵ] == 1/(2 z[ϵ])}
```

The right-hand sides of these flow equations are the components of the tangent vector or the vector field. The flow has to satisfy that the vector $\vec{x}$ at $t = 0$ is reproduced at the origin. The solution of these equations under the initial conditions $x(0) = x_0$, $y(0) = y_0$, $z(0) = z_0$ gives us the flow related to the vector field

```
flow = Simplify[
 DSolve[
   Join[flowEquations, {x[0] == x0, y[0] == y0, z[0] == z0}],
   {x[ϵ], y[ϵ], z[ϵ]}, ϵ]]
```

$$\{\{z[\epsilon] \to -\sqrt{z0^2 + \epsilon}\,,\ x[\epsilon] \to x0\,\mathrm{Cos}[\epsilon] - y0\,\mathrm{Sin}[\epsilon],$$
$$y[\epsilon] \to y0\,\mathrm{Cos}[\epsilon] + x0\,\mathrm{Sin}[\epsilon]\},\ \{z[\epsilon] \to \sqrt{z0^2 + \epsilon}\,,$$
$$x[\epsilon] \to x0\,\mathrm{Cos}[\epsilon] - y0\,\mathrm{Sin}[\epsilon],\ y[\epsilon] \to y0\,\mathrm{Cos}[\epsilon] + x0\,\mathrm{Sin}[\epsilon]\}\}$$

The result shows that the flow of our vector field is given by a rotation in the (x, y)-plane and a special translation along the z direction. This flow describes the spiral we started from in a different representation. We must remember that (x_0, y_0, z_0) is an arbitrary position of the three-dimensional space. This initial vector is transformed to another position if we change the parameter ϵ. This transformation acts like the flow in a fluid. Generalizing this example, we can define a flow of a vector field by the following:

Definition: Flow of a vector field

If $\vec{v}$ is a vector field, we call the integral curve passing through a point $\vec{x}$ in the manifold $\mathfrak{M}$ the flow $\Phi(\vec{x}, t)$ generated by $\vec{v}$. ○

The flow of a vector field has the properties

$$\Phi(\Phi(\vec{x}, \epsilon), \delta) = \Phi(\vec{x}, \epsilon + \delta), \qquad \vec{x} \in \mathfrak{M}, \tag{3.53}$$

meaning that the application of the flow on the same point $\vec{x}$ of $\mathfrak{M}$ at different values of ϵ results in the flow at $\vec{x}$ at a location of the sum in ϵ and δ. Another property the flow has to satisfy is the representation of $\vec{x}$ at the origin of ϵ; that is, at $\epsilon = 0$, we have the identity

$$\Phi(\vec{x}, 0) = \vec{x}. \tag{3.54}$$

Equation (3.54) describes the identity of the flow for a vanishing parameter ϵ reproducing the original vector $\vec{x}$. Comparing the two properties with the features discussed in Chapter 1 on groups, we see that the flow generated by a vector field $\vec{v}$ has some characteristics in common with groups. The derivation of the flow or the one-parameter group generated by a given vector field $\vec{v}$ is known as exponentiation of the vector field and represented by the notation

$$e^{\epsilon \vec{v}} = \Phi(\vec{x}, \epsilon). \tag{3.55}$$

In the following, we will denote the flow by $\Phi(\vec{x}, \epsilon)$. As already discussed, the flow is a result from the solution of a system of ordinary differential equations related to the

vector field $\vec{v}$. The vector field $\vec{v}$ is the generator; precisely, the infinitesimal generator of the transformation. Expanding the flow around $\vec{x} = 0$, we find the infinitesimal representation

$$\Phi(\vec{x}, \epsilon) = \vec{x} + \epsilon \vec{\xi}(\vec{x}) + O(\epsilon^2), \tag{3.56}$$

where $\vec{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)$ are the expansion coefficients of the vector field $\vec{v}$. The related determining equations of the flow also known as characteristic equations are given by

$$\frac{dx_i(\epsilon)}{d\epsilon} = \xi_i(\vec{x}(\epsilon)), \quad i = 1, 2, \ldots, n \tag{3.57}$$

with the initial condition $\vec{x}(\epsilon = 0) = \vec{x}$. Above, we discussed the spiral as an example in three dimensions $\mathfrak{M}=\mathbb{R}^3$. Several other examples will demonstrate the application of the theoretical considerations connecting the flow and the vector field on a manifold.

Example 1

Another example describing a one-dimensional translation in $\mathfrak{M}=\mathbb{R}$ is given by the vector field $\vec{v} = \partial_x$. The corresponding characteristic equation for the coordinate x reads

```
onedim = ∂ϵ x[ϵ] == 1
```

```
x′[ϵ] == 1
```

This equation was created by applying equation (3.57) with $\xi_1 = 1$. The solution of this simple first-order ODE under the initial condition $x(\epsilon = 0) = x$ representing the identity (3.54) is

```
flowOnedim = DSolve[{onedim, x[0] == x0}, x[ϵ], ϵ] /. x0 → x
```

```
{{x[ϵ] → x + ϵ}}
```

which is just a translation of the coordinate. We define the flow Φ in *Mathematica* by the relation

```
Φ[x_, ϵ_] := x + ϵ
```

Using this representation of the flow, we can check the properties (3.53) and (3.54). The combination of two translations ϵ and δ satisfy the condition

```
Φ[Φ[x, ϵ], δ] == Φ[x, ϵ + δ]
```

```
True
```

representing the closure relation of the group of translations. The second property (3.54), the identity of the group, gives

```
Ξ[x, 0] == x
```

```
True
```

Knowing the identity of the group, we are able to construct the inverse element of the group. The inverse element of the associated group follows from the relation

```
x == Ξ[InverseFunction[Ξ[x, ϵ]], ϵ]
 == Ξ[Ξ[x, -ϵ], ϵ]
```

```
ϵ + InverseFunction[x + ϵ] == x
```

```
Solve[ϵ + InverseFunction[x + ϵ] == x, InverseFunction[x + ϵ]]
```

```
{{InverseFunction[x + ϵ] → x - ϵ}}
```

The found solution represents an inverse translation if we assume that $\epsilon > 0$. The associativity of the underlying group follows from

```
Ξ[Ξ[Ξ[x, ϵ], δ], ω] ==
 Ξ[Ξ[Ξ[x, δ], ω], ϵ] ==
 Ξ[Ξ[Ξ[x, ω], ϵ], δ]
```

```
True
```

We demonstrated for the vector field $\vec{v} = \partial_x$, generating a translation in x, that the corresponding flow possesses all properties of a group. The symmetry of translation is one of those symmetries frequently encountered in symmetry analysis. □

Example 2

Another one-dimensional example in $\mathfrak{M} = \mathbb{R}$ also possessing the group properties is given by the vector field $\vec{v} = x\,\partial_x$. From equation (3.57), the corresponding characteristic equation is

```
scale = ∂ϵ x[ϵ] == x[ϵ]
```

```
x'[ϵ] == x[ϵ]
```

This equation allows the solution

```
flowScale = DSolve[{scale, x[0] == x0}, x[ϵ], ϵ] /. x0 → x
```

```
{{x[ϵ] → E^ϵ x}}
```

The result defines a scaling transformation of the variable x. We consider the factor E^{ϵ} as a constant greater than or less than 1 depending on the sign of ϵ. For positive ϵ, the original value of x is enlarged, and for negative values of ϵ, x is reduced in its value. We can check the two basic properties of the closure and the identity of the group by defining the flow as

```
Φ[x_, ϵ_] := x Exp[ϵ]
```

The closure of the scaling group now reads

```
Φ[Φ[x, ϵ], δ] == Φ[x, ϵ + δ]
```

```
True
```

and the identity is given by

```
Φ[x, 0] == x
```

```
True
```

The two relations demonstrate that the transformation group of scaling is closed and contains the identity transformation. □

Example 3

In this example, we will reverse our calculations. Knowing the flow, we will derive the related vector field. A global transformation commonly encountered in physics and mathematics is a rotation. To simplify things, let us consider the rotation of an object in the plane. The corresponding flow of this transformation is given by

```
Φ[x_, y_, ϵ_] := {x Cos[ϵ] - y Sin[ϵ], x Sin[ϵ] + y Cos[ϵ]}
```

Knowing the flow of a group, we are able to calculate the infinitesimal representation of the flow by calculating the vector field $\vec{v}$. According to equations (3.55) and (3.56), the infinitesimals ξ of the flow define the coefficients of the vector field. For the present example, in two dimensions the vector field has the representation

$$\vec{v} = \xi_1 (x, y)\, \partial_x + \xi_2 (x, y)\, \partial_y \,. \tag{3.58}$$

The infinitesimals are calculated by the relation

$$\xi = \frac{d\Phi}{d\epsilon}\Big|_{\epsilon=0} \,. \tag{3.59}$$

From the flow for rotations, we get the infinitesimals

```
ξ = ∂ϵ Φ[x, y, ϵ] /. ϵ → 0
```

```
{-y, x}
```

Thus, the vector field $\vec{v}$ of plane rotations has the representation $\vec{v} = -y\,\partial_x + x\,\partial_y$. Again, we can reverse our considerations and calculate the flow starting from the vector field. The flow or the global group transformation follow by solving the characteristic equations, which are a system of ordinary differential equations.

```
rotation = {∂ϵ x[ϵ] == -y[ϵ], ∂ϵ y[ϵ] == x[ϵ]}
```

```
{x'[ϵ] == -y[ϵ], y'[ϵ] == x[ϵ]}
```

The related flow follows by solving these equations:

```
flow = Simplify[
   DSolve[Join[rotation,
     {x[0] == x0, y[0] == y0}], {x[ϵ], y[ϵ]}, ϵ]] /.
  {x0 → x, y0 → y}
```

```
{{x[ϵ] → x Cos[ϵ] - y Sin[ϵ], y[ϵ] → y Cos[ϵ] + x Sin[ϵ]}}
```

reproducing the relation with which we started. □

Example 4

As a final example, let us consider the global group action of a flow containing rational expressions which are related to a projective group:

```
Φ[x_, y_, ϵ_] := {x/(1 + ϵ y x^2), y/(1 - ϵ y)}
```

Differentiating this expression by using equation (3.59), we find the infinitesimals of the flow to be

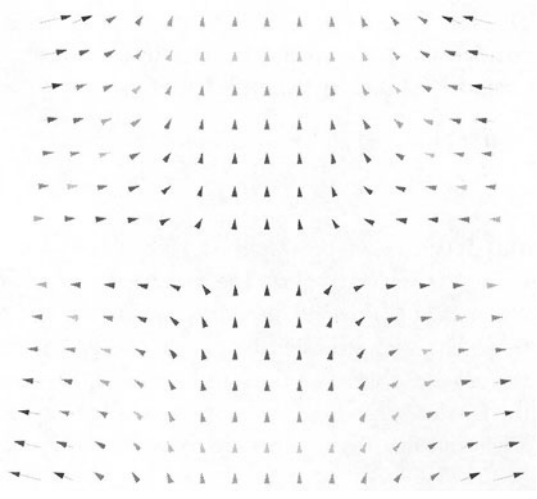
```
ξ = ∂ϵ Φ[x, y, ϵ] /. ϵ → 0
```

```
{-x^3 y, y^2}
```

which gives us the vector field in the representation $\vec{v} = -x^3\,y\,\partial_x + y^2\,\partial_y$. Knowing the infinitesimals, we are able to graphically represent the vector field. A plot of the vector field corresponding to the rational flow is given below. The package Graphics`PlotField` is useful for the representation of vector fields.

```
<< "Graphics`PlotField`"
```

```
PlotVectorField[ξ, {x, -2, 2},
 {y, -2, 2}, ColorFunction → Hue]
```



□

So far, the discussion of flows and the related vector fields was restricted to a set of independent coordinates. Let us now ask the question: What happens if we apply the flow concept to a function f depending on a set of independent coordinates? Let us assume that $\vec{v}$ is a vector field on the manifold $\mathfrak{M}$ and $f: \mathfrak{M} \to \mathbb{R}$ is a smooth function. Our intention is to get a formula describing the changes of f if we apply the transformation Φ on the independent variables. To simplify things and make them easier to read in *Mathematica*, we restrict our considerations to the case $\mathfrak{M}= \mathbb{R}$. In local coordinates, the vector field is thus given by $\vec{v} = \xi(x)\,\partial_x$. Now, let us examine the behavior of the function f if the flow Φ is applied on the independent variable x. After the transformation of the independent variables, we calculate the derivative of f with respect to the parameter ϵ:

```
Clear[Φ, ξ]
```

```
∂ε f[Φ[x, ε]]
```

```
f'[Φ[x, ε]] Φ^(0,1)[x, ε]
```

The result is an expression containing derivatives of f and Φ. From our considerations above, we know that the flow Φ at $\epsilon = 0$ has to represent the identity. We also know that the first derivative of the flow at ϵ=0 is a representation of the infinitesimals ξ. We use these conditions to define the transformation rule

```
rule1 = {Φ[x_, 0] :→ x, Φ^(0,1)[x_, 0] :→ ξ[x]}
```

```
{Φ[x_, 0] :→ x, Φ^(0,1)[x_, 0] :→ ξ[x]}
```

The second element of this list of delayed rules introduces an abbreviation for the coefficients of the vector field ξ. If we again evaluate the differentiation of the function f at $\epsilon = 0$ and use *rule1*, we find

```
∂ϵ f[Φ[x, ϵ]] /. ϵ → 0 /. rule1
```

```
ξ[x] f'[x]
```

This relation can be identified with the application of the vector field $\vec{v}$ on f; i.e., $\vec{v}\, f(x) = \xi(x)\, \partial_x f(x)$. The presented calculation shows that the notation for a vector field is generally useful for simplifying the representation of the infinitesimal flow. The one-dimensional example can be generalized to higher dimensions. The vector field $\vec{v}$ acts as a first-order partial differential operator on real functions f on $\mathfrak{M}$. On the other hand, we can expand the function f containing the transformed arguments in a Taylor series around $\epsilon = 0$. The result of this sort of calculation is

```
Series[f[Φ[x, ϵ]], {ϵ, 0, 1}] /. rule1
```

```
f[x] + ξ[x] f'[x] ϵ + O[ϵ]^2
```

representing the infinitesimal change of f under the flow generated by the vector field $\vec{v}$. We can summarize that the flow changes the function f in the following way:

$$f(\Phi(x, \epsilon)) = f(x) + \epsilon\, \vec{v}\, f(x) + O(\epsilon^2), \tag{3.60}$$

where $\vec{v}f$ gives the infinitesimal change in the function f under the flow generated by $\vec{v}$.

So far, we discussed the meaning of a vector field in a manifold $\mathfrak{M}$ of independent variables. We observed that the ensemble of the tangent vectors at different positions defines the vector field. We also introduced the vector field by replacing the Cartesian basis with a differential basis. The examination of a function under the action of the flow demonstrated that the transformed function is represented by the function itself and the infinitesimal change of the function caused by the application of the vector field on the function. Up to now, we assumed in our discussions that the variables in the manifold are independent of each other. Let us now assume that we have a manifold containing also some dependent variables. Here, the question arises of how to transform the derivatives contained in such a manifold. This is closely related to the problem of prolonging or extending a vector field.

We discussed in Section 3.4 how to extend or prolong a manifold. The procedure in a nutshell is that we add new coordinates, representing the derivatives, to the manifold. On the other hand, if we extend the manifold by new coordinates, we naturally have to extend the vector field by these new coordinates. We can write such an extension symbolically by

$$\mathrm{pr}^{(k)}\,\vec{v} = \xi_i\,\partial_{x_i} + \phi^{\alpha}\,\partial_{u^{\alpha}} + \phi'^{\alpha}\,\partial_{u_i^{\alpha}} + \cdots . \tag{3.61}$$

In expression (3.61), we divided the variables of the manifold $\mathfrak{M}$ into two sets: one known as independent variables $\vec{x} = (x_1, x_2, \ldots, x_p)$ and the other known as dependent variables $\vec{u} = (u^1, u^2, \ldots, u^q)$. ϕ'^{α} denotes the symbol of the transformed derivatives $u_i^{\alpha} = \partial u^{\alpha} / \partial x_i$. The difficulty with relation (3.61) at the moment is that we do not know how to calculate the coefficients ϕ'^{α} of the extension. However, to get a feeling of how derivatives change under a transformation, let us go back to the basics of calculus.

Assume that we have to examine the transformations of a curve $u = f(x)$ in one independent variable x and one dependent variable $u(x)$. The transformation of this curve is given by two rules defining the change of the original variables x and u to the new variables $X = \Xi(x, u)$ and $U = \Phi(x, u)$, respectively. In *Mathematica*, we define this transformation by the following set of rules:

```
Clear[U, X, Φ, Ξ];
transformation = {X → Function[{x, u}, Ξ[x, u[x]]],
  U → Function[{x, u}, Φ[x, u[x]]]};
```

where the transforming functions Ξ and Φ are given functions of the original variables x and u. Applying the transformation on the curve $u = f(x)$, a new representation $U = F(X)$ results. The variables U and X in the new representation depend on the variables of the old representation (x, u). Our intention is to examine the derivative of the curve in the new coordinates. The calculation of the derivative of U in this new coordinate system needs to take into account the changes of all the new variables. These changes are best represented if we use the total derivative to realize the derivative. The slope of the curve in the new coordinates is calculated at each point (x, u) of the old coordinate system by

```
U' = Dt[U[x, u]] / Dt[X[x, u]]; U' // LieTraditionalForm
```

$$\frac{\mathrm{Dt}[u]\,U_u + \mathrm{Dt}[x]\,U_x}{\mathrm{Dt}[u]\,X_u + \mathrm{Dt}[x]\,X_x}$$

The result shows that the derivative in the new coordinate system is a function of the old variables. If we explicitly express these dependencies by the transformations connecting the old and the new coordinates, we get

```
U' = U' /. transformation; U' // LieTraditionalForm
```

$$\frac{u_x\,\Phi_u + \Phi_x}{u_x\,\Xi_u + \Xi_x}$$

This is a basic result from calculus and fundamental for our further examinations. The formula above says that a transformed derivative itself becomes a function of the transformation in the new coordinates. The connecting link between our current examinations and the prolongation of the vector field are the transformations Ξ and Φ which represent the flow of the related vector field. The variable U' in our notation is nothing else than the representation of the symbol $\Phi^{[\alpha]}$. This symbol was introduced in relation (3.61) for the extended vector field. The difference between the considerations on vector fields and the representation of the *extension* in calculus is that the latter does not depend on a parameter ϵ. However, this dependence is not essential, as we will show in a moment. We can actually assume that the transformations Ξ and Φ depend on ϵ. The definition of these one-parameter transformations reads

```
transformation = {X → Function[{x, u}, Ξ[x, u[x], ϵ]],
  U → Function[{x, u}, Φ[x, u[x], ϵ]]};
```

These transformations do not change the previous result. The relations for the extended vector field are derived if we replace the original transformations by the ϵ-dependent transformations. Then, the derivative in new coordinates gets the form

```
U' = Dt[U[x, u]] / Dt[X[x, u]] /. transformation; U' // LieTraditionalForm
```

$$\frac{u_x \Phi_u + \Phi_x}{u_x \Xi_u + \Xi_x}$$

where ϵ is just a parameter in this expression. This representation of the transformation is a general transformation determined by the arbitrary functions Ξ and Φ. Lie demonstrated that this general transformation can be replaced by a much simpler version, the so-called infinitesimal transformation. We know from our considerations above that the infinitesimal transformations of the manifold $\mathfrak{M}$ are given by

```
infinitesimalTransformation =
 {X → Function[{x, u}, x + ϵ ξ[x, u[x]]],
  U → Function[{x, u}, u[x] + ϵ ϕ[x, u[x]]]};
```

Using these infinitesimal transformations of the variables, the first derivative in the new coordinate system becomes

```
U' = Dt[U[x, u]] / Dt[X[x, u]] /. infinitesimalTransformation;
U' // LieTraditionalForm
```

$$\frac{u_x + \epsilon\,(u_x\,\phi_u + \phi_x)}{1 + \epsilon\,(u_x\,\xi_u + \xi_x)}$$

Remembering the fact that in symmetry analysis, an infinitesimal representation is based on the linear part of the parameter ϵ, we are able to reduce this rational expression to a simpler form by expanding U' around $\epsilon = 0$

```
U' = Expand[Series[U', {ϵ, 0, 1}]]; U' // LieTraditionalForm
```

$$u_x + (u_x\,(-u_x\,\xi_u - \xi_x) + u_x\,\phi_u + \phi_x)\,\epsilon + O[\epsilon]^2$$

The result is that the derivative U' is given by the old derivative u' plus terms characteristic of the transformation. This expression represents the infinitesimal transformation of the first derivative depending on the derivatives of the infinitesimals ξ and ϕ for the independent and dependent variables. The representation of the prolonged vector field is thus given by

```
prolongation = AppendTo[infinitesimalTransformation,
  Uprime → Function[{x, u}, w] /. w → (Normal[U'] /. u[x] → u)]
```

```
{X → Function[{x, u}, x + ϵ ξ[x, u[x]]],
  U → Function[{x, u}, u[x] + ϵ ϕ[x, u[x]]],
  Uprime → Function[{x, u}, u'[x] + ϵ (u'[x] ϕ^(0,1)[x, u] +
        u'[x] (-u'[x] ξ^(0,1)[x, u] - ξ^(1,0)[x, u]) + ϕ^(1,0)[x, u])}
```

The variable *prolongation* contains the infinitesimal transformations for the independent and dependent variables x and u and the first prolongation ϕ' of this manifold. Knowing the infinitesimal transformations of the variables, we are able to write down the corresponding vector field. The once extended vector field thus becomes

```
vectorField[f_] := ξ[x, u] ∂x f + ϕ[x, u] ∂u f +
  Coefficient[Normal[U'], ϵ] ∂p f
```

The application of this function on an auxiliary function f depending on the variables (x, u, p) gives us the first extended vector field in its general form:

```
vectorField[f[x, u, p]] // LieTraditionalForm
```

$$\phi\,f_u + \xi\,f_x + f_p\,(u_x\,(-u_x\,\xi_u - \xi_x) + u_x\,\phi_u + \phi_x)$$

We changed the notation slightly by introducing the variable p for the first derivative in u. This substitution simplifies the representation and clarifies the fact that u_x is considered as another coordinate of the manifold $\mathfrak{M}$. The result shows that the first extended vector field depends in a characteristic way on the derivatives of the dependent variables as well as on the first derivatives of the flow components ξ and ϕ. Recalling the steps of the calculation for deriving the first extension of the vector field, we can go to the next order in the extension. The steps we needed in the calculation were as follows:

1. Replace the old differentials by the new differentials.
2. Use the infinitesimal representation of the transformations.
3. Expand the result around $\epsilon = 0$ up to first order.

The second extension of the vector field follows by using the same steps but incorporates the results of the first extension. The second extension is calculated by the formula

```
U'' =
 Normal[Series[∂x (Dt[U[x,u]] /. Dt[X[x,u]] infinitesimalTransformation) / ∂x X[x, u[x]] /.
    infinitesimalTransformation, {ϵ, 0, 1}]];
U'' // LieTraditionalForm
```

$$u_{x,x} + \epsilon\,(2\,(-u_x\,\xi_u - \xi_x)\,u_{x,x} + \phi_u\,u_{x,x} - u_x\,(\xi_u\,u_{x,x} + u_x\,\xi_{x,u} + u_x\,(u_x\,\xi_{u,u} + \xi_{x,u}) + \xi_{x,x}) + u_x\,\phi_{x,u} + u_x\,(u_x\,\phi_{u,u} + \phi_{x,u}) + \phi_{x,x})$$

The result of the calculation contains a large number of terms. However, looking at the first terms of the result, we observe that the second derivative, $u_{x,x}$, is altered by a sum of terms containing derivatives of the infinitesimals ξ and ϕ. The components of this expression are derivatives of the dependent variable and the flows ξ and ϕ up to second order. The twice extended vector field thus follows by

```
Clear[vectorField];
vectorField[f_] := ξ[x, u] ∂x f + ϕ[x, u] ∂u f +
  Coefficient[Normal[U'], ϵ] ∂p f +
  Coefficient[U'', ϵ] ∂q f
```

Applied to an auxiliary function, we get the expression

```
vectorField[f[x, u, p, q]] // LieTraditionalForm
```

$$\phi\,f_u + \xi\,f_x + f_p\,(u_x\,(-u_x\,\xi_u - \xi_x) + u_x\,\phi_u + \phi_x) + f_q\,(2\,(-u_x\,\xi_u - \xi_x)\,u_{x,x} + \phi_u\,u_{x,x} - u_x\,(\xi_u\,u_{x,x} + u_x\,\xi_{x,u} + u_x\,(u_x\,\xi_{u,u} + \xi_{x,u}) + \xi_{x,x}) + u_x\,\phi_{x,u} + u_x\,(u_x\,\phi_{u,u} + \phi_{x,u}) + \phi_{x,x})$$

In conclusion, the second extension of the vector field $\vec{v}$ follows from the first extension which was created using the infinitesimals itself. If we are interested in the third extension, we need the second and the first extension. In other words, the higher extensions of the vector field are recursively defined. This recursive definition was first observed by Lie and Engel [1888]. Today, the extensions are calculated by a general formula combining all the discussed steps in a nutshell. The prolongation formula in its modern form is given by

$$\mathrm{pr}^{(k)}\,\vec{v} = \vec{v} + \sum_{\alpha=1}^{q}\sum_{J}\phi_{\alpha}^{J}(x, u_{(k)})\,\frac{\partial}{\partial u_{J}^{\alpha}}. \tag{3.62}$$

The second summation in this expression extends to all multi-indices $J = (j_1, \ldots, j_l)$ with $1 \le j_l \le p$, $1 \le l \le k$. The kth expansion coefficients ϕ_{α}^{J} of the prolongation are recursively given by

$$\phi_{\alpha}^{J}(x, u_{(k)}) = D_J\left(\phi_{\alpha} - \sum_{i=1}^{p}\xi_i\,u_i^{\alpha}\right) + \sum_{i=1}^{p}\xi_i\,u_{J,i}^{\alpha}, \tag{3.63}$$

where $u_i^{\alpha} = \partial u^{\alpha}/\partial x_i$ and $u_{J,i}^{\alpha} = \partial u_J^{\alpha}/\partial x_i$. This step corresponds to the recursion discussed above. For a detailed discussion of the recursive prolongation formula, see, e.g., Bluman and Kumei [1989], Ibragimov [1985], and Olver [1986].

The problem of such a complicated recursive calculation of the prolongation is that for the kth-order calculation, we always need to know the $k-1$ previous results. If k is a large number, this can be very time- and memory-consuming if done by computer, not to mention the labor of a pencil calculation. Thus, we need a method which simplifies the calculation and makes it efficient for a computer. Actually, there exists a way to derive the extensions of a vector field much quicker.

The calculation of the prolongation of a vector field is simplified if we keep the following two points in mind. First, there exists a representation of the infinitesimals which simplifies the transformations. This representation is known as the characteristics. Second, the differentiation process of the prolongation can be eliminated by using the Fréchet derivative. The combination of these two tools provides us with a procedure to overcome the recursive definition of the prolongation.

Before discussing the implementation of the prolongation in *Mathematica*, let us briefly show the equivalence of both formulations. To fix terms, we call the first procedure the recursive prolongation formula and the second, the Fréchet prolongation. The first step to prove the equivalence of the two methods is the introduction of the characteristics of a vector field. The characteristic function of a general vector field is defined by (cf. Olver [1986])

$$Q_{\alpha} = \phi_{\alpha} - \sum_{i=1}^{p}\xi_i\,\frac{\partial u^{\alpha}}{\partial x_i}, \quad \alpha = 1, 2, \ldots, q. \tag{3.64}$$

If we assume that the characteristics Q depend on the dependent variables and its derivatives, we can write down a relation connecting the prolongation of a function Δ with the Fréchet derivative. The function Δ is a member of the extended manifold and depends on derivatives up to kth order. The vector field $\vec{v}_Q$ based on the characteristics Q allows us the inclusion of first-order derivatives of u in the transformation. The connection between the prolongation of $\vec{v}_Q$ and the Fréchet derivative is given by

$$\mathrm{pr}^{(k)}\,\vec{v}_Q(\Delta) = \mathfrak{D}_\Delta(Q)\,. \tag{3.65}$$

This relation follows from the definition of the prolongation of the vector field $\vec{v}_Q$

$$\mathrm{pr}^{(k)}\,\vec{v}_Q = \sum_{\alpha=1}^{p}\sum_{J} D_J\,Q_\alpha\,\frac{\partial}{\partial u_J^\alpha}, \tag{3.66}$$

where, in general, $Q_\alpha = Q_\alpha\,(u^{(k)})$ depends on derivatives up to order $k = 0, 1, \ldots$. If, in addition, we use the definition of the Fréchet derivative, introduced in Section 3.5, we are able to reproduce equation (3.65).

Relations (3.63) allows us to calculate the kth expansion coefficient of the prolongation by a total differentiation

$$\phi_\alpha^J = D_J\,Q_\alpha + \sum_{i=1}^{p}\xi_i\,u_{J,i}^\alpha\,. \tag{3.67}$$

Substituting this expression for the general representation of the prolongation formula (3.62) and rearranging terms in the sums, we get

$$\mathrm{pr}^{(k)}\,\vec{v} = \sum_{\alpha=1}^{q}\sum_{J} D_J\,Q_\alpha\,\frac{\partial}{\partial u_J^\alpha} + \sum_{i=1}^{q}\xi_i\left\{\frac{\partial}{\partial x_i}\sum_{\alpha=1}^{q}\sum_{J} u_{J,i}^\alpha\,\frac{\partial}{\partial u_J^\alpha}\right\}. \tag{3.68}$$

A comparison of the terms in curly brackets with the definition of the total derivative

$$D_i\,f = \frac{\partial f}{\partial x_i} + \sum_{\alpha=1}^{q}\sum_{J} u_{J,i}^\alpha\,\frac{\partial f}{\partial u_J^\alpha} \tag{3.69}$$

shows that we can replace the prolongation (3.62) by

$$\mathrm{pr}^{(k)}\,\vec{v} = \mathrm{pr}^{(k)}\,\vec{v}_Q + \sum_{i=1}^{p}\xi_i\,D_i, \tag{3.70}$$

where we used definition (3.66) to express the kth prolongation of $\vec{v}_Q$. If we now use relation (3.65), we can express the prolongation of a vector field $\vec{v}$ by the Fréchet derivative as

$$\mathrm{pr}^{(k)}\,\vec{v} \;=\; \mathfrak{D}_\Delta(Q) \;+\; \sum_{i=1}^{p} \xi_i\, D_i\,. \tag{3.71}$$

Thus, we demonstrated the equivalence of the two methods. Both procedures are available in the package *MathLie*. The advantage of relation (3.71) is that the prolongation of a vector field $\vec{v}$ is free of any recursion. If we replace the characteristics by their infinitesimal representations, we get a formula which contains all the necessary information:

$$\mathrm{pr}^{(k)}\,\vec{v} \;=\; \mathfrak{D}_\Delta(Q)\Big|_{Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i \frac{\partial u^\alpha}{\partial x_i}} \;+\; \sum_{i=1}^{p} \xi_i\, D_i\,. \tag{3.72}$$

Relation (3.72) seems very complicated at first glance. This formula is awkward if we do the calculation by hand. However, using a computer, (3.72) is just the expression we need. From a computational point of view, equation (3.72) contains simple operations. These operations are differentiation, summation, and a substitution. Each one of these operations is carried out by a computer very efficiently. Another advantage of calculating the prolongation of a vector field using (3.72) is the flexibility of its application. We can also use this formula to calculate the prolongation in connection with Lie-Bäcklund symmetries. This type of symmetries is discussed in Chapter 9.

The implementation of the prolongation follows formula (3.72) very closely. The following lines show how the prolongation is implemented in *MathLie*. The prolongation function is based on the function FrechetD[]. For the correct work of Prolongation[], it is thus necessary that the Fréchet derivative is also available. The function Prolongation[] is based on the differentiation and substitutions as represented in (3.72). The function itself needs three arguments. The first argument represents the function on which the prolongation is applied. The second and third arguments contain the sets of dependent and independent variables of the manifold. All three arguments are lists. The result is a general expression of the infinitesimals representing the kth prolongation. The order k of the prolongation is determined by the function itself. The symbolic names for the infinitesimals ξ_i and ϕ_α are also created by the function itself.

```
Clear[Prolongation];
Prolongation[equations_List,
 depend_List, independ_List] :=
```

```
Block[{Depend = {}, vars, test = {},
  eta = {}, subrule = {}, prol = {},
  prolong, mainrule, xyzt, wxc, uvw},
 Do[
  AppendTo[Depend,
   depend[[i]] @@ independ],
  {i, 1, Length[depend]}];
vars = Flatten[Join[independ, Depend]];
 Do[AppendTo[eta,
   phi[i] @@ vars - Sum[xi[j] @@ vars ∂independ[[j]] Depend[[i]], {j, 1, Length[independ]}]],
  {i, 1, Length[depend]}];
 Do[AppendTo[test,
   Unique["w$t"]],
  {i, 1, Length[Depend]}];
 mainrule = wxc → Function[xyzt, uvw];
 Do[AppendTo[subrule, mainrule /.
    {wxc → test[[i]],
     uvw → eta[[i]],
     xyzt → independ}],
  {i, 1, Length[eta]}];
 prolong = FrechetD[equations,
   depend, independ, test];
 prolong = Expand[prolong /. subrule];
 prolong = Apply[Plus, prolong, 1];
 Do[
  AppendTo[prol,
   Expand[prolong[[j]] +
     Sum[xi[i] @@ vars ∂independ[[i]] equations, {i, 1, Length[independ]}]]],
  {j, 1, Length[prolong]}];
 Flatten[prol]]
```

Now, we can use this function to check our interactive calculations given above. By applying the function Prolongation[] to an arbitrary function depending on one independent and one dependent variable and its derivative, we can check our calculations. Since the function Prolongation[] uses the Fréchet derivative in the calculation, it is not necessary to specify the highest order of derivatives. The function automatically detects the order of the derivative. The application of Prolongation[] to the simple example discussed above gives us

```
prolongation = Prolongation[{f[x, u[x], ∂x u[x]]}, {u}, {x}]
```

```
{u'[x] phi[1]^(0,1)[x, u[x]] f^(0,0,1)[x, u[x], u'[x]] -
  u'[x]^2 xi[1]^(0,1)[x, u[x]] f^(0,0,1)[x, u[x], u'[x]] +
  phi[1]^(1,0)[x, u[x]] f^(0,0,1)[x, u[x], u'[x]] -
  u'[x] xi[1]^(1,0)[x, u[x]] f^(0,0,1)[x, u[x], u'[x]] +
  phi[1][x, u[x]] f^(0,1,0)[x, u[x], u'[x]] +
  xi[1][x, u[x]] f^(1,0,0)[x, u[x], u'[x]]}
```

If we compare this result with the result obtained by the interactive calculation, we detect a complete equivalence. The result created by *Mathematica* is difficult to read. The prolongation becomes more readable if we apply the function LieTraditionalForm[] to the result of the above calculation.

```
prolongation // LieTraditionalForm
```

$$\{f_x\,\xi_1 + f_u\,\phi_1 - f_{u_x}\,u_x^2\,(\xi_1)_u - f_{u_x}\,u_x\,(\xi_1)_x + f_{u_x}\,u_x\,(\phi_1)_u + f_{u_x}\,(\phi_1)_x\}$$

The function LieTraditionalForm[] uses the variable TraditionalLieForm containing rules to transform dependent variables and their derivatives to a shorthand notation. This representation uses subscripts to denote differentiations and suppresses the arguments of the functions. The result of the transformation is shorter and contains the information in condensed form. The representation reminds one of the traditional mathematical notation but is not usable by *Mathematica.* An abbreviation for LieTraditionalForm[] is LTF[]. This function additionally represents the argument as equations in a table. Both functions deliver easy-to-read output but are inconsistent with *Mathematica*'s notation. However, we can solve this notational inconsistency for *Mathematica* by storing the result of Prolongation[] into the variable prolongation and suppressing the output. Afterward, we apply the rules TraditionalLieForm to that variable. In this way, we gain both a consistent representation in *Mathematica* and a condensed representation of the result. We will demonstrate this procedure by calculating higher-order prolongations. We can use the function Prolongation[] in a manner as simple as in the previous example. The result for the second extension of a vector field reads

```
secondProlongation =
 Prolongation[{f[x, u[x], ∂{x,1} u[x], ∂{x,2} u[x]]},
 {u}, {x}]; secondProlongation // LieTraditionalForm
```

$$\begin{aligned}\{&f_x\,\xi_1 + f_u\,\phi_1 - f_{u_x}\,u_x^2\,(\xi_1)_u - f_{u_x}\,u_x\,(\xi_1)_x + f_{u_x}\,u_x\,(\phi_1)_u + f_{u_x}\,(\phi_1)_x - \\ &3\,f_{u_{x,x}}\,u_x\,(\xi_1)_u\,u_{x,x} - 2\,f_{u_{x,x}}\,(\xi_1)_x\,u_{x,x} + f_{u_{x,x}}\,(\phi_1)_u\,u_{x,x} - \\ &f_{u_{x,x}}\,u_x^3\,(\xi_1)_{u,u} - 2\,f_{u_{x,x}}\,u_x^2\,(\xi_1)_{x,u} - f_{u_{x,x}}\,u_x\,(\xi_1)_{x,x} + \\ &f_{u_{x,x}}\,u_x^2\,(\phi_1)_{u,u} + 2\,f_{u_{x,x}}\,u_x\,(\phi_1)_{x,u} + f_{u_{x,x}}\,(\phi_1)_{x,x}\}\end{aligned}$$

Applying the transformation to the variable secondProlongation delivers the shorthand notation. Multiple differentiations of the dependent variables are denoted

by subscripts separated by commas. However, the variable secondProlongation contains the full *Mathematica* representation of the result. We can display the *Mathematica* expression by

```
secondProlongation
```

```
{u''[x] phi[1]^(0,1)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] -
  3 u'[x] u''[x] xi[1]^(0,1)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] +
  u'[x]^2 phi[1]^(0,2)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] -
  u'[x]^3 xi[1]^(0,2)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] -
  2 u''[x] xi[1]^(1,0)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] +
  2 u'[x] phi[1]^(1,1)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] -
  2 u'[x]^2 xi[1]^(1,1)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] +
  phi[1]^(2,0)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] -
  u'[x] xi[1]^(2,0)[x, u[x]] f^(0,0,0,1)[x, u[x], u'[x], u''[x]] +
  u'[x] phi[1]^(0,1)[x, u[x]] f^(0,0,1,0)[x, u[x], u'[x], u''[x]] -
  u'[x]^2 xi[1]^(0,1)[x, u[x]] f^(0,0,1,0)[x, u[x], u'[x], u''[x]] +
  phi[1]^(1,0)[x, u[x]] f^(0,0,1,0)[x, u[x], u'[x], u''[x]] -
  u'[x] xi[1]^(1,0)[x, u[x]] f^(0,0,1,0)[x, u[x], u'[x], u''[x]] +
  phi[1][x, u[x]] f^(0,1,0,0)[x, u[x], u'[x], u''[x]] +
  xi[1][x, u[x]] f^(1,0,0,0)[x, u[x], u'[x], u''[x]]}
```

The application of a prolonged vector field to a differential expression is used extensively in the symmetry analysis of differential equations. Let us demonstrate the application of this function by two examples.

Example 1

Assume that we know that the infinitesimal flow is given by a rotation in the plane:

```
flows = {xi[1] → Function[{x, y}, -y[x]],
  phi[1] → Function[{x, y}, x]};
```

We are interested in the behavior of the differential expression $\Delta = \partial^2 y(x)/\partial x^2$ under this kind of flow. First, we will calculate the prolongation of Δ by using the function Prolongation[].

```
prol2 = Prolongation[{∂{x,2} y[x]}, {y}, {x}];
prol2 // LieTraditionalForm
```

```
{-2 (ξ1)x yx,x - 3 yx (ξ1)y yx,x + (ϕ1)y yx,x - yx (ξ1)x,x -
  2 yx^2 (ξ1)x,y - yx^3 (ξ1)y,y + (ϕ1)x,x + 2 yx (ϕ1)x,y + yx^2 (ϕ1)y,y}
```

The result is an expression containing a combination of derivatives for the infinitesimals ξ_i and ϕ_α and the dependent variables. The prolongation of the vector field $\vec{v}$ is simplified if we insert the known representation of the flow

```
prol2 /. y[x] → y /. flows // LieTraditionalForm
```

```
{3 yx yx,x}
```

The result is that the prolongation of y'' under a rotation results into an expression which is closely related to the original expression Δ. We will later show that the result derived is a consequence of the symmetry group of the equation $y'' = 0$. □

Example 2

Another example invariant under the given flow is given by the following example. The differential expression Δ reads $\Delta = y''(1+y'^2)^{-3/2}$. The corresponding prolongation of Δ under the condition of the symmetry of rotation is

```
Simplify[Prolongation[
    {∂{x,2} y[x] (1 + (∂x y[x])^2)^-3/2}, {y}, {x}] /. y[x] → y /.
  flows]
```

```
{0}
```

The result shows that the prolongation of the expression $y_{xx}(1+y_x^2)^{-3/2}$ under an infinitesimal rotations vanishes. □

So far, we discussed the basic tools of symmetry analysis. The following chapters will show you how these tools are used to find the symmetries of functions and differential equations.

4

Symmetries of Ordinary Differential Equations

4.1. Introduction

Let us start with the following question. Suppose you have to solve an ordinary differential equation of second order like

```
equation1 = ∂{x,2} u[x] - (x - u[x]) ∂x u[x] == 0;
equation1 // LieTraditionalForm
```

$$-(-u+x)\,u_x + u_{x,x} == 0$$

How can we proceed to find the solution of this simple-looking equation? The first idea is to use *Mathematica* to solve the equation. If we apply DSolve[] to the equation, we get the answer

```
solution = DSolve[equation1, u, x]
```

```
DSolve[-(x - u[x]) u'[x] + u''[x] == 0, u, x]
```

After this dissatisfying result, you may check your knowledge as to whether the differential equation belongs to a class you know. Or you may try to find a transformation which will put the differential equation into a standard form. If you

are not successful, you perhaps look up a table of standard equations or you try to make some ansatz to find a solution. If none of your tasks solved the problem, you have to leave it unsolved. But you may have an uneasy feeling that there is a method you may have overlooked or of which you are unaware.

In this section, we will show you a procedure which perhaps solves your problem and which is simple in its application when *MathLie* is used as a tool. The method we will present is a rather ancient method invented at the end of the last century by the Norwegian mathematician Sophus Lie. He produced a tremendous work on symmetries which is applied not only to solve differential equations but also to fields like quantum mechanics, function theory, perturbation theory, etc. In this chapter, we restrict our considerations to ordinary differential equations. In Chapter 5, we will discuss partial differential equations, too.

The story of symmetry analysis started in the middle of the 19th century when Lie and Klein met in Berlin. Both mathematicians contributed a lot to the theory of symmetries. Lie invented his famous work to examine symmetries in connection with algebraic and differential equations. In his Göttinger program, Klein developed the discrete and algebraic parts of the application of symmetries on functions. Lie merged into the large field of differential equations which was very useful for classifying the differential equations in a new way. The theory developed by Lie is very laborious if done by hand. This was one of the reasons why the application of this theory disappeared for solving practical problems. Very few people used Lie's procedure to examine their differential equations. One of these was Birkhoff [1950] who in the 1950s applied the theory to hydrodynamic problems. In recent years, more and more attention was paid to the theory of Lie as one of the rare methods to deliver solutions, especially for non-linear differential equations. Today, Lie's procedure is accessible for a broad application if the computational power of computer algebra is used. The very extended algebraic calculations today are carried out by computers. In the past 20 years, there has been a tremendous increase of computer power and of the development of symbolic languages, allowing the problem to be tackled in an even simpler way. A summary of the development of symbolic programs was recently given by Hereman [1994, 1996]. One of these symbolic languages usable for the implementation of Lie's procedure is *Mathematica*. *Mathematica* with its powerful matching procedures is well fitted as a tool to carry out calculations used in Lie's theory.

Lie's main idea was that the symmetry properties of a differential equation can be used to solve the equation. How this works and how Lie's theory is used within *MathLie* will be discussed in the following sections.

4.2. Symmetry Transformations of Functions

Before we apply Lie's method to ordinary differential equations, we will briefly discuss symmetries in connection with functions. This section serves to introduce the main concepts of the theory occurring throughout the book.

4.2.1 Symmetries

One of the most remarkable discoveries of Lie in the theory of groups was the invariance of a function under some transformations. When dealing with differential equations, one very often tries to simplify the equation by an appropriate change of variables. This transformation generally involves both the independent and the dependent variables. In *Mathematica*, we can represent such transformations by the following list of rules:

```
rule1 = {x → Function[{x, u}, X[x, u]],
  u → Function[{x, u}, U[x, u]]}
```

```
{x → Function[{x, u}, X[x, u]], u → Function[{x, u}, U[x, u]]}
```

This kind of transformation involving the original independent and dependent variables (x,u) is usually called a point transformation, meaning that a point (x, u) of the manifold $\mathfrak{M}$ is transformed into another point (X, U). A point transformation takes only into account a change of the coordinates. The point transformations actually considered by Lie were transformations depending on at least one parameter ϵ. As we will see, the parameter ϵ is the parameter of the corresponding group. Thus, we call ϵ the group parameter. A one-parameter transformation is thus given by

```
rule2 = {x → Function[{x, u, ϵ}, X[x, u, ϵ]],
  u → Function[{x, u, ϵ}, U[x, u, ϵ]]}
```

```
{x → Function[{x, u, ϵ}, X[x, u, ϵ]],
 u → Function[{x, u, ϵ}, U[x, u, ϵ]]}
```

Such transformations have the following properties: They are invertible if the corresponding Jacobi determinant exists, repeated application yields a transformation of the same type, and the identity of the transformation for $\epsilon = 0$ exists. As we know from Chapter 2, these three properties are the basis of a Lie group. They can be summarized in the definition of symmetry transformations.

Definition: Symmetry transformation

A set G of transformations given by

$$x \to X(x, u, \epsilon),$$

$$u \to U(x, u, \epsilon)$$

is a one-parameter group if it contains the identical transformation $I = T_0$ and includes the inverse T_ϵ^{-1} and the composition $T_\epsilon \otimes T_\beta \in \mathsf{G}$. By a suitable choice of the group parameter ϵ, the main group property $T_\epsilon \otimes T_\beta \in \mathsf{G}$ can be written

$$T_\epsilon \otimes T_\beta = T_{\epsilon+\beta}, \tag{4.1}$$

that is,

$$X(X(x, u, \epsilon), U(x, u, \epsilon), \beta) = X(x, u, \epsilon + \beta), \tag{4.2}$$

$$U(X(x, u, \epsilon), U(x, u, \epsilon), \beta) = U(x, u, \epsilon + \beta). \tag{4.3}$$

In particular applications, the two conditions hold only for sufficiently small values of ϵ and β. There, we arrive at what is called a local one-parameter group G or an infinitesimal group. ○

These simple properties ensure that the transformations given in *rule2* form a one-parameter group of point transformations. A simple example to show how point transformations work can be given by considering the shift of a function. Mathematically, a shift is defined by $T_\epsilon f(x) = f(x + \epsilon)$, where T_ϵ represents the translation operator. A definition of such an operator in *Mathematica* reads

```
T[f_, ϵ_] := f /. x → x + ϵ
```

This definition assumes that f is a function of x and that x is replaced by $x + \epsilon$ in the argument of f. To demonstrate the properties stated above, we start with the verification of the identity transformation for $\epsilon = 0$:

```
T[f[x], 0]
```

```
f[x]
```

which demonstrates the existence of the identity transformation. The inverse transformation can be checked by creating a translation and the corresponding reverse by

```
T[T[f[x], ϵ], -ϵ]
```

```
f[x]
```

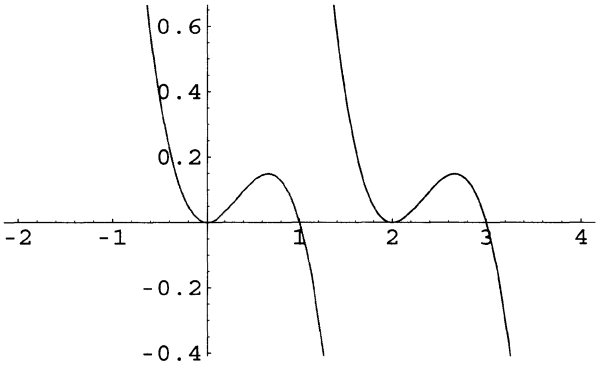
meaning that the inverse transformation is represented by a negative shift. The operation of translation and its inverse yield the original representation of the function. The closure of the transformation means that the function created by the transformation is again of the same type. This behavior can be demonstrated by the following specific example choosing $f(x) = x^2 - x^3$:

```
T[x^2 - x^3, -ϵ]
```

$$(x - \epsilon)^2 - (x - \epsilon)^3$$

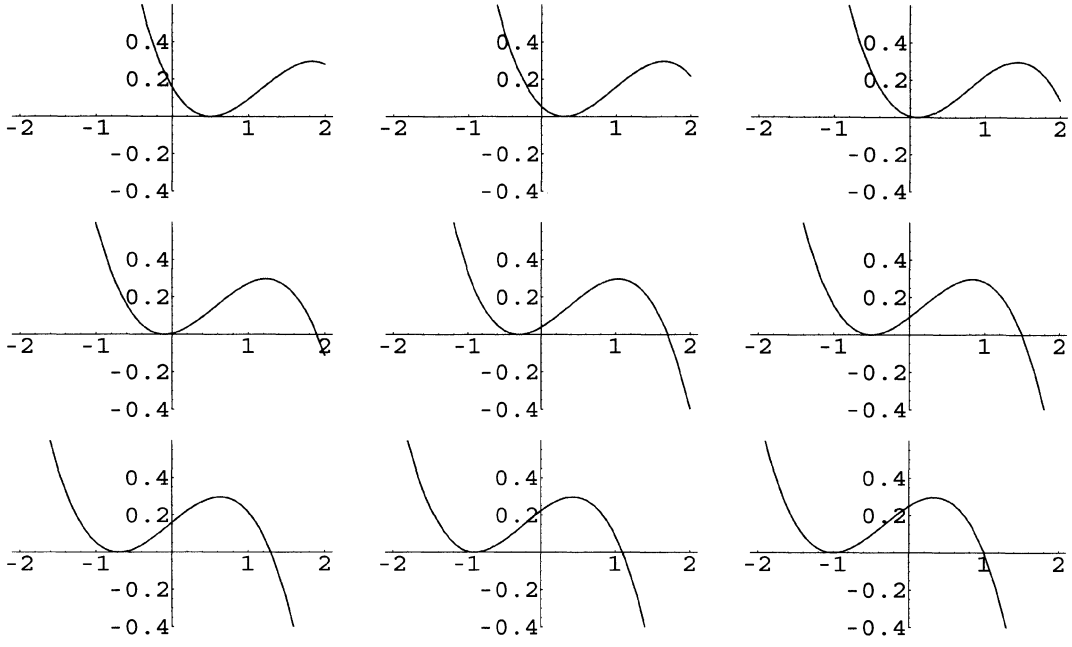
To demonstrate the action of the transformation, we will graphically represent the results for the identity and for a shift with $\epsilon = -2$.

```
Plot[Evaluate[{T[x^2 - x^3, 0], T[x^2 - x^3, -2]}],
 {x, -2, 4},
 PlotStyle → {RGBColor[1.000, 0.000, 0.000],
   RGBColor[0.000, 0.000, 0.000]}]
```



It is obvious from the figure that a shift by –2 in the argument translates the function by a distance of 2 to the right. Using the animation capabilities of *Mathematica*, we can readily demonstrate the shifting process by a small simulation. The process of shifting is demonstrated by the following animation where the parameter ϵ is varied from –1 up to 1 in steps of 1/20.

```
Do[Plot[Evaluate[T[x^2/2 - x^3/4, ϵ]],
   {x, -2, 2},
   PlotRange → {-.4, .6}, PlotStyle → {Hue[ϵ]}],
 {ϵ, -1, 1, 1/20}]
```



The sequence of graphs shows a continuous movement of the curve along the horizontal axis. This movement is created by the translation operator T_ϵ if we continuously change ϵ. Each value of ϵ is represented by a different color in the animation. Another property of our translation is the associativity of the transformation which can be formulated by

```
T[T[T[f[x], α], β], χ] == T[T[T[f[x], α], χ], β]

True
```

The result *True* states that the exchange of two of the three parameters α, β, and χ does not alter the final result. This specialty is not contained in the basic properties of a group. However, if a group satisfies the associativity, we call it an Abelian group.

Example 1

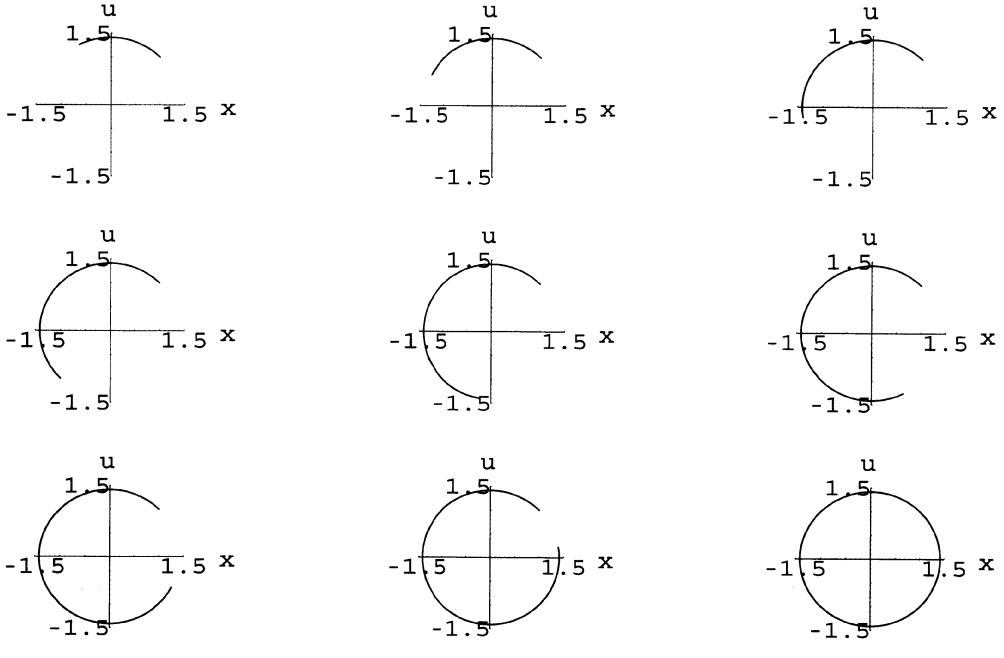
Another simple example of a one-parameter group is given by a rotation in the (x, u)-plane. This sort of point transformation is represented by the rules

```
rotation = {X → Function[{x, u, ϵ}, x Cos[ϵ] - u Sin[ϵ]],
   U → Function[{x, u, ϵ}, x Sin[ϵ] + u Cos[ϵ]]}
```

```
{X → Function[{x, u, ϵ}, x Cos[ϵ] - u Sin[ϵ]],
 U → Function[{x, u, ϵ}, x Sin[ϵ] + u Cos[ϵ]]}
```

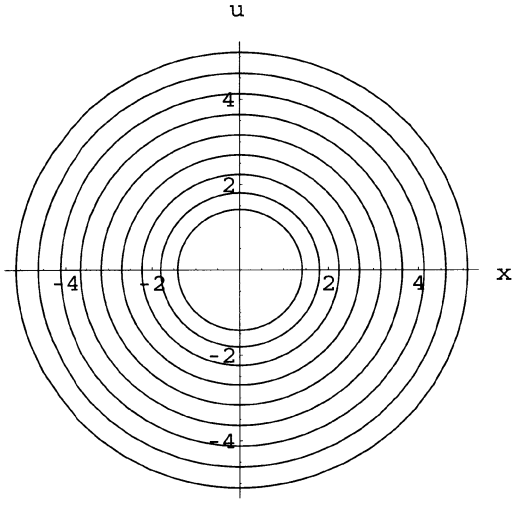
The one-parameter group defined in *rotation* and its action can be visualized as motion in an (x, u)-plane. To show the action, we take for $\epsilon = 0$ an arbitrary point (x_0, u_0) in the plane and follow the motion of the point when ϵ varies. The image of the initial point will move along some curve.

```
Table[ParametricPlot[Evaluate[
{X[x, u, ϵ], U[x, u, ϵ]} /. rotation /. {x → 1, u → 1}],
 {ϵ, 0, end},
AspectRatio → Automatic, PlotRange →
   {{-1.5, 1.5}, {-1.5, 1.5}}, PlotStyle → RGBColor[0, 0, 1],
 AxesLabel → {"x", "u"}],  {end, N[2 π/20], N[2 π], N[2 π/20]}]
```



The animation shows that an initial point (x_0, u_0) moves along a circle if we change the group parameter ϵ. By repeating this kind of transformation for different initial points, a picture representing the global action of the transformation in the (x, u)-plane is gained. The following picture contains the orbits of the transformation given in *rotation* for several initial points. The initial points are chosen along the x-axis.

```
ParametricPlot[Evaluate[
  Table[{X[x, u, e], U[x, u, e]} /. rotation /. {x → xo, u → 1},
    {xo, 1, 5, 1/2}]], {e, 0, 2 π},
 AspectRatio → Automatic, PlotStyle → Table[Hue[x],
    {x, 0, 1, 1/5}], AxesLabel → {"x", "u"}]
```



Once again, this picture shows that the given transformation will move a point along a circle if ϵ is varied. Each curve represents points that can be transformed into one another by the given transformation. □

4.2.2 Infinitesimal Transformations

One of Lie's essential findings was that a transformation as given above can be simplified. This simpler transformation is called infinitesimal transformation. The content of this idea is that it is sufficient to represent the transformation in its lowest approximation in ϵ, meaning that the finite transformation can be expanded in a

Taylor series around the identity transformation. In *Mathematica*, we have direct access to such an expansion of the transformation by applying the function Series[] to it. Let us again examine the general point transformation in the (x, u)-plane. We derive the infinitesimal representation by expanding the one-parameter transformation around $\epsilon = 0$ up to the first order in ϵ:

```
infinitesimalTrafo =
 Series[{X[x, u, ϵ], U[x, u, ϵ]}, {ϵ, 0, 1}]
```

```
{X[x, u, 0] + X^(0,0,1)[x, u, 0] ϵ + O[ϵ]^2,
 U[x, u, 0] + U^(0,0,1)[x, u, 0] ϵ + O[ϵ]^2}
```

If we use the group property of the identity $X(x, u, \epsilon = 0) = x$ and $U(x,u,\epsilon = 0) = u$, we can simplify the expression. The calculation shows that the transformation is represented by the identity plus some terms linear in ϵ. The coefficients of the parameter ϵ are called the infinitesimals of the transformation and are usually denoted by ξ and ϕ:

```
infinitesimalTrafo = TableForm[
  infinitesimalTrafo /. ({∂ϵ X[x, u, ϵ] → ξ[x, u],
     ∂ϵ U[x, u, ϵ] → ϕ[x, u],
     X[x, u, 0] → x,
     U[x, u, 0] → u} /. ϵ → 0)]
```

```
x + ξ[x, u] ϵ + O[ϵ]^2
u + ϕ[x, u] ϵ + O[ϵ]^2
```

This result for the infinitesimal representation of a transformation was summarized by Lie in his first theorem. The theorem considers the inverse problem of an infinitesimal representation. It treats the situation when the infinitesimals ξ and ϕ are known and asks for the global transformation.

Theorem: Lie's first theorem

There exists a parameter representation of a transformation such that the global transformation is equivalent to the solution of the initial value problem for the system of first-order differential equations

```
{∂ϵ X[ϵ] == ξ[X[ϵ], U[ϵ]],
  ∂ϵ U[ϵ] == ϕ[X[ϵ], U[ϵ]]} // TableForm
```

```
X'[ϵ] == ξ[X[ϵ], U[ϵ]]
U'[ϵ] == ϕ[X[ϵ], U[ϵ]]
```

with the initial conditions

```
{X[0] == x, U[0] == u} // TableForm
```

```
X[0] == x
U[0] == u
```

o

Using this theorem, the finite transformation represented by X and U is derived from the initial value problem by an integration with respect to ϵ. After the integration the parameter ϵ is eliminated.

Let us assume we know the infinitesimals ξ and ϕ. Then, we can apply Lie's first theorem in a straightforward way. Two examples will demonstrate the application of this theorem.

Example 1

As a first example, let us consider a scaling transformation. The infinitesimals of this kind of transformation are given by

```
ξ[x, u] = x;
```

and

```
ϕ[x, u] = -2 u;
```

The related defining equations for the global transformations are then

```
defequation = {∂ϵ X[ϵ] == ξ[x, u],
  ∂ϵ U[ϵ] == ϕ[x, u]} /. {x → X[ϵ], u → U[ϵ]}
```

```
{X'[ϵ] == X[ϵ], U'[ϵ] == -2 U[ϵ]}
```

The initial conditions for this system of equations are

```
initial = {X[0] == x, U[0] == u}
```

```
{X[0] == x, U[0] == u}
```

Combining these two sets of equations in a common list, we can solve the initial value problem using standard functions of *Mathematica*

```
eqin = Join[defequation, initial]
```

```
{X'[ϵ] == X[ϵ], U'[ϵ] == -2 U[ϵ], X[0] == x, U[0] == u}
```

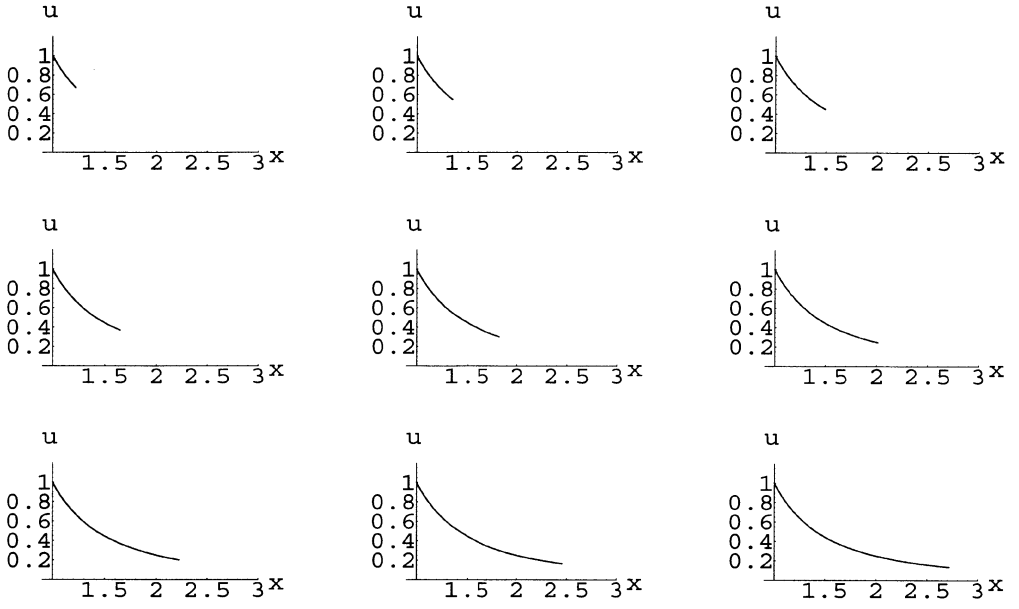
The solution of the initial value problem follows by applying DSolve[] to *eqin*

```
scalingtrafo = DSolve[eqin, {X, U}, ϵ]
```

```
{{X → (E^#1 x&), U → (E^-2 #1 u&)}}
```

The result is the representation of the scaling transformation for the variables x and u. We can represent the global transformation properties by plotting the new coordinates X and U if we change the group parameter ϵ. The following animation shows the action of the transformation:

```
Table[ParametricPlot[Evaluate[
{X[ϵ], U[ϵ]} /. scalingtrafo[[1]] /. {x → 1, u → 1}],
 {ϵ, 0, end},
AspectRatio → Automatic, PlotRange → {{0.9, 3}, {0, 1.2}},
 PlotStyle → RGBColor[0, 0, 1],
 AxesLabel → {"x", "u"}],  {end, 0.1, 1, .1}]
```



□

Lie's first theorem can be incorporated in a *Mathematica* function, allowing the determination of the global symmetry transformation. All solution steps used above are combined in this function. The function GlobalSymmetryTransformation[] will calculate the global symmetry transformation if the infinitesimals of the transformation are known. The input variables for this function are the infinitesimals ξ and ϕ. For the calculation, we also need the dependent and independent variables. The function is designed to calculate a general representation of the global transformation for an arbitrary number of independent and dependent variables.

```
Clear[GlobalSymmetyTransformation]
GlobalSymmetyTransformation[xi_List, phi_List,
 depend_List, independ_List] := Block[
  {vars, Vars, infini, dVars, equations,
   initial, sol},
  vars = Join[depend, independ];
  infini = Join[phi, xi];
  Vars = ToExpression/@ToUpperCase/@ToString/@vars;
  dVars = Table[Vars[[i]]@@{e}, {i, 1, Length[Vars]}];
  equations = Table[∂e dVars[[i]] == infini[[i]],
    {i, 1, Length[Vars]}] /. Thread[vars → dVars];
  initial = Thread[dVars == vars] /. e → 0;
  equations = Join[equations, initial];
  sol = DSolve[equations, Vars, e]]
```

The application of this function to the infinitesimals of Example 1 gives us the result:

```
GlobalSymmetyTransformation[{x}, {-2 u},
 {u}, {x}]
```

```
{{U → (E^(-2 #1) u&), X → (E^#1 x&)}}
```

which is identical with the result calculated above. Another example to check the function is the symmetry of rotation. From our earlier discussions, we know the global representation of a rotation. The infinitesimals for this kind of symmetry group are $\xi = -u$ and $\phi = x$. Applying the function GlobalSymmetryTransformation[] to these infinitesimals, we get

```
GlobalSymmetyTransformation[{-u}, {x},
 {u}, {x}]
```

```
{{U → (u Cos[#1] + x Sin[#1]&), X → (x Cos[#1] - u Sin[#1]&)}}
```

The result found is identical with the representation discussed in Section 4.2.1. It represents the rotation of an initial point (x_0, u_0) around the origin.

4.2.3 Group Invariants

In this section, a criterion of invariance is formulated which is useful in the application to ordinary and partial differential equations. A point transformation of an ordinary differential equation is a symmetry transformation if it maps solutions of the equation into solutions. We will show in this section that a symmetry transformation does not change the form of the differential equation. For the moment, we will restrict our considerations to the case in which only first-order derivatives are present. The invariance of higher-order differential equations will be discussed in

later sections. We know that a one-parameter group of transformations is represented by the infinitesimal transformations or the related tangent vector field $\vec{v}$. As the transformation is applied to an initial point, it moves along a path in such a way that the path maps into itself. This basic concept needs an analytical formulation. As mentioned in the previous section, it is always possible to represent the relations in a manifold by an algebraic expression. For the moment, we will thus restrict our considerations to curves in the plane.

Group invariants are basic quantities of a symmetry analysis. A group invariant is defined as follows:

Definition: Group invariant

A function $F(x, u)$ is an invariant of the group of transformations if

$$F(X(x, u, \epsilon), U(x, u, \epsilon)) = F(x, u) \tag{4.4}$$

identically in x and u for all values of the group parameter ϵ. ○

This definition is the basis for an invariant in symmetry analysis. It states that a function is an invariant if it has the same representation in the original and in the new coordinates. The definition also shows us a way to calculate the invariants F.

Curves in a plane can be represented by $F(x, u) = \text{const}$. As we know, this curve is said to be invariant under a transformation if the curve remains the same in both coordinate systems: the old and the new. The transformations are represented by the infinitesimal transformations due to Lie given by

```
Clear[ξ, ϕ];
infini = {X → Function[{x, u, ϵ}, x + ϵ ξ[x, u]],
  U → Function[{x, u, ϵ}, u + ϵ ϕ[x, u]]}
```

```
{X → Function[{x, u, ϵ}, x + ϵ ξ[x, u]],
 U → Function[{x, u, ϵ}, u + ϵ ϕ[x, u]]}
```

The expression of invariance is given in (4.4) and reads in terms of *Mathematica*,

```
invar = F[x, u] == F[X[x, u, ϵ], U[x, u, ϵ]]
```

```
F[x, u] == F[X[x, u, ϵ], U[x, u, ϵ]]
```

meaning that the form of the curve is the same in the old and new coordinates. The application of our transformation delivers the invariance condition in an implicit form:

```
ivarI = invar /. infini
```

```
F[x, u] == F[x + ϵ ξ[x, u], u + ϵ ϕ[x, u]]
```

The explicit representation of the invariance condition is obtained by a Taylor expansion of the equation of invariance with respect to the parameter ϵ:

```
Thread[Series[ivarI, {ϵ, 0, 1}], Equal] // LieTraditionalForm
```

$$F == F + (\phi F_u + \xi F_x)\, \epsilon + O[\epsilon]^2$$

If we examine this expression, we observe that the invariance condition is satisfied if all terms containing the infinitesimal parameter ϵ vanish. However, the first term linear in ϵ can be represented by the vector field $\vec{v}$. To show this relation, let us apply the tangent vector field to F. A definition of the function TangentVector[] for more than one independent and dependent variable follows below. So we get the infinitesimals in a subscripted form.

```
TangentVector[F[x, u], {u}, {x}] // LieTraditionalForm
```

$$F_x\, \xi_1[x, u] + F_u\, \phi_1[x, u]$$

The result is equivalent to the terms linear in ϵ. Now, if we assume that the application of the tangent vector field onto the curve has to vanish, we can conclude that all higher terms in ϵ containing multiple applications of the vector field $\vec{v}$ on the curve also have to vanish. Thus, a sufficient condition that a curve is invariant under an infinitesimal transformation is that the application of the tangent vector field to the curve must vanish.

Uncovering a practical realization of the group properties, let us formulate the following theorem. It is again Lie's first theorem in a different representation. The theorem allows us the derivation of a group invariant. The group invariant is the function for which we are looking. Lie himself was a great systems analyst who based his total work on three fundamental theorems. The first of these fundamental theorems reads:

Theorem: Group invariants

A function $F(x, u)$ in q independent $x = (x_1, x_2, \ldots, x_q)$ and p dependent variables $u = (u^1, u^2, \ldots, u^p)$ is an invariant if and only if it satisfies the partial differential equation

$$\sum_{i=1}^{q} \xi_i \frac{\partial F(x,u)}{\partial x_i} + \sum_{\alpha=1}^{p} \phi_\alpha \frac{\partial F(x,u)}{\partial u^\alpha} = 0, \tag{4.5}$$

where ξ_i and ϕ_α are the infinitesimals of a point transformation. This relation is also known as the characteristic equation of the tangent vector field

$$\vec{v} = \sum_{i=1}^{q} \xi_i \frac{\partial}{\partial x_i} + \sum_{\alpha=1}^{p} \phi_\alpha \frac{\partial}{\partial u^\alpha} \cdot \circ$$

It follows that every one-parameter group of point transformations in the plane has one independent invariant. This invariant can be taken to be the left-hand side of any first integral $J(x, u) = C$. The first integral follows by integration of the characteristic equation which are based on the tangent vector field. The determining equation for the integral reads

```
chareq = ∂x u[x] == ϕ[x, u[x]] / ξ[x, u[x]]; chareq // LieTraditionalForm
```

$$u_x == \frac{\phi}{\xi}$$

Then, any other invariant is a function of J.

Example 1

Let us consider a circle. We know that a circle is invariant under a rotation around its center. We already know that the tangent vector field of the group of a rotation is given by the operator

```
Trot[function_] := -u ∂x function + x ∂u function
```

A circle in the coordinates x and u is represented by

```
circle = x^2 + u^2 - const
```

```
-const + u^2 + x^2
```

The application of the operator Trot[] on the circle gives us

```
Trot[circle]
```

```
0
```

This result shows that the tangent vector field applied to a curve invariant under the given symmetry vanishes. Thus, each circle is mapped into itself. This condition is very useful in the determination of symmetries not only for curves but also for differential equations. □

Example 2

Before we consider differential equations, let us discuss the other non-trivial case when the mapping of the vector field does not vanish. In this example, we apply a transformation on a curve which is not mapped into itself but in another member of the same family of curves. To examine the behavior of such a case, let us again consider the transformations of rotations for the family of straight lines. Rays are represented by the left-hand side of the expression

```
rays = u/x - const
```

$$-\text{const} + \frac{u}{x}$$

The application of the vector field Trot[] on these lines gives us

```
r1 = Trot[rays]
```

$$1 + \frac{u^2}{x^2}$$

which represents a single ray with slope 1. A second application of Trot[] on the transformed rays results into

```
r2 = Trot[r1]
```

$$\frac{2\,u^3}{x^3} + \frac{2\,u}{x}$$

which is, again, a ray with slope 1 in a more or less complicated representation. Thus, we observe that the application of a tangent vector field on a curve can produce two types of results. First, a transformation of the curve into itself, and second, a transformation to a curve contained in the family of the curves. □

Example 3

In this example, we determine the invariant from the characteristic equation. Let us, again, discuss the symmetry group of rotation. The infinitesimal representation of this symmetry is

```
rot = {ξ → Function[{x, u}, -u], ϕ → Function[{x, u}, x]}
```

```
{ξ → Function[{x, u}, -u], ϕ → Function[{x, u}, x]}
```

The corresponding characteristic equation is thus

```
chareqr = chareq /. rot; chareqr // LieTraditionalForm
```

$$u_x == -\frac{x}{u}$$

which has the solution

```
sol = DSolve[chareqr, u, x]
```

$$\{\{u \to \left(-\sqrt{-\#1^2 - 2\,C[1]}\;\&\right)\},\ \{u \to \left(\sqrt{-\#1^2 - 2\,C[1]}\;\&\right)\}\}$$

The first integral follows by solving one of these expressions with respect to the constant $C[1]$:

```
solint = Flatten[Solve[(u[x] /. sol[[1]]) == u[x], C[1]]]
```

$$\{C[1] \to \frac{1}{2}\,(-x^2 - u[x]^2)\}$$

The result defines a circle of radius $\sqrt{-2\,C[1]}$. □

In the examples discussed, we used group invariants in a two-dimensional plane. As stated in the theorem, the relations are also useful in higher-dimensional space. In our discussions, we used the term tangent vector to represent an operator which is central in the theory of symmetries. In the following section, we will examine this operator in more detail.

4.2.4 Tangent Vector

A very useful concept closely related to the infinitesimals ξ and ϕ is the concept of a tangent vector $\vec{v}$. The tangent vector $\vec{v}$ is also called the infinitesimal generator of the transformation or tangent vector field or, in short, a vector field. We already discussed the term tangent vector field in Sections 3.2 and 3.7 in connection with a manifold $\mathfrak{M}$ of m independent variables. Here, we will extend the definition to a $q \times p$-dimensional manifold of q independent and p dependent variables. The tangent vector can be understood as a generator of the symmetry. The term generator indicates that repeated applications of the infinitesimal transformation will generate the finite or global transformation. This point of view for a vector field was stressed by Lie in his older papers (cf. Engel and Heegaard [1912] Vol. V, p. 2). Lie called the tangent vector the generator of the infinitesimal transformation. Today, this operator is called a tangent vector of a transformation, its definition is

Definition: Tangent vector field

A vector field on a manifold $\mathfrak{M}$ in $q \times p$ coordinates is a tangent vector $\vec{v}$ to each point $(x, u) \in \mathfrak{M}$ varying smoothly from point to point. In local coordinates, a vector field has the representation

$$\vec{v} = \sum_{i=1}^{q} \xi_i \frac{\partial}{\partial x_i} + \sum_{\alpha=1}^{p} \phi_\alpha \frac{\partial}{\partial u^\alpha}, \tag{4.6}$$

where the ξ_i and ϕ_α are smooth functions of the coordinates (x, u). ○

The functions ξ and ϕ are the infinitesimals of the related infinitesimal transformation. The present definition extends the previous definition to a manifold $\mathfrak{M}$ spanned by independent and dependent variables. Some authors call the vector field $\vec{v}$ a Lie symbol. Actually, this operator goes back to Lagrange, and, thus, as an equivalence, deserves the name Lagrange operator (cf. Kowalewski [1931]). We recognize that the vector field has an old tradition in mathematics. We already mentioned that Lie, in his lectures, used this kind of operator in connection with a hydrodynamic flow. He called the paths of the infinitesimal transformation $\vec{v}\, f$ the stream lines of the flow. The infinitesimal generator $\vec{v}$ is used to generate such flows.

The following function implements the definition of a vector field given in (4.6). The function TangentVector[] needs three arguments. The first argument specifies the function to which the vector field is applied. The second and third are lists containing the independent and dependent variables. The calculation of the vector field follows formula (4.6) very closely:

```
Clear[TangentVector];
TangentVector[function_, dependent_List,
 independent_List] := Block[{vars, xi, phi},
  vars = Join[independent, dependent];
  xi = Table[Apply[ξi, vars],
    {i, 1, Length[independent]}];
  phi = Table[Apply[φα, vars],
    {α, 1, Length[dependent]}];
  Length[independent]
        ∑          xi[[i]] ∂independent[[i]] function +
       i=1
    Length[dependent]
        ∑          phi[[α]] ∂dependent[[α]] function
       α=1
 ]
```

The result of this function is the general representation of the vector field applied to a given function depending on the variables x and u. Let us demonstrate the application for a 2×2 manifold. A general function f depending on the independent variables (x, y) and on the dependent variables (u, v) has the vector field

```
TangentVector[f[x, y, u, v], {u, v}, {x, y}] //
 LieTraditionalForm
```

$$f_x\,\xi_1[x, y, u, v] + f_y\,\xi_2[x, y, u, v] + f_u\,\phi_1[x, y, u, v] + f_v\,\phi_2[x, y, u, v]$$

The infinitesimals ξ_1, ξ_2, ϕ_1, and ϕ_2 are arbitrary functions. The function TangentVector[] allows us to calculate the general expression of the vector field for a given function f. If we know, on the other hand, the global symmetry transformations of the coordinates, we are able to derive the infinitesimals. Remember that the global transformation was a result of Lie's first theorem. If we invert this theorem, the infinitesimals of the $q \times p$-dimensional global transformations $(X(\epsilon), U(\epsilon))$ follow by

$$\xi_i = \left.\frac{dX_i(\epsilon)}{d\epsilon}\right|_{\epsilon=0}, \qquad i = 1, 2, 3, \ldots, q \tag{4.7}$$

and

$$\phi_\alpha = \left.\frac{dU(\epsilon)}{d\epsilon}\right|_{\epsilon=0}, \qquad \alpha = 1, 2, 3, \ldots, p. \tag{4.8}$$

These relations are identical with the defining equations stated in the first theorem of Lie. The side condition $\epsilon = 0$ in equations (4.7) and (4.8) guarantees that the initial conditions of Lie's theorem are satisfied. Thus, if we know the global representation of a point transformation, we are able to write down the corresponding vector field. How this works for specific transformations will be shown in the following.

To simplify the representation, we restrict our discussions to a two-dimensional manifold $\mathfrak{M}$ in x and u. This reduction of the $q \times p$-dimensional manifold allows us to create a graphical representation of the vector field. In the following examples, we will show how the vector field is calculated. The infinitesimals of the vector field will serve as components in the graphical representation. Let us start with the well-known group of rotations.

Example 1

The global representation of the group of *rotation* was discussed in an earlier example. The transformations for the coordinates x and u are defined in *rotation*. We can represent the transformed coordinates by

```
trans = {X[x, u, ϵ], U[x, u, ϵ]} /. rotation
```

$$\{x\,\text{Cos}[\epsilon] - u\,\text{Sin}[\epsilon],\ u\,\text{Cos}[\epsilon] + x\,\text{Sin}[\epsilon]\}$$

Using relations (4.7) and (4.8), we can derive the infinitesimals by

```
rotinfinitesimals = ∂ε trans /. ε → 0
```

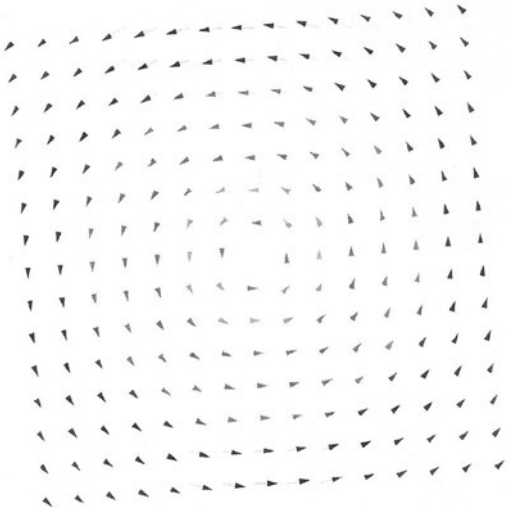
```
{-u, x}
```

The first element of this list represents the infinitesimal ξ and the second ϕ. The result is a list containing $-u$ and x as components of the tangent vector in the (x, u)-manifold. We can use this vector representation of the infinitesimal generator to graphically represent the vector field. Two-dimensional vector fields are plotted with a function contained in the package

```
<< "Graphics`PlotField`"
```

The function PlotVectorField[] allows us to plot the two-dimensional vector field

```
PlotVectorField[rotinfinitesimals, {x, -2, 2},
 {u, -2, 2}, ColorFunction → Hue]
```



As a result, we get a figure representing the vector field by vectors in the plane. We observe that vectors of the same color are tangential to a circle, thus the name tangent vector field. The vector field also contains information on the transformation properties. This information is stored in the arrangement of the arrows in the (x, u)-plane. We also gain an impression on the strength of the rotation on different locations of the (x, u)-plane by considering the lengths of the arrows. So a graphical representation of a vector field helps us to assess the behavior of a transformation. □

Example 2

Another frequently encountered example for a tangent vector field is the group of scaling. Let us assume that we know the global transformations of an inhomogeneous scaling of the coordinates x and u. A global scaling transformation is given by

```
scaling = {X → Function[{x, u, ϵ}, x E^ϵ],
           U → Function[{x, u, ϵ}, u E^(2 ϵ)]}
```

```
{X → Function[{x, u, ϵ}, x E^ϵ], U → Function[{x, u, ϵ}, u E^(2 ϵ)]}
```

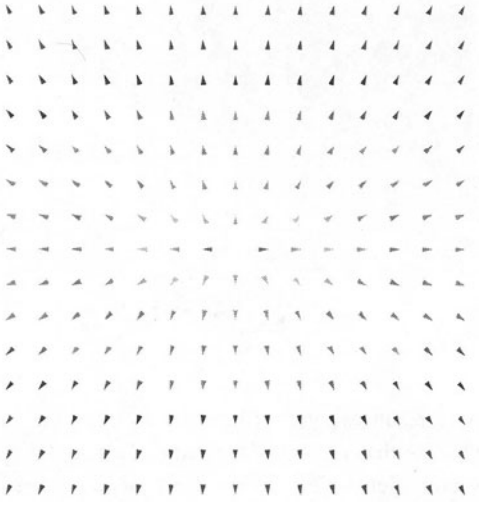
The infinitesimals of this transformation follow by using equations (4.7) and (4.8):

```
scalinginf = ∂ϵ {X[x, u, ϵ], U[x, u, ϵ]} /. scaling /. ϵ → 0
```

```
{x, 2 u}
```

Again, we get a vector of two components containing the infinitesimals ξ and ϕ of the transformation. The vector field of this transformation has the graphical representation

```
PlotVectorField[scalinginf, {x, -2, 2},
 {u, -2, 2}, ColorFunction → Hue]
```



showing us that, in both directions, the scales are changed if one moves from the center to a point of the rim. □

So far, we demonstrated some basic concepts of Lie's theory. In particular, we discussed the first theorem which deals with infinitesimal transformations. The concept of infinitesimal transformations is the basic tool in the derivation of symmetries of a differential equation and thus serves as a cornerstone in the solution of the equations. The presented infinitesimal generator or vector field is an essential part in the calculation of the infinitesimal transformations and is used to derive the symmetries from the equations. This behavior is related to the fact that the tangent vector is always a linear operator possibly creating a complicated form of the finite transformations. So far, we demonstrated transformations which only involved dependent and independent variables. However, if we examine differential equations, we have to extend or prolong the concept of transformations to derivatives as well.

4.2.5 Prolongation of Transformations

As discussed in Section 3.4, a prolongation is an extension of a transformation from the independent and dependent set of variables (x, u) to a space including the derivatives of the dependent variable (x, u, u'). This extension, for example, is necessary to examine the point symmetries of a first-order ordinary differential equation. In this section, we present the concept of prolongation in a three-dimensional space with coordinates (x, u, u') where $u' = p = \frac{du}{dx}$ is the slope of the given curve $u = u(x)$. Knowing how the curve in the plane transforms should enable us to calculate how u' transforms. To demonstrate the prolongation procedure, let us recall the results known from calculus. We assume for our examinations that the transformation of the plane into itself is given by the rules

```
rule5 = {X → Function[{x, u}, ξ[x, u[x]]],
  U → Function[{x, u}, ϕ[x, u[x]]]}
```

```
{X → Function[{x, u}, ξ[x, u[x]]],
 U → Function[{x, u}, ϕ[x, u[x]]]}
```

Our interest is to examine the transformation of an arbitrary curve and its derivatives. The functions ξ and ϕ representing the transformation are given functions of the variables x and u. The curve we will examine is given by the general expression

```
curve = u == f[x]
```

```
u == f[x]
```

which has its representation in the new coordinates by

```
newCurve = U == F[X]
```

```
U == F[X]
```

The slope of the curve in the new coordinates is calculated at each point (x, u) by the relation

```
Uprime = Dt[U[x, u]] / Dt[X[x, u]] /. rule5; Uprime // LieTraditionalForm
```

$$\frac{u_x\,\phi_u + \phi_x}{u_x\,\xi_u + \xi_x}$$

This formula expresses the derivative in new coordinates by the ratio of the total differentials of the transformed variables. Using the result for *Uprime*, we can define the first extended transformation in terms of ξ and ϕ by

```
rule6 = {X → Function[{x, u}, ξ[x, u[x]]],
  U → Function[{x, u}, ϕ[x, u[x]]],
  U' → Function[{x, u}, w] /. w → Uprime}
```

```
{X → Function[{x, u}, ξ[x, u[x]]],
 U → Function[{x, u}, ϕ[x, u[x]]],
 U' → Function[{x, u}, (u'[x] ϕ^(0,1)[x, u[x]] + ϕ^(1,0)[x, u[x]]) / (u'[x] ξ^(0,1)[x, u[x]] + ξ^(1,0)[x, u[x]])]}
```

Evidently, U' is the slope of the transformed curve at the point (X, U). The second extension of the transformation is calculated by using the same ideas. Using the result contained in *Uprime*, we are able to calculate the ratio of the total derivative of *Uprime* and X. Carrying out the calculation, we find

```
Udprime = ∂x Uprime / ∂x X[x, u[x]]; Udprime // LieTraditionalForm
```

$$\frac{1}{u_x\,X_u + X_x}\left(-\frac{(u_x\,\phi_u + \phi_x)\,(\xi_u\,u_{x,x} + u_x\,\xi_{x,u} + u_x\,(u_x\,\xi_{u,u} + \xi_{x,u}) + \xi_{x,x})}{(u_x\,\xi_u + \xi_x)^2} + \frac{\phi_u\,u_{x,x} + u_x\,\phi_{x,u} + u_x\,(u_x\,\phi_{u,u} + \phi_{x,u}) + \phi_{x,x}}{u_x\,\xi_u + \xi_x}\right)$$

This result allows us to prolong the transformation of the variables (x, u) up to second-order derivatives. *rule6* is extended by

```
AppendTo[rule6, U'' → Function[{x, u}, w] /.
  w → Udprime]; rule6 // LTF
```

$$U == \phi$$

$$X == \xi$$

$$U' == \frac{u_x\,\phi_u + \phi_x}{u_x\,\xi_u + \xi_x}$$

$$U'' == \left(-\frac{(u_x\,\phi_u + \phi_x)\,(\xi_u\,u_{x,x} + u_x\,\xi_{x,u} + u_x\,(u_x\,\xi_{u,u} + \xi_{x,u}) + \xi_{x,x})}{(u_x\,\xi_u + \xi_x)^2} + \frac{\phi_u\,u_{x,x} + u_x\,\phi_{x,u} + u_x\,(u_x\,\phi_{u,u} + \phi_{x,u}) + \phi_{x,x}}{u_x\,\xi_u + \xi_x}\right) \Big/ u_x\,X_u + X_x$$

rule6 contains the extended transformation of second order. We note that the transformation of the derivatives essentially depends on the structure of the finite transformations ξ and ϕ. The derivatives of the functions ξ and ϕ determine the slope in the new coordinates in a characteristic way. Going back to symmetry analysis, for examining differential equations we need the extensions of the infinitesimal generator represented by the vector field $\vec{v}$. To derive the relations describing the extended vector field, we consider transformations depending on the group parameter ϵ. The global transformation for the coordinates (x, u) read

```
rule7 = {X → Function[{x, u, ϵ}, ξ[x, u[x], ϵ]],
  U → Function[{x, u, ϵ}, ϕ[x, u[x], ϵ]]}
```

```
{X → Function[{x, u, ϵ}, ξ[x, u[x], ϵ]],
 U → Function[{x, u, ϵ}, ϕ[x, u[x], ϵ]]}
```

The extension formula is calculated in the same manner as in the parameter-free case by

```
Uprime = Dt[U[x, u, ϵ]] / Dt[X[x, u, ϵ]] /. rule7 /. Dt[ϵ] → 0;
Uprime // LieTraditionalForm
```

$$\frac{u_x\,\phi_u + \phi_x}{u_x\,\xi_u + \xi_x}$$

Since the group parameter ϵ is not a component of our manifold spanned by $(x, u, u', u'', ...)$, we have to assume that the variation of the group parameter vanishes; i.e., we set Dt[ϵ]→0. We simply use the fact that ϵ is a constant and thus its derivatives are zero. The first extended transformation is thus

```
rule7 = {X → Function[{x, u, ϵ}, ξ[x, u[x], ϵ]],
  U → Function[{x, u, ϵ}, ϕ[x, u[x], ϵ]],
  U' → Function[{x, u, ϵ}, w] /. w → Uprime
}
```

$$\left\{X \to \text{Function}[\{x, u, \epsilon\}, \xi[x, u[x], \epsilon]],\right.$$
$$U \to \text{Function}[\{x, u, \epsilon\}, \phi[x, u[x], \epsilon]],\ U' \to \text{Function}\Big[$$
$$\{x, u, \epsilon\},\ \frac{u'[x]\,\phi^{(0,1,0)}[x, u[x], \epsilon] + \phi^{(1,0,0)}[x, u[x], \epsilon]}{u'[x]\,\xi^{(0,1,0)}[x, u[x], \epsilon] + \xi^{(1,0,0)}[x, u[x], \epsilon]}\Big]\Big\}$$

In *rule7*, we get the global transformation of the once extended space. The second extension can be calculated in the same way as discussed above. Starting from a group $\mathbf{G}$ of point transformations and then adding the transformation of the first derivative, one obtains the group $\mathbf{G}_1$, which acts in the space of the three variables (x, u, u'). By further adding the transformation of the higher derivatives, one obtains the group $\mathbf{G}_2$ acting in the space (x, u, u', u'') and so on. The generalization of these arguments results in the definition of prolonged groups.

Definition: Prolonged groups

The groups $\mathbf{G}_1, \mathbf{G}_2, \ldots \mathbf{G}_n$ are termed the first, second, and nth prolongations of $\mathbf{G}$, respectively. ○

Actually, we are interested in the infinitesimal representation of the transformation. This sort of transformation is given by

```
infini = {X → Function[{x, u, ϵ}, x + ϵ ξ[x, u[x]]],
  U → Function[{x, u, ϵ}, u[x] + ϵ ϕ[x, u[x]]]}
```

```
{X → Function[{x, u, ϵ}, x + ϵ ξ[x, u[x]]],
 U → Function[{x, u, ϵ}, u[x] + ϵ ϕ[x, u[x]]]}
```

The representation of the first derivative in the new coordinates follows from

```
derivat = Dt[U[x, u, ϵ]] /. infini /. Dt[ϵ] → 0;
Dt[X[x, u, ϵ]]
derivat // LieTraditionalForm
```

$$\frac{u_x + \epsilon\,(u_x\,\phi_u + \phi_x)}{1 + \epsilon\,(u_x\,\xi_u + \xi_x)}$$

Since symmetry analysis in an infinitesimal representation is based on the linear dependence of the group parameter ϵ, we can restrict our considerations to first-order terms in ϵ. The infinitesimal part of the transformation follows by a Taylor expansion in ϵ around $\epsilon = 0$. The result is

```
infinider = Normal[Expand[Series[derivat,
    {ϵ, 0, 1}]]]; infinider // LieTraditionalForm
```

$$u_x + \epsilon\,(u_x\,(-u_x\,\xi_u - \xi_x) + u_x\,\phi_u + \phi_x)$$

The representation of the infinitesimal transformation of the slope shows that the old derivative is changed in the new coordinates by four parts, depending on derivatives of the infinitesimals ξ and ϕ. The representation of the prolonged transformation is thus given by

```
prolongation = {X → Function[{x, u, ϵ},
     x + ϵ ξ[x, u]],
  U → Function[{x, u, ϵ}, u[x] + ϵ ϕ[x, u]],
  U' → Function[{x, u, ϵ}, w] /.
   w → (infinider /. u[x] → u)}; prolongation // LTF
```

$$U == u + \epsilon\, \phi$$
$$X == x + \epsilon\, \xi$$
$$U' == u_x + \epsilon\, (u_x\, (-u_x\, \xi_u - \xi_x) + u_x\, \phi_u + \phi_x)$$

The second prolongation of the infinitesimal transformation is derived by

```
dderivat = ∂x derivat / ∂x X[x, u[x], ϵ] /. infini;
dderivat // LieTraditionalForm
```

$$\frac{1}{1 + \epsilon\, (u_x\, \xi_u + \xi_x)}$$
$$\Big(-(\epsilon\, (u_x + \epsilon\, (u_x\, \phi_u + \phi_x))\, (\xi_u\, u_{x,x} + u_x\, \xi_{x,u} + u_x\, (u_x\, \xi_{u,u} + \xi_{x,u}) + \xi_{x,x}))\, /$$
$$(1 + \epsilon\, (u_x\, \xi_u + \xi_x))^2 +$$
$$\frac{u_{x,x} + \epsilon\, (\phi_u\, u_{x,x} + u_x\, \phi_{x,u} + u_x\, (u_x\, \phi_{u,u} + \phi_{x,u}) + \phi_{x,x})}{1 + \epsilon\, (u_x\, \xi_u + \xi_x)}\Big)$$

and a Taylor expansion of the result up to first order

```
infinidder = Normal[Expand[Series[dderivat, {ϵ, 0, 1}]]];
infinidder // LieTraditionalForm
```

$$u_{x,x} + \epsilon\, (2\, (-u_x\, \xi_u - \xi_x)\, u_{x,x} + \phi_u\, u_{x,x} -$$
$$u_x\, (\xi_u\, u_{x,x} + u_x\, \xi_{x,u} + u_x\, (u_x\, \xi_{u,u} + \xi_{x,u}) + \xi_{x,x}) + u_x\, \phi_{x,u} +$$
$$u_x\, (u_x\, \phi_{u,u} + \phi_{x,u}) + \phi_{x,x})$$

The second-order prolongation is thus given by

```
AppendTo[prolongation,
 U'' → Function[{x, u, ϵ}, w] /.
  w → (infinidder /. u[x] → u)]; prolongation // LTF
```

$$U == u + \epsilon\, \phi$$
$$X == x + \epsilon\, \xi$$
$$U' == u_x + \epsilon\, (u_x\, (-u_x\, \xi_u - \xi_x) + u_x\, \phi_u + \phi_x)$$
$$U'' == u_{x,x} + \epsilon\, (2\, (-u_x\, \xi_u - \xi_x)\, u_{x,x} + \phi_u\, u_{x,x} -$$
$$u_x\, (\xi_u\, u_{x,x} + u_x\, \xi_{x,u} + u_x\, (u_x\, \xi_{u,u} + \xi_{x,u}) + \xi_{x,x}) + u_x\, \phi_{x,u} +$$
$$u_x\, (u_x\, \phi_{u,u} + \phi_{x,u}) + \phi_{x,x})$$

Now, we know the infinitesimal transformation rules allowing a change of the coordinates, including the transformation of the first and second derivatives. It becomes obvious from the representation of the infinitesimal transformation that only the infinitesimals ξ and ϕ have to be known to find an explicit representation of the transformation. Some examples will demonstrate the calculation of the prolongations.

Example 1

As a first example, let us again consider the group of rotation whose infinitesimals are given by

```
rotation = {ξ → Function[{x, u}, -u],
  ϕ → Function[{x, u}, x]}
```

```
{ξ → Function[{x, u}, -u], ϕ → Function[{x, u}, x]}
```

The related infinitesimal transformations for the variables and the derivatives are obtained in its explicit form for the group of rotation if we insert the infinitesimals into the prolongation formulas and the transformations

```
infirot = {X[x, u, ϵ], U[x, u, ϵ],
  U'[x, u, ϵ], U''[x, u, ϵ]} /. prolongation /.
  rotation; infirot // LieTraditionalForm
```

$$\{x - u\,\epsilon,\ u + x\,\epsilon,\ u_x + \epsilon\,(1 + u_x^2),\ u_{x,x} + 3\,\epsilon\,u_x\,u_{x,x}\}$$

Thus, the derivatives of a function are replaced by the derivative itself plus infinitesimal terms quadratic in the derivative of u. So far, we demonstrated the calculation of the prolongation by directly using the formulas derived above. In Section 3.7, we already discussed the general prolongation formula for a $q \times p$-dimensional manifold. The following example will recall the application of the function Prolongation[]. Be sure that the functions Prolongation[] and FrechetD[] are known by *Mathematica* for the following example. □

Example 2

The second example will serve to show how a vector field in 1×1 dimensions is calculated by using the function Prolongation[]. We assume that the infinitesimals of the group are known; e.g., an inhomogeneous scaling group with

```
iscaling = {ξ1 → Function[{x, u}, x],
            ϕ1 → Function[{x, u}, 3 u]}
```

```
{ξ1 → Function[{x, u}, x], ϕ1 → Function[{x, u}, 3 u]}
```

The vector field for a function f depending on a second derivative is thus given by

```
vectorfield = Prolongation[{f[x, u[x], ∂x u[x], ∂x,x u[x]]},
    {u}, {x}] /. u[x] → u /. iscaling;
vectorfield // LieTraditionalForm
```

$$\{f_x\,\xi_1 + f_u\,\phi_1 - f_{u_x}\,u_x^2\,(\xi_1)_u - f_{u_x}\,u_x\,(\xi_1)_x + f_{u_x}\,u_x\,(\phi_1)_u + f_{u_x}\,(\phi_1)_x - 3\,f_{u_{x,x}}\,u_x\,(\xi_1)_u\,u_{x,x} - 2\,f_{u_{x,x}}\,(\xi_1)_x\,u_{x,x} + f_{u_{x,x}}\,(\phi_1)_u\,u_{x,x} - f_{u_{x,x}}\,u_x^3\,(\xi_1)_{u,u} - 2\,f_{u_{x,x}}\,u_x^2\,(\xi_1)_{x,u} - f_{u_{x,x}}\,u_x\,(\xi_1)_{x,x} + f_{u_{x,x}}\,u_x^2\,(\phi_1)_{u,u} + 2\,f_{u_{x,x}}\,u_x\,(\phi_1)_{x,u} + f_{u_{x,x}}\,(\phi_1)_{x,x}\}$$

Contrary to the interactive calculation, we get the vector field as a scalar. The coefficients of the derivatives of f contain the information on the infinitesimals and their prolongations. The replacement of u[x] → u is necessary since we defined the infinitesimals as functions of x and u. The function Prolongation[], however, creates the infinitesimals depending on $u[x]$. The prolongation procedure discussed so far is usable to extend the space of variables to higher orders of derivatives as well. □

So far, we introduced basic concepts of symmetry analysis for functions. We discussed the term of a point transformation, an invariant, and the meaning of the vector field. These terms are not only useful for functions but also important in the examination of differential equations. The following sections will discuss these subjects in connection with ordinary differential equations.

4.3. Symmetry Transformations of Differential Equations

In this section, we collect the main tools for determining the symmetries of differential equations. We define the notion of symmetry for a differential equation and discuss the main properties of the symmetry group. Using these definitions, we interactively calculate the infinitesimal symmetries of differential equations. We also introduce the notion of canonical variables useful in deriving the solution of a differential equation. Let us first start with the definition of a symmetry group.

4.3.1 Definition of a Symmetry Group

Let G be a group of point transformations and let G_1, G_2, ... be its first, second, ... prolongation. Then, we define the symmetry of a differential equation as

Definition: Symmetry of a differential equation

A group $\mathbf{G}$ of point transformations is a symmetry group of an nth-order ordinary differential equation

$$\Delta\left(x, u(x), \frac{du}{dx}, \ldots, \frac{d^n u}{dx^n}\right) = 0 \tag{4.9}$$

or (4.9) admits $\mathbf{G}$ if the nth extended manifold $\mathfrak{M}^n$ is invariant with respect to the nth prolongation $\mathbf{G}_n$ of the group $\mathbf{G}$. ○

This definition actually contains the special case for first-order ordinary differential equations. For this special type of equation, we will discuss the application of the definition. A first-order differential equation

$$\Delta(x, u(x), u'(x)) = 0 \tag{4.10}$$

admits a group $\mathbf{G}$ if the once extended manifold $\mathfrak{M}^1$, the surface in the space x, u, u', is invariant with respect to the first prolongation $\mathbf{G}_1$ of $\mathbf{G}$. This means that equation (4.10) is invariant under the coordinate transformation $X = X(x, u, \epsilon)$ and $U = U(x, u, \epsilon)$. The invariance condition can be formulated as

$$\Delta(X, U, U') = \Delta(x, u, u'). \tag{4.11}$$

This kind of relation also holds for the general case of an nth-order equation. Lie demonstrated that the invariance condition of the differential equation has a direct consequence for the solutions. He summarized this behavior in a theorem which is one of the main properties of a symmetry group.

4.3.2 Main Properties of Symmetry Groups

Let us consider again the case of an nth-order ordinary differential equation. We represent the equation in such a way that $\frac{d^n u}{dx^n}$ is the left hand side of the general expression. The general equation reads

$$\frac{d^n u}{dx^n} = f(x, u(x), u'(x), \ldots, u_{(n-1)}(x)), \tag{4.12}$$

with a smooth function f depending on the derivatives up to $(n-1)$st order, $u_{(n-1)}$. The main property of a symmetry group first proved by Lie in 1891 is the following (cf. Scheffers and Lie [1891] p. 352, Theorem 1):

Theorem: Symmetry transformation

A group G is a symmetry group of an nth-order ordinary differential equation if and only if G converts any solution of the equation

$$\frac{d^n u}{dx^n} = f(x, u(x), u'(x), \ldots, u_{(n-1)}(x)) \tag{4.13}$$

into a solution of the same equation. ○

This theorem is one of the cornerstones of Lie's theory. It serves as the starting point for the calculation of the symmetries.

4.3.3 Calculation of the Infinitesimal Symmetries

We already mentioned that a sufficient condition for invariance is the vanishing of the extended tangent vector field applied to the differential equation, meaning that we use the condition $\mathrm{pr}^{(k)}\, \vec{v}(\Delta)$ as the defining equation for the infinitesimals ξ_i and ϕ_α. This condition follows from the invariance of the differential equations $\Delta = 0$ under the transformation of independent and dependent variables. The derivation of this invariance criterion follows from a similar calculation as presented for functions in Section 4.2.3. The invariance condition of the tangent vector field supplies a system of equations serving as the determining system for the infinitesimals. We will see that this system is linear but coupled in ξ_i and ϕ_α. As mentioned in Section 3.7, the invariance condition is closely related to the prolongation of the vector field.

Before discussing the application, let us describe the algorithm of constructing infinitesimal symmetries. For first-order equations, it is known that an infinite number of symmetries always exists (cf. Stephani [1989]). For special cases, however, we find so-called conformal symmetries which are a subset of the possible symmetries (cf. Olver [1986] and Hydon [1994]). Because of these difficulties, it is more convenient to start our discussion with second-order equations:

$$\Delta(x, u(x), u', u'') = 0. \tag{4.14}$$

Using the definition of the symmetry group in connection with the extended vector field,

```
Tangent[Δ_] := ξ[x, u] ∂x Δ + ϕ[x, u] ∂u Δ + ϕ(1)[x, u] ∂p Δ +
  ϕ(2)[x, u] ∂q Δ
```

where p and q are abbreviations for the first and second derivatives of u. The infinitesimal invariance criterion contained in equation (4.11) takes the form

$$\mathrm{pr}^{(k)}\, \vec{v}(\Delta)\,|_{\Delta=0} \;=\; 0\,. \tag{4.15}$$

This expression follows by expanding (4.11) around the identity $\epsilon = 0$ and taking into account the prolongation formulas for the derivatives. A detailed derivation of this formula can be found by Olver [1986] or Bluman and Kumei [1989]. The general equation (4.15) reduces for a second-order ODE to the relation

```
Tangent[Δ[x, u, p, q]] == 0 /. Δ[_] → 0 // LieTraditionalForm
```

$$\phi\, \Delta_u + \xi\, \Delta_x + \Delta_p\, \phi_1[x, u] + \Delta_q\, \phi_2[x, u] == 0$$

where ϕ_1 and ϕ_2 are the first and second components of the prolongation, respectively. They are computed via the prolongation formula given in Section 3.7. The equation derived is called the determining equation for the group admitted by the ordinary differential equation.

If the differential equation is written in the representation of (4.12)

```
equat = q - f[x, u, p]
```

```
q - f[x, u, p]
```

we can derive the explicit form of the determining equation by applying the twice extended vector field on this expression:

```
determining = Tangent[equat]; determining // LTF
```

$$-\phi\, f_u - \xi\, f_x - f_p\, \phi_1[x, u] + \phi_2[x, u] == 0$$

The remaining unknowns in this result are the first and second extensions ϕ_1 and ϕ_2. We are able to extract the representation of ϕ_1 and ϕ_2 from the variable *prolongation* calculated in the section on prolongation of transformations. Using the variables X, U, U', and U'' and searching for the coefficients of the group parameter ϵ, we end up with the infinitesimals ξ and ϕ and the first and second extension ϕ_1 and ϕ_2:

```
prol = Coefficient[{X[x, u, ϵ], U[x, u, ϵ],
    U'[x, u, ϵ], U''[x, u, ϵ]} /. prolongation, ϵ] /.
  u[x] → u; prol // LTF
```

```
ξ == 0
ϕ == 0
u_x (-u_x ξ_u - ξ_x) + u_x ϕ_u + ϕ_x == 0
2 (-u_x ξ_u - ξ_x) u_x,x + ϕ_u u_x,x -
  u_x (ξ_u u_x,x + u_x ξ_x,u + u_x (u_x ξ_u,u + ξ_x,u) + ξ_x,x) +
  u_x ϕ_x,u + u_x (u_x ϕ_u,u + ϕ_x,u) + ϕ_x,x ==
 0
```

For our calculations, we only need the expressions representing ϕ_1 and ϕ_2 given by the third and fourth element of the list *prol*. In the next step, we define two rules representing ϕ_1 and ϕ_2 by

```
rule1 = {ϕ_1 → Function[{x, u}, w],
  ϕ_2 → Function[{x, u}, v]} /.
  {w → prol[[3]], v → prol[[4]]}
```

```
{ϕ_1 → Function[{x, u}, u'[x] ϕ^(0,1)[x, u] +
     u'[x] (-u'[x] ξ^(0,1)[x, u] - ξ^(1,0)[x, u]) + ϕ^(1,0)[x, u]],
 ϕ_2 → Function[{x, u},
   u''[x] ϕ^(0,1)[x, u] + 2 u''[x] (-u'[x] ξ^(0,1)[x, u] - ξ^(1,0)[x, u]) +
    u'[x] ϕ^(1,1)[x, u] + u'[x] (u'[x] ϕ^(0,2)[x, u] + ϕ^(1,1)[x, u]) -
    u'[x] (u''[x] ξ^(0,1)[x, u] + u'[x] ξ^(1,1)[x, u] +
       u'[x] (u'[x] ξ^(0,2)[x, u] + ξ^(1,1)[x, u]) + ξ^(2,0)[x, u]) +
    ϕ^(2,0)[x, u]]}
```

These expressions are used in the determining equation to eliminate ϕ_1 and ϕ_2:

```
determ = determining /. rule1; determ // LTF
```

```
-ϕ f_u - ξ f_x - f_p (u_x (-u_x ξ_u - ξ_x) + u_x ϕ_u + ϕ_x) + 2 (-u_x ξ_u - ξ_x) u_x,x +
  ϕ_u u_x,x - u_x (ξ_u u_x,x + u_x ξ_x,u + u_x (u_x ξ_u,u + ξ_x,u) + ξ_x,x) +
  u_x ϕ_x,u + u_x (u_x ϕ_u,u + ϕ_x,u) + ϕ_x,x ==
 0
```

The above calculations deliver an expression containing the determining equations of the infinitesimals ξ and ϕ in an implicit way. The main characteristic of the result is the dependence on derivatives of u. However, we know that the equation *equat* has to be satisfied on the manifold $\mathfrak{M}$. This allows us to eliminate certain derivatives by using the equation itself. By replacing one type of derivative, we eliminate redundant information in the expression *determ*. The replacement is carried out by

```
determi = determ /. (Solve[equat == 0, q] /.
    q → ∂{x,2} u[x]); determi // LTF
```

$$-\phi\, f_u - \xi\, f_x + 2\, f\, (-u_x\, \xi_u - \xi_x) + f\, \phi_u - f_p\, (u_x\, (-u_x\, \xi_u - \xi_x) + u_x\, \phi_u + \phi_x) - u_x\, (f\, \xi_u + u_x\, \xi_{x,u} + u_x\, (u_x\, \xi_{u,u} + \xi_{x,u}) + \xi_{x,x}) + u_x\, \phi_{x,u} + u_x\, (u_x\, \phi_{u,u} + \phi_{x,u}) + \phi_{x,x} == 0$$

Here $f[x, u, p]$ is a known function. On the other hand, the infinitesimals ξ and ϕ are unknown functions of x and u. Thus, the expressions containing first-order derivatives of u are independent from each other. This independence creates a system of determining equations in the variables x and u. An additional feature of these equations is that we find more determining equations than unknown functions ξ and ϕ. Thus, the system of determining equations is overdetermined. Solving this system, we find the infinitesimal symmetries ξ and ϕ of the equation.

Example 1

Let us examine the infinitesimal symmetries of the second-order equation

```
dequ2 = ∂{x,2} u[x] + ∂x u[x] / x - Exp[u[x]]; dequ2 // LTF
```

$$-E^u + \frac{u_x}{x} + u_{x,x} == 0$$

We use here the function LTF[] to represent the equation in mathematcal index notation. The highly nonlinear ordinary differential equation of second order has a right-hand side f given by

```
subrule = f → Function[{x, u, p}, Exp[u] - p/x]
```

```
f → Function[{x, u, p}, Exp[u] - p/x]
```

The defining equation for the infinitesimals follows by inserting the rule for f into the expression *determi* and replacing the derivatives p by u'

```
detex1 = determi /. subrule /. {p → ∂x u[x]}; detex1 // LTF
```

$$-E^u\, \phi - \frac{\xi\, u_x}{x^2} + 2\, \left(E^u - \frac{u_x}{x}\right)\, (-u_x\, \xi_u - \xi_x) + \left(E^u - \frac{u_x}{x}\right)\, \phi_u + \frac{u_x\, (-u_x\, \xi_u - \xi_x) + u_x\, \phi_u + \phi_x}{x} - u_x\, \left(\left(E^u - \frac{u_x}{x}\right)\, \xi_u + u_x\, \xi_{x,u} + u_x\, (u_x\, \xi_{u,u} + \xi_{x,u}) + \xi_{x,x}\right) + u_x\, \phi_{x,u} + u_x\, (u_x\, \phi_{u,u} + \phi_{x,u}) + \phi_{x,x} == 0$$

The resulting expression is a third-degree polynomial in the derivatives of u'. Since the infinitesimals ξ and ϕ do not depend on the derivatives the determining equations decompose into the following three relations. These equations follow by setting the coefficients of the various powers of u' equal to zero

```
tab1 = Flatten[Table[Coefficient[Expand[detex1],
     (∂x u[x])^i], {i, 3, 1, -1}]]; tab1 // LTF
```

$$-\xi_{u,u} == 0$$

$$\frac{2\,\xi_u}{x} - 2\,\xi_{x,u} + \phi_{u,u} == 0$$

$$-\frac{\xi}{x^2} - 3\,E^u\,\xi_u + \frac{\xi_x}{x} - \xi_{x,x} + 2\,\phi_{x,u} == 0$$

The fourth equation free of any u' is derived by the following line

```
AppendTo[tab1,
 Expand[detex1 - Plus@@(tab1 . {(∂x u[x])^3,
       (∂x u[x])^2, ∂x u[x]})]];

tab1 = Flatten[tab1]; tab1 // LTF
```

$$-\xi_{u,u} == 0$$

$$\frac{2\,\xi_u}{x} - 2\,\xi_{x,u} + \phi_{u,u} == 0$$

$$-\frac{\xi}{x^2} - 3\,E^u\,\xi_u + \frac{\xi_x}{x} - \xi_{x,x} + 2\,\phi_{x,u} == 0$$

$$-E^u\,\phi - 2\,E^u\,\xi_x + E^u\,\phi_u + \frac{\phi_x}{x} + \phi_{x,x} == 0$$

These four equations are the defining equations for the infinitesimals ξ and ϕ. The next step in finding the symmetries is the solution of these equations. We do this step by step. If we integrate the first and second equation, we find the general expressions for the solution

```
rule2 = {ξ → Function[{x, u}, p[x] u + a[x]],
  ϕ → Function[{x, u}, (∂x p[x] + p[x]/x) u^2 +
      (2 (∂x a[x] - a[x]/x) + q[x]) u + b[x]]}
```

$$\left\{\xi \to \text{Function}[\{x, u\}, p[x]\,u + a[x]],\ \phi \to \text{Function}\left[\{x, u\}, \left(\partial_x p[x] + \frac{p[x]}{x}\right) u^2 + \left(2\left(\partial_x a[x] - \frac{a[x]}{x}\right) + q[x]\right) u + b[x]\right]\right\}$$

where p, q, a, and b are arbitrary functions of x. Substituting this result into the determining equations in *tab1*, we get a reduced set of determining equations connecting the arbitrary functions $a(x)$, $p(x)$, $b(x)$, and $q(x)$.

```
tab2 = Expand[tab1 /. rule2]; tab2 // LTF
```

```
True
```

$$\frac{4\,p}{x} == 0$$

$$-3\,E^u\,p + \frac{3\,a}{x^2} - \frac{5\,p\,u}{x^2} - \frac{3\,a_x}{x} + \frac{5\,u\,p_x}{x} + 2\,q_x + 3\,a_{x,x} + 3\,u\,p_{x,x} == 0$$

$$-b\,E^u + E^u\,q - E^u\,q\,u - \frac{2\,a\,u}{x^3} + \frac{p\,u^2}{x^3} - \frac{2\,a\,E^u}{x} + \frac{2\,a\,E^u\,u}{x} + \frac{2\,E^u\,p\,u}{x} -$$
$$\frac{E^u\,p\,u^2}{x} - 2\,E^u\,u\,a_x + \frac{2\,u\,a_x}{x^2} + \frac{b_x}{x} - E^u\,u^2\,p_x - \frac{u^2\,p_x}{x^2} +$$
$$\frac{u\,q_x}{x} + b_{x,x} + \frac{2\,u^2\,p_{x,x}}{x} + u\,q_{x,x} + 2\,u\,a_{x,x,x} + u^2\,p_{x,x,x} ==$$
$$0$$

The second equation shows us that the function $p(x)$ has to vanish identically to satisfy the equation. Using this fact in the representation of the infinitesimals, we can simplify the results to

```
rule3 = rule2 /. p → Function[x, 0]
```

$$\left\{\xi \to \text{Function}[\{x, u\}, \text{Function}[x, 0][x]\,u + a[x]],\ \phi \to \text{Function}\left[\{x, u\}, \left(\partial_x \text{Function}[x, 0][x] + \frac{\text{Function}[x, 0][x]}{x}\right) u^2 + \left(2\left(\partial_x a[x] - \frac{a[x]}{x}\right) + q[x]\right) u + b[x]\right]\right\}$$

These expressions can be used again to simplify the set of the determining equations

```
tab2 = Expand[tab1 /. rule3]; tab2 // LTF
```

```
True
True
```

$$\frac{3\,a}{x^2} - \frac{3\,a_x}{x} + 2\,q_x + 3\,a_{x,x} == 0$$

$$-b\,E^u + E^u\,q - E^u\,q\,u - \frac{2\,a\,u}{x^3} - \frac{2\,a\,E^u}{x} + \frac{2\,a\,E^u\,u}{x} -$$
$$2\,E^u\,u\,a_x + \frac{2\,u\,a_x}{x^2} + \frac{b_x}{x} + \frac{u\,q_x}{x} + b_{x,x} + u\,q_{x,x} + 2\,u\,a_{x,x,x} ==$$
$$0$$

A second glance at these equations shows us that a combination of x-dependent auxiliary functions a, b, and q occur in connection with u-dependent coefficients. Since the auxiliary functions do not depend on u, the determining equations decompose into another set of equations. Let us extract the equations from the terms containing factors like $u\,\text{Exp}[u]$ and $\text{Exp}[u]$.

```
coef1 = DeleteCases[Flatten[
   Table[Coefficient[
     tab2, Exp[u] u^i], {i, 1, 0, -1}] /. u → 0], 0]; coef1 // LTF
```

$$-q + \frac{2\,a}{x} - 2\,a_x == 0$$

$$-b + q - \frac{2\,a}{x} == 0$$

The remaining set of equations follow from the terms containing pure coefficients in u:

```
AppendTo[coef1,
 DeleteCases[Coefficient[
   tab2 /. Exp[u] → 0, u], 0]]; coef1 // Flatten // LTF
```

$$-q + \frac{2\,a}{x} - 2\,a_x == 0$$

$$-b + q - \frac{2\,a}{x} == 0$$

$$-\frac{2\,a}{x^3} + \frac{2\,a_x}{x^2} + \frac{q_x}{x} + q_{x,x} + 2\,a_{x,x,x} == 0$$

The last set of equations follows from those terms which are free of u:

```
AppendTo[coef1, DeleteCases[
  tab2 /. {Exp[u] → 0, u → 0}, 0]]; coef1 // Flatten // LTF
```

$$-q + \frac{2\,a}{x} - 2\,a_x == 0$$

$$-b + q - \frac{2\,a}{x} == 0$$

$$-\frac{2\,a}{x^3} + \frac{2\,a_x}{x^2} + \frac{q_x}{x} + q_{x,x} + 2\,a_{x,x,x} == 0$$

$$\frac{3\,a}{x^2} - \frac{3\,a_x}{x} + 2\,q_x + 3\,a_{x,x} == 0$$

$$\frac{b_x}{x} + b_{x,x} == 0$$

Thus, the complete set of determining equations reads

```
coef1 = Flatten[coef1]; coef1 // LTF
```

$$-q + \frac{2\,a}{x} - 2\,a_x == 0$$

$$-b + q - \frac{2\,a}{x} == 0$$

$$-\frac{2\,a}{x^3} + \frac{2\,a_x}{x^2} + \frac{q_x}{x} + q_{x,x} + 2\,a_{x,x,x} == 0$$

$$\frac{3\,a}{x^2} - \frac{3\,a_x}{x} + 2\,q_x + 3\,a_{x,x} == 0$$

$$\frac{b_x}{x} + b_{x,x} == 0$$

To find the most general solution of these equations, we start by solving the last equation of this set. This second-order equation in b has the solution

```
sol1 = Flatten[DSolve[Last[coef1] == 0, b, x]]
```

```
{b → (C[2] + C[1] Log[#1]&)}
```

As expected, the solution contains two integrating constants $C[1]$ and $C[2]$. Applying this solution to the remaining equations

```
coef2 = coef1 /. sol1; coef2 // LTF
```

$$-q + \frac{2\,a}{x} - 2\,a_x == 0$$

$$q - \frac{2\,a}{x} - C[2] - C[1]\,\text{Log}[x] == 0$$

$$-\frac{2\,a}{x^3} + \frac{2\,a_x}{x^2} + \frac{q_x}{x} + q_{x,x} + 2\,a_{x,x,x} == 0$$

$$\frac{3\,a}{x^2} - \frac{3\,a_x}{x} + 2\,q_x + 3\,a_{x,x} == 0$$

True

simplifies the expressions. We find determining equations for a and q. If we examine the equations, we realize that the second equation is a purely algebraic equation which can be solved either for a or q. We decide here to solve the equation for a:

```
sol2 = Flatten[Solve[coef2[[2]] == 0, a[x]] /. a[x] → w]
```

$$\left\{w \to -\frac{1}{2}\,x\,(C[2] + C[1]\,\text{Log}[x] - q[x])\right\}$$

For further use of this solution, we convert the result into a pure function.

```
sol21 = a → Function[x, w] /. sol2
```

$$a \to \text{Function}\left[x, -\frac{1}{2}\,x\,(C[2] + C[1]\,\text{Log}[x] - q[x])\right]$$

The application of this solution to the determining equations simplifies the equations to an overdetermined system in q:

```
coef3 = Simplify[coef2 /. sol21]; coef3 // LTF
```

$$-q + C[1] - x\,q_x == 0$$

True

$$\frac{2\,q_x}{x} + 4\,q_{x,x} + x\,q_{x,x,x} == 0$$

$$\frac{1}{2}\,(7\,q_x + 3\,x\,q_{x,x}) == 0$$

True

If we examine these equations, we observe that all three equations are connected by the derivatives of q. Solving the first equation with respect to q' and substituting the result into the rest gives us

```
hel1 = Solve[coef3[[1]] == 0, ∂x q[x]]
```

$$\left\{\left\{q'[x] \to -\frac{-C[1] + q[x]}{x}\right\}\right\}$$

```
coef3 = Flatten[coef3 /. hel1]; coef3 // LTF
```

True

True

$$-\frac{2\,(q - C[1])}{x^2} + 4\,q_{x,x} + x\,q_{x,x,x} == 0$$

$$\frac{1}{2}\left(-\frac{7\,(q - C[1])}{x} + 3\,x\,q_{x,x}\right) == 0$$

True

a system of two coupled equations which are connected by q''. The solution of the second equation with respect to q'' and the reinsertion of the result into the equations gives

```
hel2 = Flatten[Solve[coef3[[4]] == 0, ∂{x,2} q[x]]]
```

$$\left\{q''[x] \to -\frac{7\,(C[1] - q[x])}{3\,x^2}\right\}$$

```
coef3 = Flatten[Simplify[coef3 /. hel2]]; coef3 // LTF
```

True

True

$$\frac{22\,q - 22\,C[1] + 3\,x^3\,q_{x,x,x}}{3\,x^2} == 0$$

True

True

a single equation of third order in q. One solution of this remaining equation is given by a constant $q[x] = C[1]$. Thus, we can define q as

```
sol3 = q → Function[x, C[1]]
```

```
q → Function[x, C[1]]
```

We check this solution by

```
coef3 /. sol3
```

```
{0, 0, 0, 0, 0}
```

The infinitesimals ξ and ϕ are thus given by

```
infini = Simplify[{ξ[x, u], ϕ[x, u]} /.
   rule3 /. sol1 /. sol21 /. sol3]
```

$$\left\{-\frac{1}{2}\,x\,(-C[1]+C[2]+C[1]\,\text{Log}[x]),\ C[2]+C[1]\,\text{Log}[x]\right\}$$

Thus, the infinitesimals depend on two arbitrary parameters $C[1]$ and $C[2]$ representing the group parameters. In view of the linearity of the determining equations, the general solution can be represented as a linear combination of two independent solutions

```
nfini1 = infini /. {C[1] → 1, C[2] → 0}
```

$$\left\{-\frac{1}{2}\,x\,(-1+\text{Log}[x]),\ \text{Log}[x]\right\}$$

and

```
nfini2 = infini /. {C[1] → 0, C[2] → 1}
```

$$\left\{-\frac{x}{2},\ 1\right\}$$

This means that our original equation admits two linearly independent operators and that we have to consider a two-dimensional vector space with the basis given above. □

The example discussed shows that the derivation of the determining equations is very laborious when done interactively. Therefore, it is our goal to present a procedure which automatically delivers at least a prolongation of the equation.

We already know that the prolongation is related to an expansion of the infinitesimals. The actual connection is a special form of a derivative known as Fréchet derivative. The definition of a Fréchet derivative was introduced in Chapter 3. Here, we will shortly recall the definition in an appropriate form applicable on ordinary differential equations. A Fréchet derivative is a generalization of an ordinary derivative including a weight of the differential. The symbolic definition is

$$\mathfrak{D}_P(Q) = \frac{d}{d\epsilon} P(u + \epsilon\, Q(u))\,|_{\epsilon=0}\,. \tag{4.16}$$

In *MathLie* this definition is realized by the function FrechetD[].

The operational meaning of equation (4.16) is that the dependent variables are replaced by their variations in the support function P. The dependent variables are replaced by the variables and by the test function Q multiplied by ϵ. After the substitution a differentiation with respect to ϵ is carried out and finally we set $\epsilon = 0$. This relation defined in general for an r-dimensional support function P and for a q-dimensional test function Q allows a very efficient implementation in *Mathematica.*

In Section 3.7, we discussed the connection between the prolongation and the Fréchet derivative. This relation is given by

$$\mathrm{pr}^{(k)}\, \vec{v}(\Delta) = \mathfrak{D}_\Delta(Q) + \sum_{i=1}^{p} \xi_i\, D_i(\Delta) \tag{4.17}$$

where the test function Q is a combination of the infinitesimals ξ and ϕ.

$$Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i \frac{\partial u^\alpha}{\partial x_i} \qquad \alpha = 1, 2, \ldots, q\,. \tag{4.18}$$

The actual invariance condition for a given system of differential equations is then given by

$$\mathrm{pr}^{(k)}\, \vec{v}(\Delta)\Big|_{\Delta=0} = \left(\mathfrak{D}_\Delta(Q)\Big|_{Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i \frac{\partial u^\alpha}{\partial x_i}} + \sum_{i=1}^{p} \xi_i\, D_i(\Delta)\right)\Big|_{\Delta=0} = 0. \tag{4.19}$$

The algorithm for calculating the prolongation now consists of three steps. These steps are contained in equations (4.17)–(4.19). They are verbally expressed by the following:

1. Define the test functions Q using the infinitesimals as given in (4.18).
2. Calculate the Fréchet derivative and the complete derivative by equation (4.16).
3. Apply the side conditions (4.18) and the original equation to the result.

The three steps of the calculation for an ordinary differential equation are collected in the function ProlongationODE[]. The function ProlongationODE[] needs the equation, the dependent and independent variables, as input parameters. The function is written in such a form that only one dependent and one independent variable is allowed. The general definition of ProlongationODE[] is given in Section 3.7.

```
Clear[ProlongationODE];
ProlongationODE[equations_, dependent_,
 independent_] := Block[
  {vars, eta, testfunction, mainrule, prolong,
   ux, x, w},
  vars = Flatten[Join[
     {dependent@@{independent}}, {independent}]];
  eta = ϕ@@vars - ξ@@vars ∂independent dependent@@{independent};
  testfunction = Unique["w$x"];
  mainrule = ux → Function[x, w];
  prolong = FrechetD[{equations}, {dependent},
    {independent}, {testfunction}];
  prolong = prolong /. (mainrule /.
      {ux → testfunction, x → independent,
       w → eta});
  prolong = Expand[Apply[Plus, prolong, 1] +
     ξ@@vars ∂independent equations]]
```

The action of the function ProlongationODE[] is demonstrated by applying it to some examples.

Example 1

Let us consider the general ordinary differential equation

```
ode3 = ∂x u[x] - F[u[x], x]; ode3 // LTF
```

$$-F + u_x == 0$$

The function ProlongationODE[] actually only treats the left-hand side of the equation $\partial_x u - F(x, u) = 0$. The application of the function to the equation provides the following information:

```
prolode3 = ProlongationODE[ode3, u, x]; prolode3 // LTF
```

$$-\phi\, F_u - \xi\, F_x - u_x^2\, \xi_u - u_x\, \xi_x + u_x\, \phi_u + \phi_x == 0$$

The resulting expression contains derivatives of the arbitrary function F and the infinitesimals. We notice that the derivative of the dependent variable u occurs in different locations. We know that the first derivative of u can be expressed by the differential equation itself. In this way, we can replace the first derivative by F:

```
pode3 = prolode3 /. ∂x u[x] → F[u[x], x]; pode3 // LTF
```

$$-\phi\, F_u - \xi\, F_x - F^2\, \xi_u - F\, \xi_x + F\, \phi_u + \phi_x == 0$$

Thus, we eliminated the redundant information contained in the original differential equation. Since the prolonged vector field must vanish according to equation (4.19), we get conditions determining the infinitesimals. The arbitrary function F in our example is known for a certain type of equation. Thus, the partial differential equation for ξ and ϕ has solutions $\xi = \xi(x, u)$ and $\phi = \phi(x, u)$. The three steps of deriving the prolongation can be simplified in *Mathematica* by

```
Map[# == 0&, ProlongationODE[ode3, u, x]] /.
 Solve[ode3 == 0, ∂x u[x]] // LieTraditionalForm // TableForm
```

$$-\phi\, F_u - \xi\, F_x - F^2\, \xi_u - F\, \xi_x + F\, \phi_u + \phi_x == 0$$

With this expression, we are able to derive the invariance condition for any ordinary differential equation. A specific example may demonstrate the derivation of the invariance condition. □

Example 2

Let us assume that we have to find the invariance properties of the equation

```
ode4 = ∂x u[x] - g[u[x]] f[x];
Map[# == 0&, {ode4}] // LieTraditionalForm // TableForm
```

$$-f\, g + u_x == 0$$

where f and g are arbitrary functions of the independent and dependent variables, respectively. The invariance condition for this differential equation is given by

```
pode4 = ProlongationODE[ode4, u, x] /.
  Solve[ode4 == 0, ∂x u[x]]; pode4 // Flatten // LTF
```

$$-g\, \xi\, f_x - f\, \phi\, g_u - f^2\, g^2\, \xi_u - f\, g\, \xi_x + f\, g\, \phi_u + \phi_x == 0$$

A solution satisfying the invariance equation is given by

```
infiode4 = {ξ → Function[{u, x}, 1/f[x]],
   ϕ → Function[{u, x}, 0]}
```

$$\left\{\xi \to \text{Function}\left[\{u, x\}, \frac{1}{f[x]}\right], \phi \to \text{Function}[\{u, x\}, 0]\right\}$$

We can check this result directly by inserting the solutions into the invariance condition

```
pode4 /. u[x] → u /. infiode4
```

$$\{\{0\}\}$$

The result demonstrates that the given solutions satisfy the invariance condition. □

A special note on the arguments of functions is appropriate here. In the above calculations, we used the functions F, ξ, and ϕ depending on independent and dependent variables. In paper and pencil calculations, we are free to interchange these arguments because we know that these functions are the same, independent of the order of the arguments. However, in a symbolic calculation, we cannot change the slots of the variables, since a computer does not know how to handle the same function with interchanged arguments. So a good rule is to fix the arguments at the beginning of the calculation and to use the same order in all calculations.

In the above example, the infinitesimals are given. How these solutions are derived from the invariance condition will be discussed below. We saw in the example discussed that we can always express the invariance condition of first-order differential equations free of any derivative. One consequence of this observation is that a first-order differential equation actually has an infinite number of symmetries. This behavior can be read off directly from relation *pode3*, where ξ is connected by ϕ. However, there are exceptions to this statement where only a finite number of symmetries exist.

Example 3

The derivation of a finite number of symmetries can be best demonstrated with a higher-order differential equation. In the examples above we restricted our discussions to first-order differential equations. If one has to examine higher-order differential equations, we have to extend or prolong the tangent vector field to the order of the differential equation; e.g., to the second prolongation for second-order differential equations. To demonstrate the calculation let us examine the following general second-order equation

```
ode5 = ∂{x,2} u[x] - F[x, u[x], ∂x u[x]]; ode5 // LTF
```

$$-F + u_{x,x} == 0$$

The invariance condition (4.19) for this equation stays the same. The only difference is the higher order of differentiation which the function ProlongationODE[] detects by itself.

```
pode5 = ProlongationODE[ode5, u, x] /.
  Solve[ode5 == 0, ∂{x,2} u[x]]; pode5 // Flatten // LTF
```

$$-\phi\, F_u - \xi\, F_x - 3\, F\, u_x\, \xi_u + F_{u_x}\, u_x^2\, \xi_u - 2\, F\, \xi_x + F_{u_x}\, u_x\, \xi_x + F\, \phi_u - F_{u_x}\, u_x\, \phi_u - F_{u_x}\, \phi_x - u_x^3\, \xi_{u,u} - 2\, u_x^2\, \xi_{u,x} - u_x\, \xi_{x,x} + u_x^2\, \phi_{u,u} + 2\, u_x\, \phi_{u,x} + \phi_{x,x} == 0$$

From this relation we have to determine ξ and ϕ. The equation found is an identity in x, u and u_x. As a consequence of the point symmetries the infinitesimals are independent of u_x. This condition will split the general relation into several equations according to the different dependence of its parts on u'. We see that the same function ProlongationODE[] is useful to calculate the invariance relation (4.19) independent of the order of differentiation. In Section 4.4.2 on second-order differential equations we will show how we can extract the determining equations from such a relation. For the moment we stop at this point and discuss another useful tool called canonical variables.

4.3.4 Canonical Variables

In his work Lie pointed out that the introduction of suitable variables will drastically simplify the representation of a group. We discuss here the so-called canonical variables. On the other hand, canonical variables are a very efficient tool in the solution of ordinary differential equations. In this section we consider the distinguished situation of having a group consisting of two independent infinitesimal transformations.

Two infinitesimal transformations given by their vector fields $\vec{v}_1$ and $\vec{v}_2$ are independent from each other if the following relations do not exist

$$\vec{v}_1 = c\,\vec{v}_2, \tag{4.20}$$

where c is a constant, and the Lie product delivers

$$[\vec{v}_1, \vec{v}_2\,] = c_1\,\vec{v}_1 + c_2\,\vec{v}_2 \tag{4.21}$$

where c_1 and c_2 are constants again. The second relation can be simplified by assuming that the two independent transformations can be used to represent the product in a different way. If we assume that c_1 and c_2 are equal to zero, we get

$$[\vec{v}_1, \vec{v}_2] = 0. \tag{4.22}$$

On the other hand, we can assume that the infinitesimal transformations are given by a linear combination of the transformations by

$$[a_1\,\vec{v}_1 + a_2\,\vec{v}_2\,,\, b_1\,\vec{v}_1 + b_2\,\vec{v}_2\,] = 0, \tag{4.23}$$

where a_i and b_i, $i = 1, 2$, are constants. In these cases, the group is represented by commuting infinitesimal transformations. If, for example, $c_1 \neq 0$, we represent the two infinitesimal transformations by

$$\overrightarrow{\overline{v}}_1 = \vec{v}_1 + \frac{c_2}{c_1}\vec{v}_2 \tag{4.24}$$

and

$$\overrightarrow{\overline{v}}_2 = \frac{1}{c_1}\vec{v}_2 \,. \tag{4.25}$$

By using this representation, we can rewrite the above condition as

$$\left[\overrightarrow{\overline{v}}_1, \overrightarrow{\overline{v}}_2\right] = \frac{1}{c_1}[\vec{v}_1, \vec{v}_2] + \frac{c_2}{c_1^2}[\vec{v}_1, \vec{v}_2] = \frac{1}{c_1}(c_1\,\vec{v}_1 + c_2\,\vec{v}_2) = \overrightarrow{\overline{v}}_1 \,. \tag{4.26}$$

Thus, we can write down the following theorem

Theorem: Canonical variables

Each two-dimensional group of infinitesimal transformations $\vec{v}_1$ and $\vec{v}_2$ can be used to represent the product of the two transformations in each of the following forms:

$$[\vec{v}_1, \vec{v}_2] = 0 \tag{4.27}$$

or

$$[\vec{v}_1, \vec{v}_2] = \vec{v}_1 \,. \tag{4.28}$$

Each of the two results is independent from the other. ○

This theorem divides the two-dimensional groups in two classes. Each of these two classes can be divided into two subclasses.

Canonical variables now follow from the definition:

Definition: Canonical variable

Every one-parameter group of transformations reduces to the group of translations $\bar{t} = t + \epsilon$ and $\overline{w} = w$, with the vector field

$$\vec{v} = \partial_t \tag{4.29}$$

by a suitable change of variables $t = t(x, u)$ and $w = w(x, u)$. The variables t and w are the canonical variables. ○

The proof of this theorem follows from the fact that the tangent vector field in the original variables transforms according to the formula

$$\vec{v} = (\vec{v}\, t)\, \partial_t + (\vec{v}\, w)\, \partial_w \ . \tag{4.30}$$

In other words, canonical variables follow from the solution of the following linear partial differential equations. These two equations represent the invariance of the transformation

```
canonicalEquations = {ξ[x, u] ∂x t[x, u] +
    ϕ[x, u] ∂u t[x, u] == 1,
  ξ[x, u] ∂x w[x, u] + ϕ[x, u] ∂u w[x, u] == 0};
canonicalEquations // LieTraditionalForm // TableForm
```

$$\phi\, t_u + \xi\, t_x == 1$$
$$\phi\, w_u + \xi\, w_x == 0$$

The two equations follow from the definition by applying the tangent vector $\vec{v}$ to the canonical variables t and w

$$\vec{v}\, t = 1, \qquad \vec{v}\, w = 0. \tag{4.31}$$

The relations (4.24)–(4.26) define the canonical variables and present a way how these variables can be determined. The following examples discuss the derivation of canonical variables. First, we carry out a manual calculation of the canonical variables, and at the end of the section, we discuss an automatic procedure.

Example 1

To see how these concepts work in practical applications, let us examine a specific problem. We will examine a scaling symmetry represented by the vector field

```
veccan = {ξ → Function[{x, u}, x],
  ϕ → Function[{x, u}, -u]}
```

```
{ξ → Function[{x, u}, x], ϕ → Function[{x, u}, -u]}
```

For this example, the defining equations for the canonical variables reduce to

```
caneq = canonicalEquations /. veccan;
caneq // LieTraditionalForm // TableForm
```

$$-u\, t_u + x\, t_x == 1$$
$$-u\, w_u + x\, w_x == 0$$

These two linear partial differential equations are solved by using the function DSolve[] capable of solving first-order partial differential equations. The solution for t follows by

```
solc1 = DSolve[caneq[[1]], t[x, u], {x, u}]
```

```
{{t[x, u] → Log[x] + C[1][u x]}}
```

and the solution for w reads

```
solc2 = DSolve[caneq[[2]], w[x, u], {x, u}]
```

```
{{w[x, u] → C[1][u x]}}
```

The result for t logarithmically depends on x and allows an arbitrary function $C[1]$ combining the two old variables x and u by a product. The second canonical variable w is given by an arbitrary function which also connects x and u by a product. Since we are only interested in a special solution of the defining equations, we can simplify the arbitrary function by

```
r1 = C[1][u x] → x u
```

```
C[1][u x] → u x
```

For the inhomogeneous equation, we are mainly interested in a special solution. Thus, the general solution derived by the function DSolve[] reduces if we set the arbitrary function $C[1][x, u]$ to zero. Applying the reasoning to our solutions, we get

```
csol = Flatten[{solc1 /. C[1][x u] → 0,
   solc2 /. r1}]
```

```
{t[x, u] → Log[x], w[x, u] → u x}
```

We can check the derived result by inserting the solutions into the defining equations *caneq*. First, we have to transform the representation of the solutions into a pure function

```
r2 = Thread[{t, w} →
   (Function[{x, u}, #1]&) /@ ({t[x, u],
      w[x, u]} /. csol)]
```

```
{t → Function[{x, u}, Log[x]], w → Function[{x, u}, u x]}
```

The application of the result on the original equations gives us a list containing *True* for each equation:

```
Simplify[caneq /. r2]
```

```
{True, True}
```

This result shows that the solutions given in *csol* satisfy the defining equations for the canonical variables. □

Example 2

Another example of great interest is the group of rotations. For this kind of symmetry group, the infinitesimals are given by

```
veccan = {ξ → Function[{x, u}, -u],
  ϕ → Function[{x, u}, x]}
```

```
{ξ → Function[{x, u}, -u], ϕ → Function[{x, u}, x]}
```

The equations determining the canonical variables are thus given by

```
caneq = canonicalEquations /. veccan;
caneq // LieTraditionalForm // TableForm
```

$$x\,t_u - u\,t_x == 1$$
$$x\,w_u - u\,w_x == 0$$

The solution for the new independent variable t follows from

```
solct = DSolve[caneq[[1]], t[x, u], {x, u}]
```

$$\left\{\left\{t[x, u] \to -\text{ArcTan}\left[\frac{x}{\sqrt{u^2}}\right] + C[1]\left[\frac{1}{2}\left(-u^2 - x^2\right)\right]\right\},\right.$$
$$\left.\left\{t[x, u] \to \text{ArcTan}\left[\frac{x}{\sqrt{u^2}}\right] + C[1]\left[\frac{1}{2}\left(-u^2 - x^2\right)\right]\right\}\right\}$$

The new dependent variable w is determined by

```
solcw = DSolve[caneq[[2]], w[x, u], {x, u}]
```

$$\left\{\left\{w[x, u] \to C[1]\left[-\frac{u^2}{2} - \frac{x^2}{2}\right]\right\}\right\}$$

Again, we are only interested in a special solution which follows from

```
solct = PowerExpand[solct[[2]] /. C[1][_] → 0]
```

$$\left\{t[x, u] \to \text{ArcTan}\left[\frac{x}{u}\right]\right\}$$

The solution for w is extracted with

```
solcw = Flatten[solcw /. C[1][x_] → x]
```

$$\left\{w[x, u] \to -\frac{u^2}{2} - \frac{x^2}{2}\right\}$$

The complete expression of the canonical variables are thus

```
r3 = Thread[{t, w} →
    (Function[{x, u}, #1]&) /@ ({t[x, u],
       w[x, u]} /. {solct[[1]], solcw[[1]]})]
```

$$\left\{t \to \text{Function}\left[\{x, u\}, \text{ArcTan}\left[\frac{x}{u}\right]\right],\right.$$
$$\left. w \to \text{Function}\left[\{x, u\}, -\frac{u^2}{2} - \frac{x^2}{2}\right]\right\}$$

The canonical variables of the rotation group are given by the *ArcTan* of x divided by u and the representation of a circle. □

Example 3

Another kind of symmetry frequently encountered in problems is given by a projective group

```
veccan = {ξ → Function[{x, u}, x u],
  ϕ → Function[{x, u}, u]}
```

```
{ξ → Function[{x, u}, x u], ϕ → Function[{x, u}, u]}
```

The corresponding determining equations are

```
caneq = canonicalEquations /. veccan;
caneq // LieTraditionalForm // TableForm
```

$$u\, t_u + u\, x\, t_x == 1$$
$$u\, w_u + u\, x\, w_x == 0$$

We solve these equations by using the function DSolve[] for the first and second equations:

```
solct = DSolve[caneq[[1]], t[x, u], {x, u}]
```

```
{{t[x, u] → Log[u] + C[1][u - Log[x]]}}
```

```
solcw = DSolve[caneq[[2]], w[x, u], {x, u}]
```

```
{{w[x, u] → C[1][u - Log[x]]}}
```

We extract a special solution from these expressions by

```
csol = Flatten[{solct /. C[1][_] → 0,
    solcw /. C[1][x_] → x}]
```

```
{t[x, u] → Log[u], w[x, u] → u - Log[x]}
```

and transform the relations to

```
r4 = Thread[{t, w} →
    (Function[{x, u}, #1]&) /@ ({t[x, u],
       w[x, u]} /. csol)]
```

```
{t → Function[{x, u}, Log[u]], w → Function[{x, u}, u - Log[x]]}
```

The result is a representation of canonical variables for the projective group. □

The interactive steps of the calculation can be collected in a *Mathematica* function CanonicalVariables[]. The function needs the dependent and independent variables of the old coordinates. We also supply the infinitesimals for these coordinates as input information. The last two arguments contain lists of the new variable names for the dependent and independent variables, respectively.

The function CanonicalVariables[] calculates the general transformation using the first-order partial differential equations defining the determining equations. The function is designed to generalize the theory in such a way that an arbitrary number of independent and dependent variables can be transformed. The returned result of the function CanonicalVariables[] is a list of solutions represented in a pure function form. The following code serves as the definition of this function.

```
Clear[CanonicalVariables]
CanonicalVariables[depend_List, indep_List,
 xi_List, phi_List,
 ndepend_List, nindep_List] :=
 Block[{equations = {}, solutions = {},
   ssol, rule, csol},
  vars = Join[indep, depend];
  infini = Join[xi, phi];
  nvars = Join[ndepend, nindep];
  newvars = Table[Unique["w$"],
    {i, 1, Length[vars]}];
  dnewv = Table[newvars[[i]] @@vars,
    {i, 1, Length[newvars]}];
  rule = Thread[newvars → nvars];
  Do[
   AppendTo[equations,
    Sum[infini[[i]] ∂vars[[i]] dnewv[[j]] == 0, {i, 1, Length[vars]}]],
   {j, 1, Length[dnewv] - 1}];
  AppendTo[equations,
   Sum[infini[[i]] ∂vars[[i]] Last[dnewv] == 1, {i, 1, Length[vars]}]];
```

```
Do[
 AppendTo[solutions,
  DSolve[equations[[i]], dnewv[[i]], vars]],
 {i, 1, Length[equations]}];
ssol = PowerExpand[Table[solutions[[i]] /.
    C[_][x___] :> {x}[[i]],
   {i, 1, Length[solutions] - 1}]];
AppendTo[ssol, PowerExpand[Last[solutions] /.
   C[_][___] → 0]];
csol = Flatten[ssol] /. rule;
Thread[nvars →
  (Function[vars$, #1]&) /@ (dnewv /.
     rule /. csol)] /. vars$ → vars]
```

The application of the function CanonicalVariables[] to different symmetry groups is demonstrated in the examples below. Let us first examine an inhomogeneous scaling group for two variables u and x. The names of the new variables are w and t:

```
CanonicalVariables[{u}, {x}, {a x},
 {b u}, {w}, {t}]
```

$$\{w \to \text{Function}[\{x, u\}, u\, x^{-\frac{b}{a}}],\ t \to \text{Function}[\{x, u\}, \frac{\text{Log}[x]}{a}]\}$$

The result is a logarithmic dependence in the new independent variable t and a quotient of u and $x^{b/a}$. Another example is related to the projective group. The result becomes more readable if we apply the function LTF[]:

```
CanonicalVariables[{u}, {x}, {x^2 u}, {1},
 {w}, {t}] // LTF
```

$$t == -u$$

$$w == \frac{u^2}{2} + \frac{1}{x}$$

A further example is the inhomogeneous translation group

```
CanonicalVariables[{u}, {x}, {1}, {k},
 {w}, {t}] // LTF
```

$$t == \frac{x}{1}$$

$$w == -\frac{-1\,u + k\,x}{1}$$

The symmetry of rotation is connected with the canonical variables

```
CanonicalVariables[{u}, {x}, {u}, {-x},
 {w}, {t}] // LTF
```

```
t == -ArcTan[x/u]
w == -u^2/2 - x^2/2
```

The scaling of one of the coordinates allows the canonical variables

```
CanonicalVariables[{u}, {x}, {x}, {1},
 {w}, {t}] // LTF
```

```
t == Log[x]
w == u - Log[x]
```

```
CanonicalVariables[{u}, {x}, {1}, {u},
 {w}, {t}] // LTF
```

```
t == x
w == E^-x u
```

Several other examples demonstrate the capabilities of this function. The occurrence of the Log[] in one of the infinitesimals demonstrates the flexibility of the function

```
CanonicalVariables[{u}, {x}, {Log[x] u}, {1},
 {w}, {t}] // LTF
```

```
t == -u
w == u^2/2 - LogIntegral[x]
```

An example with parameters in the infinitesimals shows that CanonicalVariables[] can handle rational expressions:

```
CanonicalVariables[{u}, {x}, {k1 + k2 x},
 {k3 + 2 k2 u}, {w}, {t}] // LTF
```

$$t == \frac{\text{Log}[k1 + k2\,x]}{k2}$$

$$w == -\frac{-k3 - 2\,k2\,u}{2\,k2\,(k1 + k2\,x)^2}$$

An example for a higher-dimensional manifold $\mathfrak{M}$ is examined next:

```
CanonicalVariables[{u, v}, {x, t}, {x, t},
 {0, 0}, {un, vn}, {xn, tn}] // LTF
```

```
tn == Log[x]
un == u
vn == v
xn == t/x
```

This list of examples can be extended by the reader's experiments. We demonstrated that the knowledge of the infinitesimals allows the introduction of new variables. Lie pointed out that these canonical variables simplify the solution of the original equation. The advantage of canonical coordinates is that we may simplify the solution algorithm. This algorithm consists of first finding the infinitesimals, second, calculating the canonical variables, and third, transforming the original equation to a simpler form. These three steps will be the subject of the next sections.

4.4. Analysis of Ordinary Differential Equations

The Lie point symmetries of an ordinary differential equation $\Delta = 0$ are determined by calculating the general solution for the infinitesimals ξ and ϕ. The procedure follows the steps discussed in Section 4.3. The first step is to write down the invariance condition for the equation and then solve the linear determining equations. The steps of deriving and solving the determining equations are accessible within *MathLie* and can be carried out automatically. The calculations done by hand are very cumbersome, but using computer algebra, the work is easy to accomplish. However, the view taken here is somewhat optimistic and cases exist which sometimes involve peculiar results.

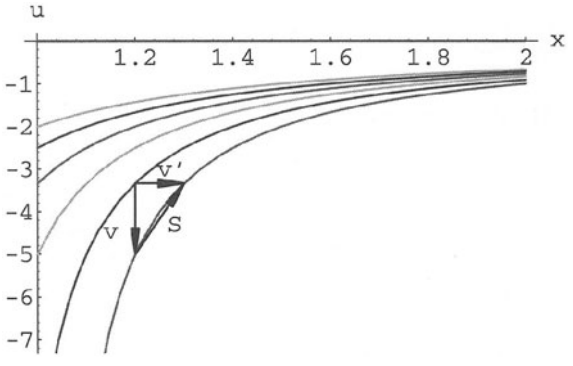
One of these peculiarities is the case of first-order differential equations. As we will soon show, first-order differential equations always have an infinite number of symmetries and are thus not very appropriate for Lie's method. However, we will discuss a procedure which allows us to find a restricted class of point symmetries, the so-called conformal symmetries.

4.4.1 First-Order Equations

The general representation of a first-order ordinary differential equation is given by

$$F\left(x, u(x), \frac{du}{dx}\right) = 0, \tag{4.32}$$

where F is an arbitrary function combining the independent and dependent variables x and u, respectively, and the derivatives in a general way. Our intention here is to use point symmetries to solve this type of equation. We state the main result at the beginning of this section. A first-order differential equation always has an infinite number of symmetries. This is immediately obvious if we consider the geometrical interpretation of the equation. As we know, the set of solutions of a first-order differential equation is a one-parameter family of curves which look like the ensemble in the following figure:



A symmetry transformation is by definition a transformation mapping solutions into solutions. As the example in the above figure shows, new solutions follow from old solutions by successively changing the involved integration constant. Every symmetry transformation is represented by a generator $\vec{v}$ corresponding to a vector field leading from curves to their neighboring curves. As we know, the vector field always exists and can be represented by the infinitesimals ξ and ϕ. A different choice $\vec{v}'$ of the vector field will result in a different location on the target curve; however, both points on the target curve are connected by a third vector field $\vec{S}$ (see the figure). This is one of several ways to interpret a differential equation. Another view is the following:

From a conceptual point of view, a differential equation contains two components:

(i) the skeleton of the differential equation

(ii) the solution manifold.

These two parts of a differential equation will be the subject of the following. The term skeleton was introduced by Lie [1899] to denote the extended manifold on which the differential equation exists. We will show you how the symmetries connect these two parts. We start with the skeleton of a first-order ordinary differential equation.

4.4.1.1 The Skeleton of an Ordinary Differential Equation

As mentioned, the concept of an ordinary differential equation has two components. Let us first consider the first-order differential equation in the form

$$F(x, u(x), u') = 0, \tag{4.33}$$

where $u' = \frac{du}{dx}$ is the first-order derivative. The two components of an ordinary differential equation are as follows:

Definition: The Skeleton

The skeleton of a differential equation is defined as the surface

$$F(x, u, p) = 0 \tag{4.34}$$

in the space of the independent variables x, u, and p. u and p denote the sets of dependent variables and derivatives, respectively. The corresponding differential equation follows from the skeleton with the replacement of p by the derivatives $u_{(1)}$. ○

This general definition reduced to first-order equations introduces nothing more than an extension of the manifold $\mathfrak{M}$. This extended manifold consists of the independent and dependent variables with the third direction denoting the first derivative. The once extended manifold is a very useful term in the discussion of first-order ordinary differential equations. Another term we need is the class of solutions.

The class of solutions is defined in consensus with certain mathematical or physical assumptions.

Definition: A class of solutions

A solution is a continuously differentiable function $h(x)$ such that the curve $u = h(x)$, $u' = \frac{dh(x)}{dx}$ belongs to the skeleton, that is, $F(x, h(x), \frac{dh(x)}{dx}) = 0$ identically in x for some interval. ○

The combination of both terms allows us to solve the first-order equation. The crucial step in integrating differential equations is a simplification of the skeleton. This simplification can be gained by a suitable change of variables x and u. To this end, we use symmetry groups of differential equations, leaving the skeleton invariant. Provided a symmetry group is known, a simplification of the skeleton can be carried out by introducing canonical variables, for example. This kind of simplification is demonstrated by the following examples.

Example 1

Let us consider the Riccati equation as a first example:

```
riccati = ∂x u[x] + u[x]^2 - 2/x^2 == 0; riccati // LTF
```

$$u^2 - \frac{2}{x^2} + u_x == 0$$

The skeleton of this Riccati equation is defined by the algebraic equation

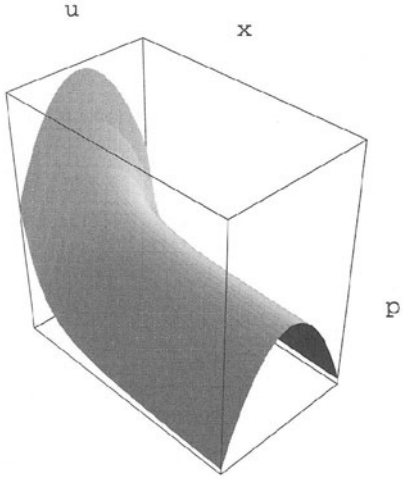
$$p + u^2 - \frac{2}{x^2} == 0$$

and its surface, a so-called hyperbolic paraboloid, can be displayed by defining the skeleton as a *Mathematica* function:

$$f[x\_, u\_] := -u^2 + \frac{2}{x^2}$$

The corresponding surface in three dimensions is given by

```
pl1 = Plot3D[f[x, u], {x, 0.25, 2}, {u, -5, 5},
  Boxed → True, Axes → True, Mesh → False,
  Ticks → None, PlotPoints → 30,
  BoxRatios → {1, 3/5, 1},
  ViewPoint → {1.975, -1.884, 2.000},
  AxesLabel → {"x", "u", "p"}]
```



This figure shows that the skeleton in the coordinates x, u, and p has a singularity in the limit $x \to 0$. We also observe the parabolic shape of the surface for large x-values. Thus, the surface is twisted in two directions, which obviously baffles the discovery of the solution. Our goal is to find a transformation which reduces the twisted shape to a simpler representation. For the Riccati equation, a one-parameter symmetry group is provided by the following scaling transformations. Ibragimov [1994] calls this transformation a non-homogeneous dilation.

```
transformation = {x → r Exp[-a],
  u → Function[x, w[x Exp[a]] Exp[a]]}
```

```
{x → E^-a r, u → Function[x, w[x Exp[a]] Exp[a]]}
```

We can check the invariance of the Riccati equation by taking into account that the derivatives also need to be transformed by the rule

```
dtrafo = v_^(n_) [a_. x_] :> a^-n v^(n) [a x]
```

```
v_^(n_) [a_. x_] :> a^-n v^(n) [a x]
```

The rule *dtrafo* represents the fact that the nth derivative of a function under a scaling is replaced by the nth derivative divided by the scaling factor a^n. The application of *transformation* and *dtrafo* to the Riccati equation gives us

```
triccati = riccati /. transformation /. dtrafo; triccati // LTF
```

$$-\frac{2\,E^{2\,a}}{r^2} + E^{2\,a}\,w^2 + E^{2\,a}\,w_r == 0$$

We note that the original equation *riccati* is reproduced up to a common factor E^{2a}. In the definition of the transformation, it is essential that we use the new variables in the representation for the original variables. x is simply replaced by the new independent variable r multiplied by the factor E^{-a}. The dependent variable u is replaced by wE^a. Since w depends on the new variable r, we have to use the representation of r in the form $x\,E^a$. We also have to take into account that the derivatives need a special treatment which is contained in the rule *dtrafo*. Combining all these rules in the transformation of the original equation, we end up with the equation given in *triccati*. The different replacements of variables are actually the steps necessary to carry out the transformation by hand. The application of the transformation on the first derivative shows us that its behavior is

```
∂x u[x] /. transformation /. dtrafo // LTF
```

$$E^{2\,a}\,w_r == 0$$

which in conventional notation reads $w' \to u'\,e^{-2a}$. Thus, we observe that the equation's skeleton is invariant under the inhomogeneous stretching $r \to x\,e^a$, $w \to ue^{-a}$, $w' \to u'\,e^{-2a}$ which is obtained by extending the transformations of the group to the first derivative u'. We can also check the invariance of the skeleton equation by applying the extended vector field to the skeleton equation. We define the vector field by

```
Vect[f_] := x ∂x f - u ∂u f - 2 p ∂p f
```

The skeleton of the Riccati equation is

```
skeleton = riccati /. {∂x u[x] → p, u[x] → u}
```

$$p + u^2 - \frac{2}{x^2} == 0$$

The application of the vector field gives us

```
Vect /@ skeleton
```

$$-2\,p - 2\,u^2 + \frac{4}{x^2} == 0$$

If we compare the two expressions, the original and the transformed, we observe that the original equation is reproduced up to a factor –2. Thus, if the skeleton vanishes, the application of the vector field to the skeleton also vanishes. This result shows us that under a scaling transformation, the skeleton of the Riccati equation is invariant. □

Example 2

Another example for a first-order ordinary differential equation is

```
example2 = -∂x u[x] / x^2 + (u[x]^2 / x^2 + u[x]^3 / x) == 0; example2 // LTF
```

$$\frac{u^2}{x^2} + \frac{u^3}{x} - \frac{u_x}{x^2} == 0$$

This example is also invariant with respect to an inhomogeneous scaling transformation. We define this sort of transformation by a transformation rule like

```
scalingtrafo = {x → Exp[-a] r,
  u → Function[x, w[x Exp[a]] Exp[a]]};
dtrafo = v_^(n_) [a_. x_] :> a^-n v^(n) [a x];
scaling[x_] := x /. scalingtrafo /. dtrafo
```

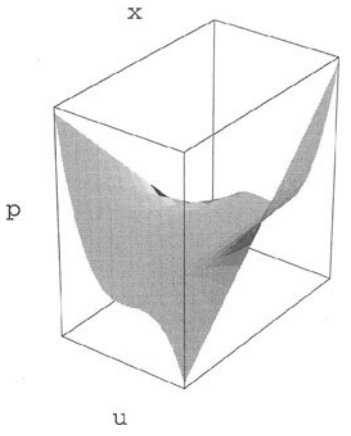
The application of the function scaling[] to the second example shows the invariance of this equation

```
scaling[example2] // LTF
```

$$\frac{E^{4\,a}\,w^2}{r^2} + \frac{E^{4\,a}\,w^3}{r} - \frac{E^{4\,a}\,w_r}{r^2} == 0$$

The graphical representation of the skeleton for this equation is created by

```
Plot3D[u^2 - x u^3, {x, -1, 1}, {u, -2, 2},
 PlotRange → All, AxesLabel → {"x", "u", "p"},
 ViewPoint → {5.287, 3.905, 3.613},
 Mesh → False, Ticks → None, BoxRatios → {1, 3/5, 1}]
```



This three-dimensional representation of the skeleton looks like a stingray. The structure of this entangled surface is simplified if we apply the following canonical transformation:

```
canonical = {x → Exp[t], u → Function[x, w[Log[x]]/x]};
canonicaltransform[x_] := Simplify[PowerExpand[
   x /. canonical]]
```

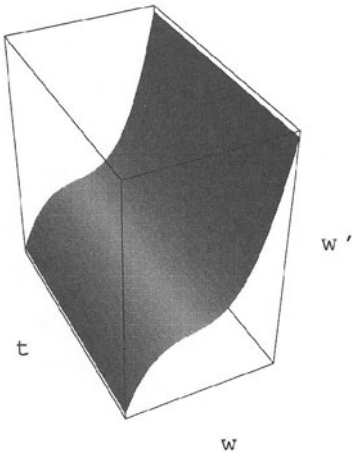
The transformation is carried out by

```
canonicaltransform[Thread[example2 Exp[4 t], Equal]] // LTF
```

$$w + w^2 + w^3 - w_t == 0$$

The related skeleton simplifies the surface:

```
Plot3D[w + w^2 + w^3, {t, -1, 1}, {w, -2, 2},
 PlotRange → All, ViewPoint → {2.460, -1.182, 2.000},
 AxesLabel → {"t", "w", "w'"},
 ViewPoint → {5.287, 3.905, 3.613},
 PlotPoints → 30, Mesh → False, Ticks → None,
 BoxRatios → {1, 3/5, 1}]
```



which is a third-order parabola translated along the t-axis. This example shows that the introduction of canonical coordinates radically simplifies the shape of the skeleton and thus allows access to the solution. □

Example 3

Another example demonstrating the concept of the skeleton is given by the first-order equation

```
example3 = x ∂x u[x] - u[x] + Sqrt[u[x]/x] == 0; example3 // LTF
```

$$-u + \sqrt{\frac{u}{x}} + x\,u_x == 0$$

This type of equation is invariant with respect to the global transformation $\Xi = \frac{x}{1-\epsilon x}$ and $\Phi = \frac{y}{1-\epsilon x}$; the related canonical variables are $w = \frac{u}{x}$ and $t = \frac{-1}{x}$. The equation in canonical variables reads

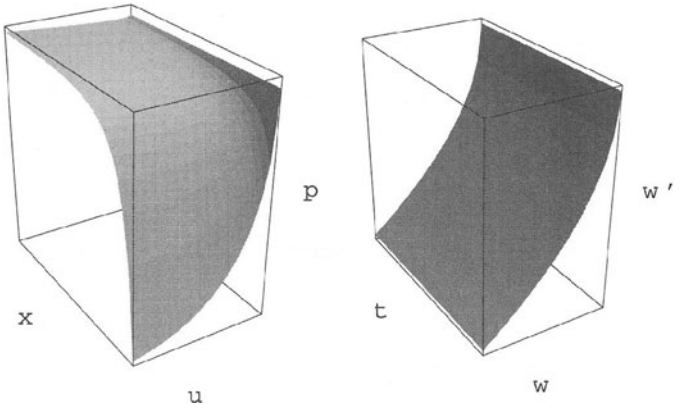
```
cexample3 = ∂t w[t] + Sqrt[w[t]] == 0; cexample3 // LTF
```

$$\sqrt{w} + w_t == 0$$

Each skeleton of these two equations is represented in the following figure:

```
Show[GraphicsArray[
  {Plot3D[u/x - Sqrt[u/x]/x, {x, .1, 1}, {u, 0, 2},
    PlotRange → All, AxesLabel → {"x", "u", "p"},
```

```
    ViewPoint → {-2.566, 1.572, 1.548}, Mesh → False,
    Ticks → None, PlotPoints → 30,
    DisplayFunction → Identity, BoxRatios → {1, 3/5, 1}],
  Plot3D[-√w, {t, .1, 1}, {w, 0, 2},
    PlotRange → All, AxesLabel → {"t", "w", "w'"},
    ViewPoint → {-2.566, 1.572, 1.548}, Mesh → False,
    Ticks → None, PlotPoints → 30,
    DisplayFunction → Identity,
    BoxRatios → {1, 3/5, 1}]}],
DisplayFunction → $DisplayFunction]
```



It is obvious from this figure that the skeleton of the original equation is reduced to a much simpler representation. The convex shape of the original skeleton reduces to a concave surface which is invariant with respect to a translation along the t-axis. □

Upon observing that a canonical transformation simplifies the skeleton, we approach the second component of a differential equation. The second component was concerned with the solution of the equation. The question now is: How can we use the information of the symmetries to find solutions of any first-order equation possessing some symmetries? One idea to solve this problem is to use canonical variables which allow a transformation to a simpler representation of the equation.

Returning to our first example of the Riccati equation, we know that the canonical variables for a non-homogeneous dilation is given by

```
trafo1 = {x → Exp[t], u → Function[x, w[Log[x]]/x]}
```

$$\{x \to E^t,\ u \to \text{Function}\left[x, \frac{w[\text{Log}[x]]}{x}\right]\}$$

In Section 4.3.4, we derived this type of transformations by solving the defining equations for t and w. The solution used here is a special type of the general solution derived in Section 4.3.4. Application of this transformation to the Riccati equation gives us the following result:

```
riccati1 = Simplify[Thread[
    PowerExpand[riccati /. trafo1] Exp[2 t], Equal]];
riccati1 // LTF
```

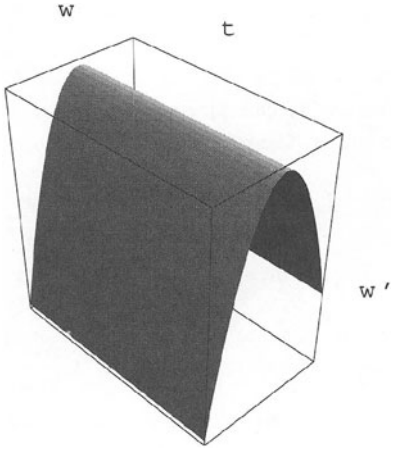
$$-2 - w + w^2 + w_t == 0$$

The transformation straightens out the skeleton of the original Riccati equation, taking it to a parabolic cylinder

```
g[t_, w_] := 2 + w - w^2
```

The right-hand side of g is independent of t; thus, the skeleton reduces to a cylindrical surface centered along the t-axis.

```
pl2 = Plot3D[g[t, w], {t, 0.25, 2}, {w, -5, 5},
  Boxed → True, Axes → True, Mesh → False, Ticks → None,
  PlotPoints → 30, BoxRatios → {1, 3/5, 1},
  ViewPoint → {1.975, -1.884, 2.000},
  AxesLabel → {"t", "w", "w'"}]
```



The simplification of the skeleton is the reason why the Riccati equation takes the integrable form when written in canonical variables (cf. the skeleton of the original Riccati equation).

The stretching of the original Riccati equation is replaced by a group of translations $\bar{t} = t + \epsilon$, $w = u$, and $w' = u'$. The calculations so far executed by hand can be collected in a *Mathematica* function. The information we need for the calculation are the original equation and the names of the dependent and independent variables. We also need the canonical transformations which are derived by the function CanonicalVariables[]. Finally, the function has to know the names of the new coordinates. The representation of the original equation is carried out by the function CanonicalRepresentation[]:

```
Clear[CanonicalRepresentation]
CanonicalRepresentation[equation_, depend_,
 independ_, canonicaltrafo_List, newdepend_,
 newindepend_] :=
 Block[{pattern1, trafo1, eqin, ctrafo,
   canonicalvariables, sol, equat},
  pattern1 = a_. u_'[x_] + p_. == 0 :> a Dt[u]/Dt[x] + p;
  trafo1 = depend@@{independ} → depend;
  eqin = equation /. pattern1 /. trafo1;
  ctrafo = canonicaltrafo /. Rule → Equal;
  canonicalvariables = Flatten[canonicaltrafo /.
      (a_ → b_) → {a}];
  sol = Solve[ctrafo, {depend, independ}];
  equat = Expand[eqin /. sol];
  equat = equat /. newdepend → newdepend@@{newindepend};
  equat[[1]] == 0]
```

We demonstrate the use of this function by applying it to the Riccati equation. We know that the Riccati equation is invariant with respect to an inhomogeneous scaling transformation. The related canonical variables are given by

```
ccoord = CanonicalVariables[{u}, {x}, {x}, {-u},
  {w}, {t}]
```

```
{w → Function[{x, u}, u x], t → Function[{x, u}, Log[x]]}
```

The result is identical with the variables stated above. Using this transformation, we can reduce the Riccati equation to

```
cricc = CanonicalRepresentation[riccati, u, x,
  {w → x u, t → Log[x]}, w, t]; cricc // LTF
```

```
-2 E^-2 t - E^-2 t w + E^-2 t w^2 + E^-2 t w_t == 0
```

The resulting equation has the same structure as the manually derived one. The common factor e^{-2t} is a non-vanishing term which can be eliminated by multiplying with the inverse:

```
cricc = Simplify[Thread[cricc Exp[2 t], Equal]]; cricc // LTF
```

```
-2 - w + w^2 + w_t == 0
```

This equation can be solved by quadrature or a separation of variables. We use here the *Mathematica* function DSolve[] to integrate the equation.

```
sricc = DSolve[cricc, w, t]
```

$$\left\{\left\{w \to \left(\frac{2\,E^{3\,\#1} + E^{3\,C[1]}}{E^{3\,\#1} - E^{3\,C[1]}}\&\right)\right\}\right\}$$

The solution in the original variables follows by applying the canonical transformation again,

```
solricc = Simplify[w[x, u] == (w[t] /. sricc)[[1]] /.
    t -> t[x, u] /. ccoord]
```

$$u\,x == \frac{E^{3\,C[1]} + 2\,x^3}{-E^{3\,C[1]} + x^3}$$

and solving the implicit solution with respect to u,

```
Solve[solricc, u]
```

$$\left\{\left\{u \to \frac{E^{3\,C[1]} + 2\,x^3}{x\,(-E^{3\,C[1]} + x^3)}\right\}\right\}$$

The second of our examples discussed above allows the same scaling symmetries. The reduced equation follows from

```
cex2 = CanonicalRepresentation[example2, u, x,
  {w -> x u, t -> Log[x]}, w, t]; cex2 // LTF
```

```
E^-4 t w + E^-4 t w^2 + E^-4 t w^3 - E^-4 t w_t == 0
```

The common factor is eliminated by

```
cex2 = Simplify[Thread[cex2 Exp[4 t], Equal]]; cex2 // LTF
```

```
w + w^2 + w^3 - w_t == 0
```

The solution in the canonical variable w follows by separation of variables and integrating the left-hand side and the right-hand side:

```
sex2 = Simplify[∫ 1/(w + w^2 + w^3) dw == ∫ 1 dt + c]
```

$$-\frac{\text{ArcTan}\left[\frac{1+2w}{\sqrt{3}}\right]}{\sqrt{3}} + \text{Log}[w] - \frac{1}{2}\text{Log}[1 + w + w^2] == c + t$$

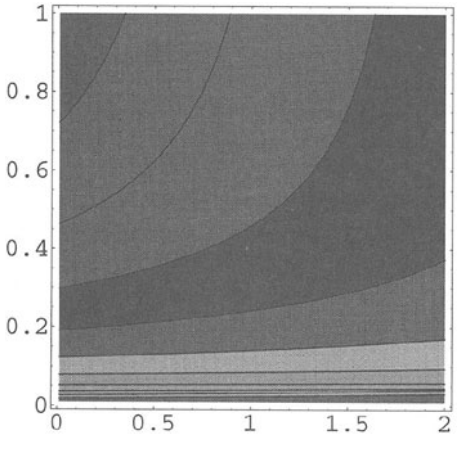
Since the solution contains transcendent functions, it is hard to get an explicit solution at this stage. The inversion of the canonical transformation does not resolve this problem:

```
iex2 = sex2 /. {w → w[x, u], t → t[x, u]} /. ccoord
```

$$-\frac{\text{ArcTan}\left[\frac{1+2ux}{\sqrt{3}}\right]}{\sqrt{3}} + \text{Log}[ux] - \frac{1}{2}\text{Log}[1 + ux + u^2x^2] == c + \text{Log}[x]$$

However, we can resolve this problem by a graphical representation of the solution. We create a contour plot by displaying the implicit function for a fixed parameter c:

```
iex2h = iex2 /. a_ == b_ → a - b /. c → 1;

ContourPlot[iex2h, {x, .01, 2}, {u, .01, 1},
 ColorFunction → Hue, PlotPoints → 20]
```



Generally, we observe that u increases if x increases. For small values of u, there exists a nearly linear relation between u and x.

The third example discussed above is represented in canonical variables by

```
cex3 = CanonicalRepresentation[example3, u, x,
  {w → u/x, t → -1/x}, w, t]; cex3 // LTF
```

$$\sqrt{w} + w_t == 0$$

The solution of this equation for w follows from

```
sex3 = DSolve[cex3, w, t]
```

$$\left\{\left\{w \to \left(\frac{1}{4}\left(\#1^2 - 2\,\#1\,C[1] + C[1]^2\right)\&\right)\right\}\right\}$$

Inverting the transformation and solving for u gives us the explicit solution

```
solex3 = u/x == (w[t] /. sex3)[[1]] /. t → -1/x
```

$$\frac{u}{x} == \frac{1}{4}\left(\frac{1}{x^2} + \frac{2\,C[1]}{x} + C[1]^2\right)$$

```
sol3 = Solve[solex3, u]
```

$$\left\{\left\{u \to -\frac{-1 - 2\,x\,C[1] - x^2\,C[1]^2}{4\,x}\right\}\right\}$$

So far, we have discussed some examples to show that canonical variables can be used to simplify the integration process of first-order equations. The method of canonical variables is useful not only in the integration process of first-order ordinary differential equations but also in the integration of higher-order ordinary differential equations. We will come back to this general procedure in Section 4.4.2.2.

Before proceeding with second-order equations, we first discuss another approach to integrate first-order ordinary differential equations. This method uses the fact that a first-order equation can be integrated if an integrating factor is known. The procedure introduced by Lie in 1891 is very useful if one knows the point symmetries of the equation in an infinitesimal representation.

4.4.1.2 Integrating Factor

The common belief in literature is that the method of an integrating factor is only useful in connection with a first-order ordinary differential equation. For the moment, we will take this point of view. However, in Section 4.4.2.2, we will generalize the method of an integrating factor to higher-order equations. In Section 4.2.3, we demonstrated by several examples that a curve is invariant under a symmetry transformation if the tangent vector applied to the curve vanishes. Let us assume that the curves discussed are integral curves or solutions of a first-order differential equation. For example, the ODE is

```
ode1 = ∂x u[x] == f[x, u[x]]; ode1 // LTF
```

$$-f + u_x == 0$$

where f is given by the ratio of two functions U and X

```
f[x, u[x]] = U[x, u[x]] / X[x, u[x]];
```

If $F[x,u]$ = const. are integral curves, we get

```
invar1 = Dt[F[x, u]] == 0; invar1 // LTF
```

$$\mathrm{Dt}[u]\ F_u + \mathrm{Dt}[x]\ F_x == 0$$

Let us further assume that the differential equation is invariant under an infinitesimal transformation represented by the vector field $\vec{v}$. Then, we find

```
invar2 = TangentVector[F[x, u], {u}, {x}] == H; invar2 // LTF
```

$$-H + F_x\ \xi_1[x, u] + F_u\ \phi_1[x, u] == 0$$

We always can rescale *invar2* in such a way that the right-hand side equals 1.

```
invar2 = invar2 /. H → 1; invar2 // LTF
```

$$-1 + F_x\ \xi_1[x, u] + F_u\ \phi_1[x, u] == 0$$

Thus, we derived two equations for the derivatives of the invariant curve which have to be satisfied under the given infinitesimal transformation. We can solve these two equations for the derivatives of F by

```
sol = Solve[{invar1, invar2},
  {∂x F[x, u], ∂u F[x, u]}] /. {Dt[u] → U[x, u],
   Dt[x] → X[x, u]}
```

$$\left\{\left\{F^{(1,0)}[x, u] \to \frac{U[x, u]}{U[x, u]\ \xi_1[x, u] - X[x, u]\ \phi_1[x, u]},\right.\right.$$
$$\left.\left.F^{(0,1)}[x, u] \to -\frac{X[x, u]}{U[x, u]\ \xi_1[x, u] - X[x, u]\ \phi_1[x, u]}\right\}\right\}$$

We understand from this result that the partial derivatives of the integral curve are known as functions of x and u. On the other hand, the result shows that the total differential of the integral curve F is known. The following expression represents the total differential Dt[F]:

```
invar3 = Simplify[invar1 /. sol /. 0 → Dt[F]]
```

$$\left\{\frac{\mathrm{Dt}[x]\ U[x, u] - \mathrm{Dt}[u]\ X[x, u]}{U[x, u]\ \xi_1[x, u] - X[x, u]\ \phi_1[x, u]} == \mathrm{Dt}[F]\right\}$$

Recall the definition of an integrating factor $\mu(x, u)$ which is, by definition, a function of x and u that makes Dt[x] Y − Dt[u] X the differential of the integral curve F. This information allows us to extract from our result *invar3* the integrating factor

$$\mu = \frac{1}{\text{Denominator[invar3[[1, 1]]]}}$$

$$\frac{1}{\text{U[x, u]}\ \xi_1\text{[x, u]} - \text{X[x, u]}\ \phi_1\text{[x, u]}}$$

This result was obtained by Lie in 1874. For the derivation of integrating factors, it is important to know the structure of the equation and the symmetries.

In 1874, Lie proved that a first-order ordinary differential equation can be solved by a quadrature if the symmetries of the equation are known. He collected his observations in the following theorem.

Theorem: Integrating factor

The first-order ordinary differential equation

$$U(x, u)\,dx - X(x, u)\,du = 0 \tag{4.35}$$

possesses a one-parameter group allowing the vector field $\vec{v} = \xi\,\partial_x + \phi\,\partial_u$ if and only if the function

$$\mu = \frac{1}{\xi U - \phi X} \tag{4.36}$$

is an integrating factor with $\xi U - \phi X \neq 0$. If this is the case, the original equation is solved by a quadrature:

$$\int \frac{U\,dx - X\,du}{\xi U - \phi X} = \text{const.}\ \circ \tag{4.37}$$

Relation (4.37) can be simplified if we introduce the following determinants:

$$d\Delta = \det\begin{pmatrix} dx & du \\ X & U \end{pmatrix} \tag{4.38}$$

and

$$\Delta = \det\begin{pmatrix} \xi & \phi \\ X & U \end{pmatrix} = \frac{1}{\mu}. \tag{4.39}$$

Then, equation (4.37) reduces to the simple relation

$$\int \left(\frac{d\Delta}{\Delta}\right) = \text{const}. \tag{4.40}$$

Equation (4.40) combines all information necessary for a solution in a nutshell. All we need to know are the infinitesimals and the left-hand side of equation (4.35). Integration over the manifold provides us with the solution. A few examples will demonstrate the application of Lie's theorem.

Example 1

As an example of these considerations, let us examine the first-order ODE

```
ode2 = ∂x u[x] == 1 + Tan[x - u]/x; ode2 // LieTraditionalForm
```

$$u_x == 1 - \frac{\text{Tan}[u - x]}{x}$$

The functions $X[x, u]$ and $U[x, u]$ are found by extracting the coefficients of the differentials. First, we extract this part from the ODE free of any differential:

```
ode2h = ode2[[2]]
```

$$1 - \frac{\text{Tan}[u - x]}{x}$$

Then, we generate a common denominator,

```
ode2h = Together[ode2h]
```

$$\frac{x - \text{Tan}[u - x]}{x}$$

and determine $X[x, u]$ by

```
X[x, u] = Denominator[ode2h]
```

```
x
```

$U[x, u]$ is then given by

```
U[x, u] = Numerator[ode2h]
```

```
x - Tan[u - x]
```

The infinitesimals of the first-order equation *ode2* are

```
ξ1[x, u] = 0;
```

and

$$\phi_1[x, u] = \frac{1}{x\,\mathrm{Cos}[x-u]};$$

Knowing these relations, we use the definition of μ to get the explicit representation of the integrating factor

```
μ
```

```
-Cos[u - x]
```

The related total differential of the integral curve reads

```
Expand[invar3]
```

```
{-Cos[u - x] (-x Dt[u] + Dt[x] (x - Tan[u - x])) == Dt[F]}
```

This equation has to be integrated on the right-hand side with respect to F, and on the left-hand side with respect to x and u. The result is

```
ip = F == ∫Coefficient[
     Expand[invar3[[1, 1]]], Dt[u]] du +
    ∫Coefficient[Expand[invar3[[1, 1]]], Dt[x]] dx
```

```
F == 2 x Sin[u - x]
```

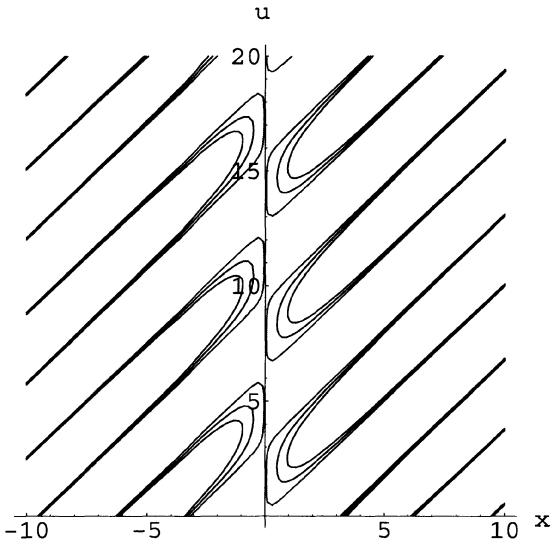
defining the solution of *ode2* in an implicit form. To see how this solution behaves in the variables x and u, we graphically represent this solution for three values of the constant F. The use of the function ImplicitPlot[] creates a contour plot of the implicit function in the (x, u)-plane:

```
<< "Graphics`ImplicitPlot`"
```

```
iph = Table[ip /. F → i, {i, .1, 2, .9}]
```

```
{0.1 == 2 x Sin[u - x], 1. == 2 x Sin[u - x], 1.9 == 2 x Sin[u - x]}
```

```
ImplicitPlot[iph, {x, -10, 10}, {u, 0, 20},
 AxesLabel → {"x", "u"}, PlotPoints → 100,
 PlotStyle → {RGBColor[1.000, 0.000, 0.000],
   RGBColor[0.000, 0.000, 1.000],
   RGBColor[0.000, 0.251, 0.000]}]
```



We clearly observe that small values for F will force the function to be located at the vertical axis at $x = 0$. Larger values of F push the function away from this vertical axis. Examination of this example demonstrates that the knowledge of the infinitesimals is very useful in constructing solutions of first-order differential equations. For the present equation, we only stated the existence of the infinitesimal transformations. In Section 4.4.1.3, we will show how such infinitesimals can be derived from the invariance condition of the differential equation. □

Let us demonstrate the application of Lie's integrating factor theorem by another example. We are especially interested in collecting the steps of calculation in a *Mathematica* function. The basis of this function is the above theorem.

Example 2

Let us again examine the Riccati equation discussed in connection with the skeleton of a first-order ordinary differential equation:

```
riccati // LTF
```

$$u^2 - \frac{2}{x^2} + u_x == 0$$

Considering this equation, it is useful to determine the symmetry. Since this equation is of a polynomial type, it is natural to assume that the symmetries of this equation are of a scaling type. We already know that the infinitesimals are given by $\xi = x$ and $\phi = -u$. If we represent the Riccati equation in the form

```
ricc = Dt[u] + (u^2 - 2/x^2) Dt[x] == 0
```

$$\text{Dt}[u] + \left(u^2 - \frac{2}{x^2}\right) \text{Dt}[x] == 0$$

we are able to use Lie's theorem to determine the integrating factor in a straightforward way:

```
μ = Simplify[Together[1/(x (u^2 - 2/x^2) - u)]]
```

$$\frac{x}{-2 - u\,x + u^2\,x^2}$$

Multiplying the equation *ricc* by the integrating factor μ, we get

```
ricc1 = Simplify[Thread[ricc mu, Equal]]
```

$$\frac{x\,(\text{Dt}[u] + (u^2 - \frac{2}{x^2})\,\text{Dt}[x])}{-2 - u\,x + u^2\,x^2} == 0$$

Integrating the coefficients of the total differentials with respect to the differential, we get for *Dt*[*u*] and *Dt*[*x*] respectively

```
solu = ∫Coefficient[Expand[ricc1[[1]]], Dt[u]] du
```

$$\frac{1}{3}\text{Log}[-2 + u\,x] - \frac{1}{3}\text{Log}[1 + u\,x]$$

and

```
solx = ∫Coefficient[Expand[ricc1[[1]]], Dt[x]] dx
```

$$\text{Log}[x] + \frac{1}{3}\text{Log}[-2 + u\,x] - \frac{1}{3}\text{Log}[1 + u\,x]$$

Comparing the two integrals, we observe that common terms exist. Thus, the complete solution which is equal to a constant *C*[1] follows:

```
isol = Expand[1/2 ((solx - solu) + solu + solx)] == C[1]
```

$$\text{Log}[x] + \frac{1}{3}\text{Log}[-2 + u\,x] - \frac{1}{3}\text{Log}[1 + u\,x] == C[1]$$

After the collection of logarithmic terms and an exponentiation, we get

```
iisol = Thread[Exp[isol //. a_. Log[b_] + c_. Log[d_] →
      Log[b^a d^c]], Equal]
```

$$\frac{x\,(-2+u\,x)^{1/3}}{(1+u\,x)^{1/3}} == E^{C[1]}$$

The solution of this relation with respect to u reproduces the known result:

```
Solve[iisol, u]
```

$$\left\{\left\{u \to \frac{E^{3\,C[1]} + 2\,x^3}{x\,(-E^{3\,C[1]} + x^3)}\right\}\right\}$$

All these steps to solve a first-order ordinary differential equation are completely algorithmic. That is why we can collect them in a common function called IntegratingFactor[]. The function IntegratingFactor[] needs as input parameters, the equation, the dependent and independent variables, and the infinitesimals. The following lines define this function using the steps of the calculation discussed above:

```
Clear[IntegratingFactor]
IntegratingFactor[equation_, depend_: Symbol,
 independ_: Symbol, xi_, phi_] :=
 Block[{pattern1, pattern2, eq, eqin, q, p,
   ifactor, t1, t2, it1, it2},
  If[equation /. p_. u_^(n_)[t_] + q_ == 0 :> n > 1,
   Return[Hold[IntegratingFactor[equation,
      depend, independ, xi, phi]]]];
  If[FreeQ[equation, Equal],
   Return[Hold[IntegratingFactor[equation,
      depend, independ, xi, phi]]]];
  trafo = depend@@{independ} -> depend;
  itrafo = depend -> depend@@{independ};
  pattern1 = p_. u_'[t_] + q_ == 0 :> -p Dt[u] + q Dt[t];
  pattern2 = p_. u_'[t_] + q_ == 0 :> {q, -p};
  eqin = equation /. trafo; eq = eqin /. pattern1;
  {q, p} = eqin /. pattern2;
  ifactor = 1/(xi q - phi p);
  eqh = Together[eq ifactor];
  num = Numerator[eqh];
  den = Denominator[eqh];
  t1 = Coefficient[num, Dt[independ]]/den;
  t2 = Coefficient[num, Dt[depend]]/den;
  it1 = Integrate[t1 dindependent];
```

```
it2 = ∫ t2 ddepend;
If[FreeQ[it1, Integrate] && FreeQ[it2, Integrate],
 it1 = Expand[1/2 ((it1 - it2) + it1 + it2)],
 Return[Hold[IntegratingFactor[equation,
    depend, independ, xi, phi]]]];
Simplify[it1 == C[1] /. itrafo]     ■
```

The following examples demonstrate the application of the function IntegratingFactor[].

Example 3

The first equation solved by this function is a first-order equation containing a ratio of dependent and independent variables:

```
eq2 = ∂x u[x] - (x^2 + u[x]^2)/(x u[x]) == 0; eq2 // LTF
```

$$-\frac{u^2 + x^2}{u\,x} + u_x == 0$$

Knowing the infinitesimals $\xi = x$ and $\phi = u$, the function IntegratingFactor[] is able to calculate the implicit solution in the form

```
ieq2 = IntegratingFactor[eq2, u, x, x, u]
```

$$\text{Log}[x] - \frac{u[x]^2}{2\,x^2} == C[1]$$

Solving this equation with respect to u, we find that the solution is given by two expressions containing a Log[x] in the radicand of a square root:

```
Solve[ieq2, u[x]]
```

$$\{\{u[x] \to -I\sqrt{2}\ \sqrt{x^2\,C[1] - x^2\,\text{Log}[x]}\},\ \{u[x] \to I\sqrt{2}\ \sqrt{x^2\,C[1] - x^2\,\text{Log}[x]}\}\}$$

Extracting -1 from the constant of integration $C[1]$, we find the same solution as DSolve[] does:

```
DSolve[eq2, u[x], x]
```

$$\{\{u[x] \to -\sqrt{2}\ \sqrt{x^2\,C[1] + x^2\,\text{Log}[x]}\},\ \{u[x] \to \sqrt{2}\ \sqrt{x^2\,C[1] + x^2\,\text{Log}[x]}\}\}$$

□

Example 4

Another example for which an integrating factor exists but the quadrature is impossible is given by

```
eq3 = ∂x u[x] - u[x] (1 - u[x]^2 Exp[u[x]]) == 0; eq3 // LTF
```

$$-u\,(1 - E^{u}\,u^{2}) + u_{x} == 0$$

The function IntegratingFactor[] returns the original input

```
ieq3 = IntegratingFactor[eq3, u, x, 1, 0]
```

```
IntegratingFactor[-u[x] (1 - E^u[x] u[x]^2) + u'[x] == 0, u, x, 1, 0]
```

The function IntegratingFactor[] is simple to use. If the function cannot find an integrating factor or is unable to carry out the integrations, it returns the input line. □

The presented method of an integrating factor for first-order equations can be generalized to higher-order equations. This generalization is further discussed in Section 4.4.2.2 for second-order and in Section 4.4.3.1 for higher-order equations. The method of an integrating factor was only useful in cases where we knew the infinitesimals. At the beginning of this section, we noted that the first-order equations have some peculiarities in determining the symmetries. The following section will discuss how this problem can be partially solved by introducing conformal symmetries or using heuristic ansätze for the infinitesimals.

4.4.1.3 Infinitesimals of First-Order Ordinary Differential Equations

The determination of infinitesimal transformations of first-order ordinary differential equations is a special problem in Lie's theory. The problem is that first-order ordinary differential equations allow an infinite number of symmetries. This property is an essential obstacle in the calculation of the symmetries. The central point in a practical calculation is that the determining equations for the infinitesimals reduce to a first-order partial differential equation. Let us demonstrate this by the general equation of a first-order ODE:

```
ode11 = ∂x u[x] - ω[x, u[x]]; ode11 // LTF
```

$$-\omega + u_{x} == 0$$

Applying the function DeterminingEquations[] of *MathLie* to this ODE, we find a single first-order PDE for the two unknowns ξ_1 and ϕ_1:

```
DeterminingEquations[{ode11}, {u}, {x}, {∂x u[x]}] // LTF
```

$$-\phi_1\,\omega_u - \xi_1\,\omega_x - \omega^2\,(\xi_1)_u - \omega\,(\xi_1)_x + \omega\,(\phi_1)_u + (\phi_1)_x == 0$$

Such an equation has no unique solution in general. Thus, the problem is that we cannot derive an overdetermined system of determining equations allowing us to calculate the symmetries. This is due to the elimination of the first-order derivatives in the prolongation formula.

To solve this difficulty, several approaches are discussed in the literature. The first and original, already discussed by Lie, is to use group classification, i.e., to find families of ODEs that are invariant under the group generated by a particular transformation. Most elementary methods are based on this idea. A more recent idea, suggested by Olver [1986], is to regard the first-order equation as an inappropriate reduction of a second-order ODE which has a solvable non-Abelian Lie algebra. This procedure will lead to hidden symmetries of type I. Hidden symmetries are used by Abraham-Schrauner and Guo [1993] to classify families of ODEs.

Neither of the above methods is helpful when ω is given and the equation *ode11* does not belong to a family of ODEs having known point or hidden symmetries. A way out of this dilemma is a restriction of the admitted symmetries. Olver [1986] and Hydon [1994] introduced the term of a conformal symmetry. The background of this notion is that a vector field $\vec{v} = \sum_{i=1}^{n} \alpha_i\,\partial_{x_i}$ generates a one-parameter group of conformal transformations if

$$\frac{\partial \alpha_i}{\partial x_j} + \frac{\partial \alpha_j}{\partial x_i} = 0 \tag{4.41}$$

and

$$\frac{\partial \alpha_i}{\partial x_i} = \varphi(x) \tag{4.42}$$

for a certain function $\varphi(x)$. The condition for a one-parameter group thus reduces to the Cauchy-Riemann equations

$$\xi_x = \phi_y \quad \text{and} \quad \xi_y = -\phi_x\,. \tag{4.43}$$

In turn, the invariance condition simplifies to

$$\phi_x(1-\omega^2) - \xi\,\omega_x - \phi\,\omega_y = 0. \tag{4.44}$$

This relation is a result of Lie's theorem on first-order differential equations.

Theorem: Symmetries of first-order ODEs

A first-order ODE $u' = \omega(x, u)$ allows a one-parameter group $\xi\,\partial_x + \phi\,\partial_u$ if the relation

$$\frac{d}{dx}\left(\frac{1}{\phi - \omega\xi}\right) + \frac{d}{du}\left(\frac{\omega}{\phi - \omega\xi}\right) = 0 \tag{4.45}$$

or, equivalently,

$$\phi_x + (\phi_u - \xi_x)\,\omega - \xi_u\,\omega^2 - \xi\,\frac{\partial\omega}{\partial x} - \phi\,\frac{\partial\omega}{\partial u} = 0 \tag{4.46}$$

holds for all values of x and u. ○

This theorem was given by Lie (Vol. 3, XIII, Theorem 1, Engel and Heegaard [1912]) to determine the infinitesimals of a first-order ODE.

Introducing in (4.43) the complex variables

$$w = \phi - i\,\xi,\ \ \overline{w} = \phi + i\,\xi, \tag{4.47}$$

and

$$z = x + iu,\ \ \bar{z} = x - iu \tag{4.48}$$

and the real functions

$$\mu(z, \bar{z}) = \arctan\left(\omega\left(\frac{z+\bar{z}}{2}, \frac{z-\bar{z}}{2\,i}\right)\right) \qquad \text{with } \mu \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \tag{4.49}$$

the invariance condition reduces to

$$\mathrm{Im}\left(\frac{i}{2}\,\frac{dw}{dz} + w\,\mu_z\right) = 0. \tag{4.50}$$

The bars over w and z in relations (4.46)–(4.48) denote the complex conjugate values. Subscripts in these relations indicate a differentiation with respect to the variable.

The replacement of the variables w and $\overline{w}$ by the functions

$$\zeta(z) = \int \frac{1}{w(z)}\,dz \ \text{ and } \ \overline{\zeta}(\bar{z}) = \int \frac{1}{\overline{w}(\bar{z})}\,d\bar{z} \tag{4.51}$$

and the derivatives

$$\zeta'(z) = \frac{1}{w(z)} \quad \text{and} \quad \overline{\zeta}'(\bar{z}) = \frac{1}{\overline{w}(\bar{z})} \tag{4.52}$$

allows the representation of the infinitesimals by

$$\xi = \frac{\mathrm{Im}(\zeta'(z))}{|\zeta'(z)|^2} \quad \text{and} \quad \phi = \frac{\mathrm{Re}(\zeta'(z))}{|\zeta'(z)|^2}. \tag{4.53}$$

In addition to the infinitesimals, the canonical variables are defined by

$$s = \frac{\zeta(z) + \overline{\zeta}(\bar{z})}{2} = \int \frac{\phi\, dx - \xi\, du}{\xi^2 + \phi^2} \tag{4.54}$$

and

$$t = \frac{\zeta(z) - \overline{\zeta}(\bar{z})}{2i} = \int \frac{\xi\, dx + \phi\, du}{\xi^2 + \phi^2}. \tag{4.55}$$

Relations (4.45)–(4.53) are helpful to reformulate Lie's theorem on infinitesimal symmetries.

Theorem: Conformal symmetries

A first-order ordinary differential equation $u' = \omega(x, u) = \tan(\mu(x + iu, x - iu))$ allows a one-parameter group of conformal transformations if

$$\mu(z, \bar{z}) = F(r) + \frac{i}{2}\ln(\zeta'(z)) - \frac{i}{2}\ln\overline{\zeta}'(\bar{z}), \tag{4.56}$$

where F is a real function and ζ an analytic complex function. ○

If we know the conformal symmetries, we also know the general solution of the equation. The solution follows either from the theorem on an integrating factor or via canonical variables.

So far, we have discussed symmetries and solution procedures of first-order ODEs. The problem of first-order ODEs was the assessment of the infinitesimal transformation. This problem dissolves if we consider ODEs of higher order. In the following section, we discuss the solution of second-order ODEs by utilizing symmetry methods.

4.4.2 Second-Order Ordinary Differential Equations

Second-order ordinary differential equations are very important for applications in physics and engineering. All equations based on Newton's second law are second-order equations. Thus, mechanics, for example, is mainly based on second-order ordinary differential equations. The integration theory of second-order ODEs was developed by Lie during the years 1871–1874. In 1891, Lie's theory of integration was published by Scheffers and Lie [1891] in *Vorlesungen über Differentialgleichungen mit bekannten infinitesimalen Transformationen*. Scheffers, a student of Lie, assembled all of Lie's work on ordinary differential equations in a beginner's book. In another series of books compiled by Lie and Engel [1888], Lie describes the problem of integrating a differential equation as follows:

"I observed that a large number of ordinary differential equations integrated by older integration methods are invariant under easily derivable classes of transformations. The older integration methods are all based on the transformation properties of the equation. In other words, I realized that the term differential invariant of a finite continuous group is contained implicitly in every textbook on ordinary differential equations. Discovering the connection between transformation groups and older integration strategies I started to develop a general integration theory based on the finite or infinitesimal transformation of the equation. In my investigations it was clear from the beginning that the related transformations always created a group for each case."

We will exemplify Lie's line of thought below. The most general form of a second-order ODE is given by

$$F(x, u, u', u'') = 0, \tag{4.57}$$

where primes denote differentiation with respect to x. For our purposes, we assume that equation (4.57) is solvable with respect to the second-order derivative. Thus, we consider equations in the form

$$u'' = \omega(x, u, u'), \tag{4.58}$$

where ω is a given function of x, u, and u'.

For the general equation (4.58) or (4.57), there exist several procedures to derive the solution. Common to each method is the symmetry of the equation. In contrast to first-order equations, the determination of symmetries is not difficult. However, the problem here is to apply the appropriate solution procedure to a specific equation. In the following, we discuss three methods which allow us to identify the solution of a second-order ODE.

4.4.2.1 Integration by Group Classification

Following the reasoning of Lie, we can solve a second-order ODE if we can classify the group. The idea is the following: If a second-order equation admits a Lie algebra of dimension $r \geq 2$, it can be integrated by a group-theoretic quadrature method. This can be done in various ways, one of which is given by the following algorithm:

1. Compute the admitted Lie algebra L_r. A basis of L_r is the set $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r$.

2. If $r = 2$, go to the next step;
   if $r > 2$, then distinguish any two-dimensional subalgebra L_2 of L_r.
   If $r = 1$, The order of the equation may be lowered;
   if $r = 0$, the group method is not useful.

3. Determine the type of the algebra L_2 obtained by the following table:

I	$[\vec{v}_1, \vec{v}_2] = 0$	$\vec{v}_1 \otimes \vec{v}_2 \neq 0$	$\vec{v}_1 = \partial_x, \quad \vec{v}_2 = \partial_u$	$u'' = f(u')$
II	$[\vec{v}_1, \vec{v}_2] = 0$	$\vec{v}_1 \otimes \vec{v}_2 = 0$	$\vec{v}_1 = \partial_u, \quad \vec{v}_2 = \partial_x$	$u'' = f(x)$
III	$[\vec{v}_1, \vec{v}_2] = \vec{v}_1$	$\vec{v}_1 \otimes \vec{v}_2 \neq 0$	$\vec{v}_1 = \partial_u, \quad \vec{v}_2 = x\,\partial_x + u\,\partial_u$	$u'' = f(u')/x$
IV	$[\vec{v}_1, \vec{v}_2] = \vec{v}_1$	$\vec{v}_1 \otimes \vec{v}_2 = 0$	$\vec{v}_1 = \partial_u, \quad \vec{v}_2 = u\,\partial_u$	$u'' = f(x)\,u'$

(4.59)

   Cases I to IV of the table are identified by computing the commutator $[\vec{v}_1, \vec{v}_2]$ of $\vec{v}_1$ and $\vec{v}_2$, and their pseudo-scalar product $\vec{v}_1 \otimes \vec{v}_2 = \xi_1\,\phi_2 - \phi_1\,\xi_2$. The subscripts of the infinitesimals ξ_i and ϕ_i denote the number of the vector field $\vec{v}_i$. If $[\vec{v}_1, \vec{v}_2]$ is neither 0 nor $\vec{v}_1$, then choose a new basis $\vec{v}_1', \vec{v}_2'$, such that $[\vec{v}_1', \vec{v}_2'] = \vec{v}_1'$.

4. Bring the basis of L_2 into agreement with cases I–IV by going over to canonical variables t, w. Rewrite the equation in canonical variables and integrate it.

5. Rewrite the solution in terms of the original variables.

The stated algorithm is based on the fact that in the complex case, any Lie algebra of dimensionality $r > 2$ has a distinguished two-dimensional subalgebra. However, the structure of a two-dimensional Lie algebra with bases $\vec{v}_1$ and $\vec{v}_2$ can be described in terms of the commutator $[\vec{v}_1, \vec{v}_2] = \vec{v}_1\,\vec{v}_2 - \vec{v}_2\,\vec{v}_1$ and the pseudo-scalar product $\vec{v}_1 \otimes \vec{v}_2 = \xi_1\,\phi_2 - \phi_1\,\xi_2$. For more details compare the work by Scheffers and Lie [1891], Olver [1986], and Bluman and Kumei [1989]. Let us demonstrate these five steps by two examples.

Example 1

The first example considers a second-order ODE. This equation was discussed by Ibragimov [1994] in connection with Lie's group classification. We use the same example here to demonstrate how symbolic calculations with a computer can clarify the solution steps. The equation reads

```
equation = {∂{x,2} u[x] - ∂x u[x] / u[x]^2 + 1 / (u[x] x)}; equation // LTF
```

$$\frac{1}{u\,x} - \frac{u_x}{u^2} + u_{x,x} == 0$$

The first step of the algorithm consists in finding the Lie algebra of the equation. This step is practically revealed by using the package *MathLie*. *MathLie* contains a function designed to determine the infinitesimal transformations. The name of the function is Infinitesimals[] and has a symbolic template of the form $\mathcal{PS}^{\gamma}_{u,x}[\Delta]$. This operator takes the independent and dependent variables as subscripts and the parameters as superscripts. The equation is given as a fourth argument. The equation above is free of any parameters. The determination of the infinitesimals is carried out by

```
infi = PS[{}]{u},{x} [equation]; infi // LTF
```

$$\phi_1 == \frac{1}{2}\,u\,(k1 + 2\,k2\,x)$$
$$\xi_1 == x\,(k1 + k2\,x)$$

The result is a representation of the infinitesimals for the independent and dependent variables. *xi[1]* corresponds to the independent variable x and *phi[1]* to the dependent variable u. It turns out that our equation admits a two-dimensional Lie group. The two parameters k1 and k2 are the group parameters. As discussed in Chapter 2, to each symmetry group there exists a related Lie algebra. We can inspect the structure of this algebra again by applying *MathLie*. The package provides tools to calculate the commutator table and the structure constants. The commutator table is created by

```
LieCommutatorTable[infi, {u}, {x}] // TableForm
```

0	-V[2]
V[2]	0

The result of this calculation shows that the corresponding algebra L_2 belongs to type III of Lie's classification. This becomes obvious if we interchange the vector fields $\vec{v}_i$ and calculate the pseudo-scalar product of the infinitesimals

$$\vec{v}_1 \otimes \vec{v}_2 = \xi_1\,\phi_2 - \phi_1\,\xi_2 = \frac{u\,x^2}{2} \neq 0\,.$$

The calculation of the pseudo-scalar product in *Mathematica* needs the lines

```
infi1 = {xi[1][x, u], phi[1][x, u]} /. infi /. {k1 → 1, k2 → 0};

infi2 = {xi[1][x, u], phi[1][x, u]} /. infi /. {k1 → 0, k2 → 1};

pseudoScalarProduct = infi1[[1]] infi2[[2]] - infi1[[2]] infi2[[1]]
```

$$\frac{u\,x^2}{2}$$

In step 4 of the integration algorithm, we introduce canonical variables which have to satisfy the conditions $\vec{v}_1(t) = 0$ and $\vec{v}_1(u) = 1$. Solving the related characteristic equations by conducting the function CanonicalVariables[], we end up with transformations probably simplifying the original equation. In the following *Mathematica* line, the first two arguments, $\{u\}$ and $\{x\}$, denote the dependent and independent variables, and the next two, $\{x\}$ and $\left\{\frac{u}{2}\right\}$, are the infinitesimals for k1 = 1 and k2 = 0 of the independent and dependent variables, respectively. The last pair $\{w\}$ and $\{t\}$ are the new dependent and independent canonical variables, respectively:

```
substitution =
 CanonicalVariables[{u}, {x}, {x}, {u/2}, {w}, {t}];
substitution // LTF
```

$$t == \text{Log}[x]$$

$$w == \frac{u}{\sqrt{x}}$$

The result belongs to the subgroup with k1 = 1 and k2 = 0. The second set of transformations follows by the choice k1 = 0 and k2 = 1:

```
secondtransformation =
CanonicalVariables[{u}, {x}, {x^2}, {x u}, {w}, {t}];
secondtransformation // LTF
```

$$t == -\frac{1}{x}$$

$$w == \frac{u}{x}$$

The next step in the algorithm is to use these transformations to change the representation of the equation. For this kind of calculation, *MathLie* offers the function CanonicalRepresentation[]. This function needs the input of the original equation, the canonical transformations given by rules, and the target variables. The first set of canonical variables descending from the subgroup with k1 = 1 and k2 = 0 leads to the reduced equation

```
canonical1 = CanonicalRepresentation[
  equation[[1]], u, x, {w → u/Sqrt[x], t → Log[x]}, w, t];
canonical1 // LTF
```

$$2\,w - w^3 - 4\,w_t + 4\,w^2\,w_{t,t} == 0$$

At this stage of the calculation, we can use the *Mathematica* function DSolve[] to solve the second-order equation. We note that the achieved equation is as complicated as the original equation and, thus, DSolve[] may fail to find a solution. In fact, we get

```
sol1 = DSolve[canonical1, w, t]
```

```
DSolve[2 w[t] - w[t]^3 - 4 w'[t] + 4 w[t]^2 w''[t] == 0, w, t]
```

However, using the second set of canonical variables, we discover the reduction

```
canonical2 = CanonicalRepresentation[
  equation[[1]], u, x, {w → u/x, t → -1/x}, w, t];
canonical2 // LTF
```

$$t^3\left(\frac{w_t}{w^2} - w_{t,t}\right) == 0$$

which looks much simpler. This simplification is one aim of symmetry analysis. Exerting the *Mathematica* function DSolve[] to this equation, we acquire

```
sol2 = DSolve[canonical2, w, t]
```

```
InverseFunction::ifun : Warning: Inverse functions are
   being used. Values may be lost for  multivalued inverses.
```

$$\left\{\left\{w \to \left(\frac{1 + \text{ProductLog}\left[E^{-1+C[1]^2\,(\#1-C[2])}\right]}{C[1]}\&\right)\right\}\right\}$$

As expected, the solution of the equation in canonical variables is given by a function containing two constants of integration. The ProductLog[] function depicts the solution for w in $z = we^w$. The function is a generalization of a logarithm. It can be used to represent solutions to a variety of transcendental equations.

The last step of the integration procedure by Lie is the inversion of the transformation. This back substitution of the canonical variables is also supported by the package *MathLie*. The related function is BackTrafoCanonical[]. For the inverse canonical transformation, we need the solution in canonical variables, the set of canonical variables themselves, the original variables, and the transformation

between the two sets of variables. Knowing these quantities, we are able to apply the function to the solution:

```
solu = BackTrafoCanonical[sol2, w, t, u, x, {u → w x, t → -1 / x}]
```

$$\left\{u \to \text{Function}\left[x, \frac{x\left(1 + \text{ProductLog}\left[E^{-1+C[1]^2\left(-\frac{1}{x}-C[2]\right)}\right]\right)}{C[1]}\right]\right\}$$

The derived solution satisfies the original equation. We can check this by inserting the solution into the equation:

```
equation /. solu // FullSimplify
```

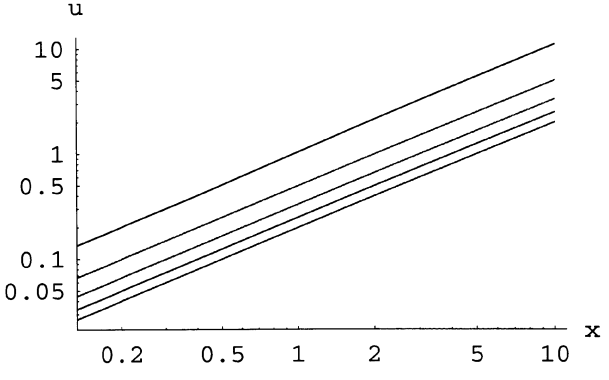
```
{0}
```

The result confirms the solution. Since the solution contains a special function, we have no clear idea of the graph of this function. We can graphically represent the solution by specifying the constants of integration. For a set of five constants $C[1]$ at fixed $C[2]$, we create a table containing the different solutions:

```
soluC = Table[u[x] /. solu /. {C[1] → ϵ, C[2] → 1}, {ϵ, 1, 5}];
```

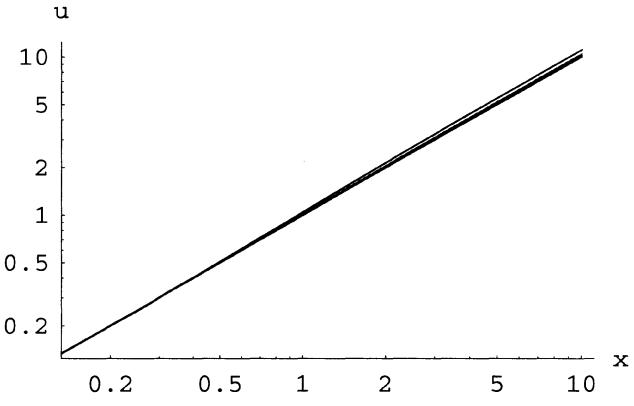
A LogLog plot shows that the solutions with fixed $C[2]$ have a common slope:

```
<< Graphics`Graphics`

LogLogPlot[Evaluate[soluC], {x, 0.001, 10}, PlotStyle →
  {RGBColor[0, 0, 0.250004], RGBColor[0.996109, 0, 0],
   RGBColor[0, 0.500008, 0], RGBColor[0.500008, 0, 0.250004],
   RGBColor[0.700008, 0, 0]},
 AxesLabel → {"x", "u"}]
```



Varying $C[2]$ we observe only a minor change in the slope:

```
soluC = Table[u[x] /. solu /. {C[1] → 1, C[2] → y}, {y, 1, 5, 1}];

LogLogPlot[Evaluate[soluC], {x, 0.001, 10}, PlotStyle →
  {RGBColor[0, 0, 0.250004], RGBColor[0.996109, 0, 0],
   RGBColor[0, 0.500008, 0], RGBColor[0.500008, 0, 0.250004],
   RGBColor[0.700008, 0, 0]},
 AxesLabel → {"x", "u"}]
```



In conclusion, we see that Lie's algorithm of group classification is straightforward to derive explicit solutions of a second-order ODE. All the steps needed to carry out the calculation are supported by *MathLie*. Thus, it is fairly easy to construct a solution. The next example discusses the solution procedure for a more complicated equation.

Example 2

The second example is connected with kinetics and heat transfer (cf. Ames [1968]). Ames' equation also occurs in certain other problems like vortex motion of incompressible fluids, in the theory of the space charge of elasticity around a glowing wire, and in the nebular theory for the mass distribution of gaseous interstellar material under the influence of its own gravitational field.

We concentrate our attention on the one-dimensional representations of these problems. The equation is discussed in cylindrical coordinates. For this special case, the equation reduces to a second-order ordinary differential equation (Ames [1968]) given by

```
ames = ∂x,x u[x] + ∂x u[x] / x + α Exp[u[x]]; ames == 0 // LTF
```

$$E^u\ \alpha + \frac{u_x}{x} + u_{x,x} == 0$$

When trying to solve this simple equation by DSolve[], we end up with the dissatisfying result

```
DSolve[ames == 0, u, x]
```

$$\text{DSolve}\left[E^{u[x]}\ \alpha + \frac{u'[x]}{x} + u''[x] == 0,\ u,\ x\right]$$

Thus, *Mathematica* is unable to find the solution. However, we are currently discussing a constructive procedure to derive solutions of second-order equations. Thus, a solution should be accessible if we know the symmetry transformations. The symmetries of the equation are calculated by the function Infinitesimals[] or the operator $\mathcal{PS}^{\gamma}_{u,x}[\Delta]$. The application of this operator provides

```
PS{α}{u},{x} [ames] // LTF
```

$$\phi_1 == k2 + k1\ \text{Log}[x]$$
$$\xi_1 == -\frac{1}{2}\ x\ (-k1 + k2 + k1\ \text{Log}[x])$$

a two-dimensional symmetry group with group parameters k1 and k2. The subgroups created by the parameters k1 and k2 are the cornerstones of the integration process. Let us consider the subgroup related to k1 = 0 and k2 = 1. This choice of the group parameters allows to derive the canonical variables w and t to be

```
ctransformation =
CanonicalVariables[{u}, {x}, {-x / 2}, {1}, {w}, {t}];
ctransformation // LTF
```

```
t == -2 Log[x]
w == u + 2 Log[x]
```

Converting the original equation *ames* into the new coordinates simplifies the representation of the equation:

```
canonical = CanonicalRepresentation[
  ames, u, x, {w → u + 2 Log[x], t → -2 Log[x]}, w, t];
canonical // LTF
```

$$E^t\ (E^w\ \alpha + 4\ w_{t,t}) == 0$$

The introduction of the canonical variables allows the elimination of the term containing first derivatives. As a result, the original equation was simplified. Trying again DSolve[] to disclose the solution in canonical coordinates,

```
csol = DSolve[canonical, w, t]
```

$$\left\{\left\{w \to \left(\text{Log}\left[\frac{1}{\alpha}\left(C[1] - C[1]\,\text{Tanh}\left[\frac{1}{4}\left(\sqrt{2}\,\#1\,\sqrt{C[1]} - \sqrt{2}\,\sqrt{C[1]}\,C[2]\right)\right]^2\right)\right]\&\right)\right\},\right.$$
$$\left.\left\{w \to \left(\text{Log}\left[\frac{1}{\alpha}\left(C[1] - C[1]\,\text{Tanh}\left[\frac{1}{4}\left(-\sqrt{2}\,\#1\,\sqrt{C[1]} + \sqrt{2}\,\sqrt{C[1]}\,C[2]\right)\right]^2\right)\right]\&\right)\right\}\right\}$$

The inversion of the canonical transformation provides the solutions in the original variables u and x:

```
sol1 = BackTrafoCanonical[sol, w, t, u, x,
  {u → w - 2 Log[x] , t → -2 Log[x]}]
```

$$\left\{u \to \text{Function}\left[x, -2\,\text{Log}[x] + \text{Log}\left[\frac{1}{\alpha}\left(C[1] - C[1]\right.\right.\right.$$
$$\left.\left.\left.\text{Tanh}\left[\frac{1}{4}\left(-\sqrt{2}\,\sqrt{C[1]}\,C[2] - 2\sqrt{2}\,\sqrt{C[1]}\,\text{Log}[x]\right)\right]\text{^}2\right)\right]\right],$$
$$u \to \text{Function}\left[x, -2\,\text{Log}[x] + \text{Log}\left[\frac{1}{\alpha}\left(C[1] - \right.\right.\right.$$
$$\left.\left.\left.C[1]\,\text{Tanh}\left[\frac{1}{4}\left(\sqrt{2}\,\sqrt{C[1]}\,C[2] + 2\sqrt{2}\,\sqrt{C[1]}\,\text{Log}[x]\right)\right]\text{^}2\right)\right]\right]\right\}$$

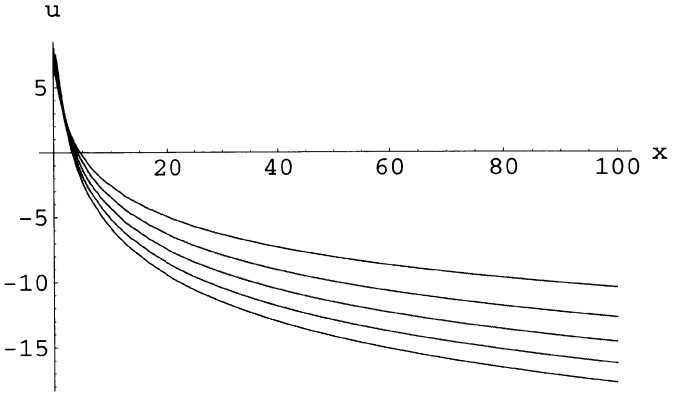
Inserting the derived solutions into the original equation *ames,* we can check the solution

```
ames /. sol1 // Simplify
```

```
0
```

To get an impression of the solution, we plot it for a set of constants $C[1]$ at fixed $C[2]$ and α.

```
Plot[Evaluate[Table[u[x] /. sol1[[1]] /.
    {C[1] → i, C[2] → 1, α → 1/100}, {i, 1, 5, 1}]],
 {x, .1, 100}, PlotStyle → RGBColor[0.996109, 0, 0],
 AxesLabel → {"x", "u"}]
```



This example shows that solutions of a second-order ordinary differential equation are easy to derive if we know the symmetries of the equation.

Actually, we did not use in our calculations the complete theory of Lie discussed above. In this second example, we only used the existence of a certain symmetry. This symmetry is sufficient to determine the corresponding canonical variables. Thus, a canonical transformation can be carried out independently of the classification scheme by Lie. The canonical transformation of the equation into new variables simplified the representation. In both examples, this simplification was the main step toward the solution. However, in Lie's theory, there exists a more efficient way to detect the solvability of the equation. In turn, there is a procedure which reduces the complete calculations to quadratures. The following section will discuss this procedure in detail.

4.4.2.2 The Integrating Factor Method

In Section 4.4.1.2, we discussed the method of an integrating factor for a first-order ODE. We remarked that this method has a generalization to higher-order equations. In this section, we generalize the method to second-order equations. The main result of this procedure is that a second-order equation can be solved by pure quadratures if the equation possesses an appropriate number of symmetries. The solution procedure based on integrating factors is completely algorithmic. The algorithm consists of five steps which Lie discussed in his numerous papers (cf. Engel and Heegaard [1912]).

According to Lie, we can state the integration procedure in the following way: If a second-order equation admits a finite symmetry of dimension $r \geq 2$, it can be integrated with a group-theoretic quadrature method by

1. Computing the Lie algebra L_r. A basis of L_r is the set $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r$. The tangent vector fields $\vec{v}_i$ follow by appropriately specifying the group constants.

2. If $r = 2$, go to step 3;
   if $r > 2$, distinguish any two-dimensional subalgebra L_2 of L_r.
   If $r = 1$, the order of the equation may be lowered;
   if $r = 0$, the group method is not useful.

3. Calculate the Lie determinants $d\Delta_i$ and Δ and determine the two first integrals by integration. The Lie determinants are defined by

$$d\Delta_1 = \det\begin{pmatrix} dx & du & du' \\ \xi_2 & \phi_2 & \phi_2^{'} \\ 1 & u' & \omega \end{pmatrix}, \qquad d\Delta_2 = \det\begin{pmatrix} \xi_1 & \phi_1 & \phi_1{}' \\ dx & du & du' \\ 1 & u' & \omega \end{pmatrix} \tag{4.60}$$

   and

$$\Delta = \det\begin{pmatrix} \xi_1 & \phi_1 & \phi_1^{'} \\ \xi_2 & \phi_2 & \phi_2^{'} \\ 1 & u' & \omega \end{pmatrix}, \tag{4.61}$$

   where ξ_i and ϕ_i are the infinitesimals corresponding to the vector field $\vec{v}_i$, and $\phi_i^{'}$ denote the first extensions of the infinitesimals ϕ_i. The first integrals ψ_i related to the Lie determinants $d\Delta_i$ are given by

$$\psi_i = \int \frac{d\Delta_i}{\Delta} = c_i, \qquad i = 1, 2. \tag{4.62}$$

   The constants c_i denote the integration constants of the ODE.

4. Solve one of the two integrals with respect to the first-order derivative and substitute the result into the remaining relation.

5. If we can solve the resulting relation from step 4 with respect to the unknown function u, we end up with an explicit solution. Otherwise, we get the solution in an implicit form.

These five steps are implemented in the package *MathLie*. The functions of *MathLie* carry out the necessary calculations automatically. To show how the solution procedure works interactively, we demonstrate the algorithm by two examples. The functions needed for the interactive calculation are Infinitesimals[] for the determination of the symmetries, the function SecondOrderAlgebras[] for the extraction of the second-order subalgebras, the functions DeltaMatrix[], and the FirstIntegral[], which are responsible for the determination of the integrals.

Example 1

The first example considers a non-linear second-order equation in which the nonlinearity is given by the square of the first derivative. This non-linear term is multiplied by a real constant α:

```
firstExample = ∂{x,2} u[x] - α (∂x u[x])^2; firstExample // LTF
```

$$-\alpha\, u_x^2 + u_{x,x} == 0$$

The infinitesimal symmetries are derived by the function Infinitesimals[], which is part of the package *MathLie.*

```
infi = Infinitesimals[firstExample,
  u, x, {α}, SubstitutionRules → {∂{x,2} u[x]}];
infi // LTF
```

$$\phi_1 == \frac{-E^{-u\,\alpha}\, k6 + E^{u\,\alpha}\, k7 + E^{u\,\alpha}\, k1\, x + k8\,\alpha + k2\, x\,\alpha}{\alpha}$$

$$\xi_1 == k4 + (k5 + E^{-u\,\alpha}\, k6)\, x - \frac{E^{-u\,\alpha}\, k3}{\alpha} - k2\, x^2\, \alpha$$

The option SubstitutionRules is set to the second-order derivative $u_{x,x}$ to help Infinitesimals[] to find the side conditions more easily. The result is a symmetry group with eight group parameters k_i, i = 1,2,...8. The infinitesimals ξ_1 = xi[1] and ϕ_1 = phi[1] are represented in a pure function form. The group constants *ki* characterize the symmetries of the equation. We note that this eight-parameter group is the largest group a second-order ODE can have. Lie proved that such an equation allows a transformation reducing the original equation to the simple form $y'' = 0$. This reduction is always possible if a second-order equation allows a symmetry group of order eight (Scheffers and Lie [1891]). Thus, the above equation should be solvable.

To detect that the non-linear second-order ODE is solvable, we examine the algebraic properties of the corresponding Lie algebra. If we can find a solvable subgroup of order two in the eight-dimensional algebra, we succeeded. This argument is based on the fact that all second-order Lie algebras are solvable. To detect all the solvable subgroups, we apply the function SecondOrderAlgebras[] to the infinitesimals. This function determines all the second-order solvable subalgebras and represents them by a set of rules for the group constants:

```
secAlgebras = SecondOrderAlgebras[infi, {u}, {x}, {α}]
```

```
{{{k1 → 1}, {k2 → 1}}, {{k1 → 1}, {k5 → 1}}, {{k1 → 1}, {k7 → 1}},
 {{k1 → 1}, {k8 → 1}}, {{k2 → 1}, {k5 → 1}}, {{k2 → 1}, {k6 → 1}},
 {{k2 → 1}, {k8 → 1}}, {{k3 → 1}, {k4 → 1}}, {{k3 → 1}, {k5 → 1}},
```

```
{{k3 → 1}, {k6 → 1}}, {{k3 → 1}, {k8 → 1}}, {{k4 → 1}, {k5 → 1}},
{{k4 → 1}, {k7 → 1}}, {{k4 → 1}, {k8 → 1}}, {{k5 → 1}, {k6 → 1}},
{{k5 → 1}, {k7 → 1}}, {{k5 → 1}, {k8 → 1}}, {{k6 → 1}, {k8 → 1}},
{{k7 → 1}, {k8 → 1}}}
```

The function SecondOrderAlgebras[] returns a list containing substitution rules for second-order algebras. The input of the function are the infinitesimals, the dependent and independent variables, and the parameters of the equation. This set of rules is useful in selecting one of the possible two-dimensional solvable subalgebras which will serve to solve the equation. For the following calculation, we select the seventh rule to represent the set of infinitesimals by

```
infhelp = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /. k8 → β /.
     secAlgebras[[7]] /. {k1 → 0, k2 → 0, k3 → 0,
     k4 → 0, k5 → 0, k6 → 0, k7 → 0, k8 → 0} /. u → u[x] //
  Simplify
```

$$\{\{\{-x^2\,\alpha\}, \{x+\beta\}\}, \{\{0\}, \{\beta\}\}\}$$

Actually, we changed the subgroup by choosing the group constant k8 to be an arbitrary constant β. In addition to the infinitesimals of the subgroup, we need the Lie matrix for the integration. The function DeltaMatrix[] serves to create this kind of matrix, which is defined by relation (4.61).

Lie's matrix, part of the integrating factor, is calculated by the function DeltaMatrix[]. This function needs information on the independent and dependent variables on the right-hand side ω of the ODE, the order of the ODE, and the selected subgroup of the algebra:

```
Δmatrix = DeltaMatrix[x, u, α (∂x u[x])^2, 2, infhelp ];
TableForm[Δmatrix] // LieTraditionalForm
```

$$\begin{array}{lll} -x^2\,\alpha & x+\beta & 1+2\,x\,\alpha\,u_x \\ 0 & \beta & 0 \\ 1 & u_x & \alpha\,u_x^2 \end{array}$$

The result is a 3×3 matrix containing the infinitesimals, the first prolongation of the two subgroups, and the right-hand side of the equation. The determinant of this matrix

```
Det[Δmatrix] // LieTraditionalForm
```

$$-\beta - 2\,x\,\alpha\,\beta\,u_x - x^2\,\alpha^2\,\beta\,u_x^2$$

is a non-vanishing expression containing a second-order polynomial of first derivatives. The coefficients of this polynomial depend on the independent variable x and the parameters α and β. Inserting the Lie matrix into equation (4.62), we are able to calculate the first integrals of the non-linear ODE

```
integrals =
 Thread[FirstIntegral[x, u, Δmatrix, {1, 2}] == {c1, c2}];
integrals // LieTraditionalForm // TableForm
```

$$\frac{1}{x\alpha} - \frac{1}{x\alpha\,(1 + x\alpha\,u_x)} == c1$$

$$-\frac{1}{x\alpha} + \frac{u}{\beta} - \frac{\mathrm{Log}[1 + x\alpha\,u_x]}{\alpha\beta} + \frac{1}{x\alpha\,(1 + x\alpha\,u_x)} == c2$$

The result contains two expressions for the integrals combining the variable u and its first-order derivative in an algebraic way. These two first integrals define two surfaces in the space (x, u, u'). The projection of the intersection of these two surfaces onto the (x, u)-plane defines the solution for which we are looking. The following figure represents a case with fixed values c_1 and c_2.

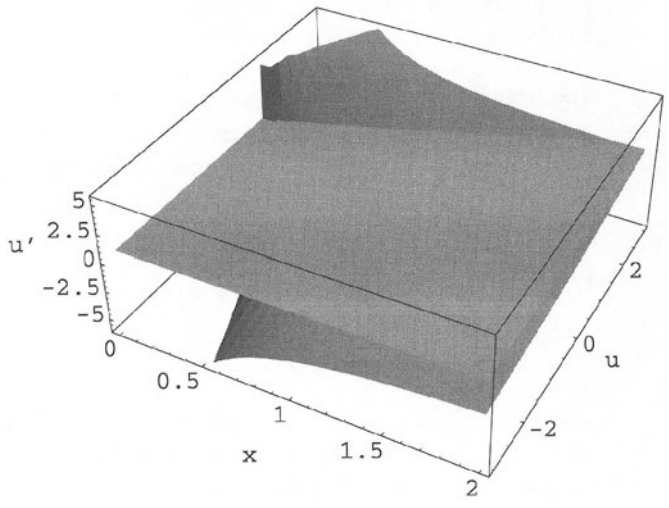


Figure 4.1. Intersection of the two integrals for $c1 = c2 = 1$. The parameters α and β take the values $\alpha = 1/10$ and β=1. The intersecting line represents the solution of the equation if we project the intersection to the (x, u)-plane.

Since we know two first integrals, we are able to eliminate the derivatives from the two integrals analytically. We solve the first relation with respect to u':

```
sol1 = Solve[integrals[[1]], u'[x]] // Simplify
```

$$\left\{\left\{u'[x] \to \frac{-1 + E^{\alpha\,(-c1\,\beta+u[x])}}{x\,\alpha}\right\}\right\}$$

Substituting this result into both integral expressions delivers

```
int21 = integrals /. sol1[[1, 1]] // PowerExpand // Simplify
```

$$\left\{\text{True},\ \frac{1 - E^{c1\,\alpha\,\beta-\alpha\,u[x]}}{x\,\alpha} == c2\right\}$$

The resulting list contains the identity and an implicit representation of the solution. The explicit solution follows from the second relation if we solve it with respect to u:

```
solution = Solve[int21[[2]], u[x]]
```

$$\left\{\left\{u[x] \to -\frac{-c1\,\alpha\,\beta + \text{Log}\left[x\,\left(-c2 + \frac{1}{x\,\alpha}\right)\,\alpha\right]}{\alpha}\right\}\right\}$$

The constants c_1 and c_2 denote constants of integration. α and β are the model parameter and the introduced group parameter, respectively. The occurrence of the group parameter β as a multiplier reminds us of the fact that the additive integration constant c_1 is related to the group of translations.

The solution steps so far discussed are collected in the *MathLie* function SecondOrderIntegrate[]. The application of this function to a second-order ODE is similar to the use of DSolve[]. The function needs the equation under discussion, the dependent and independent variables, and the parameters contained in the ODE. The example discussed above is solved by

```
sol = SecondOrderIntegrate[firstExample, u, x, {α}]
```

$$\left\{u \to \text{Function}\left[x, -\frac{\text{Log}[-x\,CI[1] - CI[2]]}{\alpha}\right]\right\}$$

where *CI*[*1*] and *CI*[*2*] are constants of integration. The solution obtained looks different in comparison with the solution presented above. However, the extraction of the multiplier *CI*[*1*] and a rescaling of *CI*[*2*] will create the same representation of the solution. The same solution as found by the manual calculation is derived by the function DSolve[]:

```
DSolve[firstExample == 0, u, x]
```

$$\left\{\left\{u \to \left(C[2] - \frac{\text{Log}[\#1\,\alpha - C[1]]}{\alpha}\&\right)\right\}\right\}$$

Again, a constant is extracted from the argument of the logarithm. This example shows that the solution steps of Lie's method of first integrals result in the same solution as that of *Mathematica*. □

Example 2

The second example for a second-order ODE is related to the problem of a suspended cable equation:

$$u_{x,x} - \alpha (1 - u_x^2)^{1/2} = 0. \tag{4.63}$$

The problem of the suspended cable is discussed by Ames [1968]. We examine here a generalization of the cable equation by introducing an arbitrary power ν in the second term of the ODE:

$$u_{x,x} - \alpha (1 - u_x^2)^{\nu} = 0. \tag{4.64}$$

The original model follows from our model with $\nu = 1/2$. The present problem is similar to the first example we discussed. The difference is that we added α to the square of the derivative and raised the second term to the νth power. These small changes lead to substantial variations in the solution:

```
secondExample = ∂x,x u[x] - α (1 + (∂x u[x])^2)^ν == 0;
secondExample // LTF
```

$$-\alpha\,(1 + u_x^2)^{\nu} + u_{x,x} == 0$$

In applying Lie's algorithm to this equation, we first calculate the point symmetries of this equation for arbitrary ν.

The infinitesimals follow by applying the *MathLie* function Infinitesimals[] to the equation:

```
infi = Infinitesimals[secondExample,
  u, x, {α, ν}, SubstitutionRules → {∂{x,2} u[x]}];
infi // LTF
```

$$\phi_1 == k1$$
$$\xi_1 == k2$$

The result is a two-dimensional symmetry transformation which itself is solvable. The symmetries represent translations in the independent and dependent variables. The first difference in comparison to Example 1 is that the group is smaller. This reduction of the group order has consequences with regard to the solutions.

The two subgroups necessary for integration are derived by independently setting each of the group parameters *k1* or *k2* to unity. Since the symmetry group is two dimensional, we know from Lie that the algebra is always solvable. In the worst case, the solution may in the end be represented by an implicit representation. The infinitesimals for the two subgroups follow from

```
inf1 =
 {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /. {k1 → 1, k2 → 0} /.
  u → u[x];

inf2 =
 {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /. {k1 → 0, k2 → 1} /.
  u → u[x];
```

So far, we put no restrictions on the exponent ν. The following derivation of the solution however assumes a specific value for ν. We arbitrarily choose $\nu = 3$. The related Lie matrix for this case is calculated by

```
Δmatrix = DeltaMatrix[x, u, α (1 + (∂x u[x])^2)^3 , 2, {inf1, inf2}];
TableForm[Δmatrix] // LieTraditionalForm
```

$$\begin{array}{lll} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & u_x & \alpha\,(1+u_x^2)^3 \end{array}$$

The determinant of this matrix is given by a polynomial of sixth order in u':

```
Det[Δmatrix] // LieTraditionalForm
```

$$-\alpha - 3\,\alpha\,u_x^2 - 3\,\alpha\,u_x^4 - \alpha\,u_x^6$$

The first integrals for the generalized cable equation with $\nu = 3$ follow from

```
integrals =
 Thread[FirstIntegral[x, u, Δmatrix, {1, 2}] == {c1, c2}];
integrals // LieTraditionalForm // TableForm
```

$$u + \frac{1}{4\,\alpha\,(1+u_x^2)^2} == c1$$

$$x - \frac{3\,\text{ArcTan}[u_x]}{8\,\alpha} - \frac{u_x}{4\,\alpha\,(1+u_x^2)^2} - \frac{3\,u_x}{8\,\alpha\,(1+u_x^2)} == c2$$

Solving the first relation with respect to u' allows us to eliminate this term in the second integral:

```
sol1 = Solve[integrals[[1]], u'[x]] // Simplify
```

$$\left\{\left\{u'[x] \to -\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1 + 2\,u[x]}}\right\},\right.$$

$$\left\{u'[x] \to \sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1 + 2\,u[x]}}\right\},$$

$$\left\{u'[x] \to -\frac{\sqrt{\frac{-2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} + 2\,u[x]}{c1-u[x]}}}{\sqrt{2}}\right\},\ \left\{u'[x] \to \frac{\sqrt{\frac{-2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} + 2\,u[x]}{c1-u[x]}}}{\sqrt{2}}\right\}\Bigg\}$$

The solution for u' consists of four expressions containing square roots of u. The differences in the four solutions are the signs in front of the first and second terms. Inserting, for example, the second solution into the second first integral, we get the final solution in an implicit representation:

```
int21 = integrals[[2]] /. sol1[[2, 1]]
```

$$x - \frac{3\,\text{ArcTan}\left[\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}\right]}{8\,\alpha} - \frac{\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}}{4\,\alpha\left(1 + \frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}\right)^2} - \frac{3\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}}{8\,\alpha\left(1 + \frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}\right)} ==$$

$$c2$$

An explicit solution of this expression is impossible since it contains transcendental functions:

```
solution = Solve[int21, u[x]]
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$\text{Solve}\left[x - \frac{3\,\text{ArcTan}\left[\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}\right]}{8\,\alpha} - \frac{\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}}{4\,\alpha\left(1 + \frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}\right)^2} - \frac{3\sqrt{\frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}}}{8\,\alpha\left(1 + \frac{2\,c1 + \frac{\sqrt{c1-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,c1+2\,u[x]}\right)} == c2,\ u[x]\right]$$

At this point, we have to accept that the solution can only be represented in an implicit form. The solution of the original equation follows in one shot by

```
sol = SecondOrderIntegrate[secondExample /. v → 3, u, x, {α}]

Solve::tdep :
 The equations appear to involve transcendental functions
   of the  variables in an essentially non-algebraic way.
```

$$\frac{1}{8\alpha}\left(8x\alpha + 3\,\text{ArcTan}\left[\sqrt{\frac{2\,CI[1] + \frac{\sqrt{CI[1]-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-2\,CI[1] + 2\,u[x]}}\right] + 4\alpha\,CI[1]\sqrt{\frac{4\,CI[1] + \frac{2\sqrt{CI[1]-u[x]}}{\sqrt{\alpha}} - 4\,u[x]}{-CI[1] + u[x]}} - 4\alpha\,u[x]\sqrt{\frac{4\,CI[1] + \frac{2\sqrt{CI[1]-u[x]}}{\sqrt{\alpha}} - 4\,u[x]}{-CI[1] + u[x]}} - 3\sqrt{2}\,\sqrt{\alpha}\,\sqrt{CI[1]-u[x]}\sqrt{\frac{2\,CI[1] + \frac{\sqrt{CI[1]-u[x]}}{\sqrt{\alpha}} - 2\,u[x]}{-CI[1] + u[x]}}\right) == CI[2]$$

The result is an implicit representation of the solution. *CI*[*1*] and *CI*[*2*] are the constants of integration. Trying to solve the original equation by *Mathematica*, we learn that DSolve[] is not capable of resolving the relation for the first integral. A glance at the result of DSolve[] explains the reason:

```
DSolve[secondExample /. v → 3, u, x]

Solve::dinv : The expression (1 - I <<30>>[x])^(<<30>>[x]^2 (2+<<1>>^2))
   involves unknowns in more than one argument, so
   inverse functions cannot be used.

Solve::dinv : The expression (1 - I <<30>>[x])^(<<30>>[x]^2 (2+<<1>>^2))
   involves unknowns in more than one argument, so
   inverse functions cannot be used.

Solve::dinv : The expression (1 - I u'[x])^(u'[x]^2 (2+<<1>>^2))
   involves unknowns in more than one argument, so
   inverse functions cannot be used.
```

```
General::stop : Further output of Solve::dinv will
   be suppressed during this calculation.
{Solve[(3 ArcTan[u'[#1]] - 8 α #1 +
      5 u'[#1] + 6 ArcTan[u'[#1]] u'[#1]^2 - 16 α #1 u'[#1]^2 +
      3 u'[#1]^3 + 3 ArcTan[u'[#1]] u'[#1]^4 - 8 α #1 u'[#1]^4) /
    (8 (1 + u'[#1]^2)^2) == C[1],
  {u'[#1]}]}
```

The above result shows that a transcendental function is given in an essential non-algebraic way. The steps presented above demonstrate that the solution of the generalized cable equation is solvable in an implicit form. □

The two examples demonstrate that the technique of an integrating factor can be generalized to second-order equations. We also realize that the presented procedure is capable of deriving solutions for cases in which *Mathematica* fails. In Section 4.4.3.1, we will discuss the extension of the integrating factor technique to a general *n*th-order ODE. Lie called this procedure the *method of generalized multipliers.* In the following section, we discuss another solution procedure helpful in solving second-order ODEs. This method is related to canonical variables and the skeleton introduced in Sections 4.3.4 and 4.4.1.1 for first-order ODEs. The following method uses the canonical variables to integrate the equation.

4.4.2.2 Method of Canonical Variables

Here, we demonstrate by a single example that the term skeleton is also useful for the case of second-order equations. The combination of canonical variables and the method of first integrals serves to derive an explicit solution for an ODE for which only an implicit representation of the solution is known (cf. Ibragimov [1994]). The considerations serve to demonstrate that a proper combination of different methods will lead to a solution of the ODE.

Example 1

Again, we use the equation from Example 1 of Section 4.4.2.1. This second-order ODE serves to show how canonical variables simplify the skeleton and the solution steps. The specific example we discuss is given by the equation

```
firstExample = ∂x,x u[x] - (∂x u[x])/u[x]^2 + 1/(x u[x]);
firstExample // LTF
```

$$\frac{1}{u\,x} - \frac{u_x}{u^2} + u_{x,x} == 0$$

This second-order equation admits the symmetries

```
infinites = Infinitesimals[
  firstExample, u, x, SubstitutionRules → {∂x,x u[x]}];
infinites // LTF
```

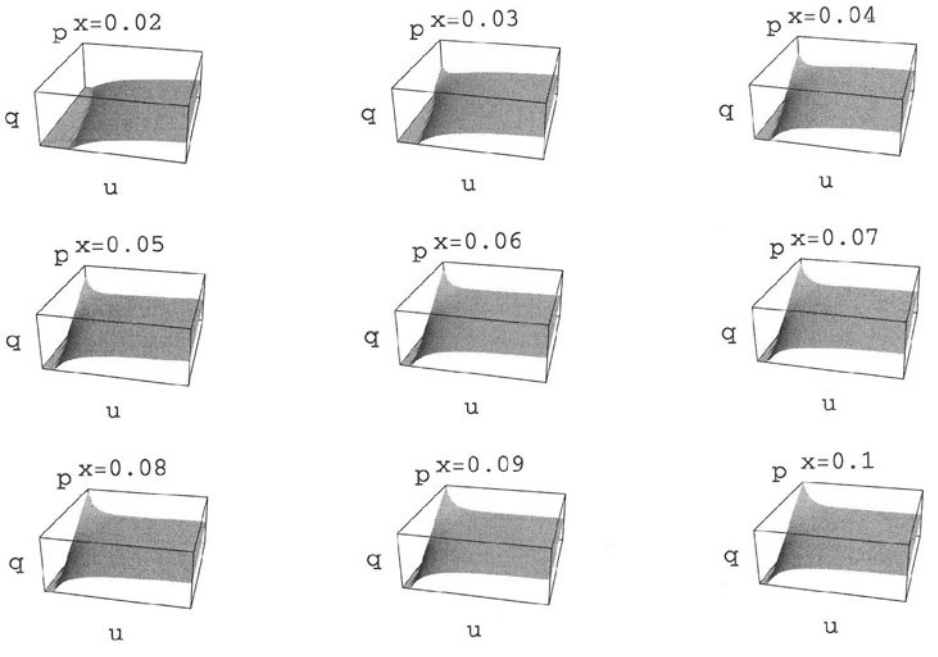
$$\phi_1 == \frac{1}{2}\,u\,(k1 + 2\,k2\,x)$$
$$\xi_1 == x\,(k1 + k2\,x)$$

representing a two-dimensional group of scaling and projections. The skeleton of this equation exists in a four-dimensional manifold $\mathfrak{M} = \{x, u, u' = p, u'' = q\}$. Since the dimension of the manifold $\mathfrak{M}$ is larger than three, we cannot directly represent the skeleton as a surface. However, *Mathematica* with its animation capabilities offers the opportunity to represent the fourth dimension in a sequence of figures. The combination of these figures in an animation allows us to represent the manifold $\mathfrak{M}$ in a special way if one of the coordinates of $\mathfrak{M}$ is smoothly changed. The resulting sequence creates the impression of an evolution of the manifold if one moves along the distinguished coordinate. For the above equation, we define the skeleton in the form

```
skeleton[u_, x_, p_] := p/u^2 - 1/(x u)
```

representing the surface for $u'' = q$ in an explicit form. For our animation, we select the x-axis as the distinguished coordinate. The three-dimensional surface represents the submanifold $\mathfrak{M}_s = \{u, u' = p, u'' = q\}$ for certain values of x. The different pictures are created by

```
Map[Plot3D[skeleton[u, #, p],
    {u, 0.1, 1}, {p, -3, 3}, PlotRange → {-200, 200},
  Ticks → False, PlotPoints → 35,
   Mesh → False, AxesLabel → {"u", "p", "q"},
   ViewPoint -> {0.717, -2.988, 1.417},
   PlotLabel → StringJoin["x=", ToString[#]]]&,
 Table[i, {i, .01, .1, .01}]]
```



On top of each picture, the specific x-value is given. This allows us to locate the position along the x-axis. The animation shows that the manifold along the x-axis changes rapidly for small x-values. For greater values of x, there are no dramatic changes in $\mathfrak{M}$. We again realize that the skeleton in the original coordinates represents a complicated manifold.

In Section 4.4.1.1, we remarked that the method of canonical variables allows us to simplify the skeleton. To demonstrate this behavior, let us calculate the canonical coordinates related to subgroups *k1* and *k2* for the above ODE. First, we carry out the calculation for subgroup *k1* representing the scaling group for this equation

```
cck1 = CanonicalVariables[{u}, {x}, {x}, {u/2},
  {w}, {t}]; cck1 // LTF
```

$$t == \text{Log}[x]$$

$$w == \frac{u}{\sqrt{x}}$$

The second set of canonical variables follows from the subgroup with k1 = 0 and k2 = 1 by

```
cck2 = CanonicalVariables[{u}, {x}, {x^2}, {x u},
  {w}, {t}]; cck2 // LTF
```

$$t == -\frac{1}{x}$$

$$w == \frac{u}{x}$$

For each set of canonical variables, there exists a representation of the original ODE. The equation in canonical variables for k1 = 1 and k2 = 0 reads

```
ceqk1 = CanonicalRepresentation[firstExample, u, x,
  cck1, w, t]; ceqk1 // LTF
```

$$2\,w - w^3 - 4\,w_t + 4\,w^2\,w_{t,t} == 0$$

The second representation related to the second set of canonical variables is

```
ceqk2 = CanonicalRepresentation[firstExample, u, x,
  cck2, w, t]; ceqk2 // LTF
```

$$t^3\left(\frac{w_t}{w^2} - w_{t,t}\right) == 0$$

Both equations are embedded in the reduced manifold $\mathfrak{M}_c = \{w, w', w''\}$. This manifold is free of the independent variable t and thus simplifies the representation of the equation. The surface of the two manifolds in canonical coordinates is given in the following figure:

```
Show[GraphicsArray[{Plot3D[-(2 w - w^3 - 4 p)/(4 w^2), {w, .1, 2},
    {p, -3, 3}, AxesLabel -> {"w", "P", "Q"}, PlotPoints -> 35,
    Mesh -> False, ViewPoint -> {0.717, -2.988, 1.417},
    Ticks -> False, DisplayFunction -> Identity],
   Plot3D[p/w^2, {w, .1, 2}, {p, -3, 3},
    AxesLabel -> {"w", "P", "Q"}, PlotPoints -> 35,
    Mesh -> False, ViewPoint -> {0.717, -2.988, 1.417},
    Ticks -> False, DisplayFunction -> Identity]},
  DisplayFunction -> $DisplayFunction]]
```

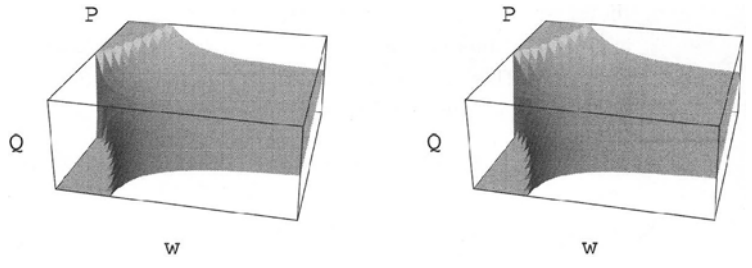


Figure 4.2. The two figures show the skeleton of the equation

$$\partial_{x,x}\, u[x] - \frac{(\partial_x\, u[x])}{u[x]^2} + \frac{1}{x\, u[x]} = 0$$

in canonical representations. Left: skeleton of equation

$$2\, w[t] - w[t]^3 - 4\, w'[t] + 4\, w[t]^2\, w''[t] = 0.$$

Right: skeleton of

$$\left(\frac{w'[t]}{w[t]^2} - w''[t]\right) = 0.$$

The variables P and Q denote the first and second derivatives of canonical variable w, respectively.

We observe that the two figures look very similar. However, the skeletons in their analytical representations are different. Applying canonical coordinates to the original equation, we impressively simplified the skeleton of the equation. The simplification occurs by the elimination of one of the coordinates from the manifold $\mathfrak{M}$.

The question arises of whether there are different solutions existing for similar looking skeletons or whether the solutions are equal. We will examine this question by solving the two canonical representations. First, let us solve the original equation by DSolve[]:

```
DSolve[firstExample == 0, u, x]
```

$$\texttt{DSolve}\left[\frac{1}{x\,u[x]} - \frac{u'[x]}{u[x]^2} + u''[x] == 0,\ u,\ x\right]$$

The result is disappointing. *Mathematica* is unable to solve the second-order equation. Calculating the solution of the first canonical reduction with DSolve[],

```
sol1 = DSolve[ceqk1, w, t]
```

$$\texttt{DSolve}[2\,w[t] - w[t]^3 - 4\,w'[t] + 4\,w[t]^2\,w''[t] == 0,\ w,\ t]$$

shows that again *Mathematica* is unable to solve the equation. The following idea might reveal this problem. We know that a given equation admits a certain type of symmetry. This is also true for the first canonical reduction. The symmetries of the first canonical reduction follow by

```
infceqk1 = Infinitesimals[ceqk1, w, t]; infceqk1 // LTF
```

$$\phi_1 == E^t\ k1\ w$$

$$\xi_1 == 2\ E^t\ k1 + k2$$

representing a two-dimensional symmetry group with an exponential dependence in t. Since the subgroup with $k2 \neq 0$ only represents a translation in t, we restrict our examinations to the case $k1 = 1/2$ and $k2 = 0$. The canonical variables for this subgroup follow by

```
ccceqk11 = CanonicalVariables[{w}, {t}, {E^t}, {E^t w / 2},
  {v}, {s}]; ccceqk11 // LTF
```

$$s == -E^{-t}$$

$$v == E^{-t/2}\ w$$

Inserting these new coordinates into the first canonical reduction, we get a second canonical representation of the first reduction:

```
ceqk11 = CanonicalRepresentation[ceqk1, w, t,
  ccceqk11, v, s]; ceqk11 // Flatten // LTF
```

$$-4\ I\ \sqrt{s}\ \ (v_s - v^2\ v_{s,s}) == 0$$

The calculation shows that the two canonical representations of the first and second symmetry levels are identical. Compare the second canonical reduction of the first symmetry level with the present result:

```
eqh = Thread[ceqk11[[1]] / (-4 I Sqrt[s]), Equal] == 0; eqh // LTF
```

$$v_s - v^2\ v_{s,s} == 0$$

The symmetry analysis of this equation illustrates that the equation admits a second-order group. We know from the above discussions that a second-order group is sufficient to solve this kind of ODE. The infinitesimals of this ODE follow by

```
infh = Infinitesimals[eqh, v, s]; infh // LTF
```

$$\phi_1 == \frac{k2\ v}{2}$$

$$\xi_1 == k1 + k2\ s$$

To solve the equation *eqh*, we apply the method of first integrals in an adapted form. First, let us determine the Lie matrix with the infinitesimals

```
inf1 =
 {{xi[1][s, v]}, {phi[1][s, v]}} /. infh /. {k1 → 1, k2 → 0} /.
  v → v[s];

inf2 =
 {{xi[1][s, v]}, {phi[1][s, v]}} /. infh /. {k1 → 0, k2 → 1} /.
  v → v[s];
```

Inserting the infinitesimals into the Lie matrix, we get

```
Δmatrix = DeltaMatrix[s, v, v'[s]/v[s]^2, 2, {inf1, inf2}];

TableForm[Δmatrix] // LieTraditionalForm
```

$$\begin{array}{lll} 1 & 0 & 0 \\ s & \frac{v}{2} & -\frac{v_s}{2} \\ 1 & v_s & \frac{v_s}{v^2} \end{array}$$

whose determinant is a non-vanishing quantity

```
Det[Δmatrix] // LieTraditionalForm
```

$$\frac{v_s}{2\,v} + \frac{v_s^2}{2}$$

One of the two first integrals follows from

```
integral2 = FirstIntegral[s, v, Δmatrix, 2] == c2
```

```
2 Log[v[s]] - 2 Log[1 + v[s] v'[s]] == c2
```

The second first integral is not accessible by an integration:

```
integral1 = FirstIntegral[s, v, Δmatrix, 1] == c1
```

$$\int \left(v[s]'[t]\left(-\frac{1}{2}v'[s][t] - \frac{s[t]\,v'[s][t]}{v[s][t]^2}\right) + s'[t]\left(\frac{v'[s][t]}{2\,v[s][t]} + \frac{1}{2}v'[s][t]^2\right) + \left(-\frac{1}{2}v[s][t] + s[t]\,v'[s][t]\right)v''[s][t]\right) \Big/ \left(\frac{v'[s][t]}{2\,v[s][t]} + \frac{1}{2}v'[s][t]^2\right) dt == c1$$

However, the solution of the equation *eqh* is derived if we take the first integral as a defining equation for v. Thus, another integration by DSolve[] gives us the solution

```
csolex2 = DSolve[integral2, v, s]
```

```
InverseFunction::ifun : Warning: Inverse functions are
   being used. Values may be lost for multivalued inverses.
```

$$\left\{\left\{v \to \left(E^{c2/2}\left(1 + \text{ProductLog}\left[E^{-1+E^{-c2}\,(\#1 - C[1])}\right]\right)\&\right)\right\}\right\}$$

This type of solution also follows by applying the function SecondOrderIntegrate[] to the equation in canonical variables:

```
SecondOrderIntegrate[eqh, v, s]
```

```
InverseFunction::ifun : Warning: Inverse functions are
   being used. Values may be lost for multivalued inverses.
```

$$\left\{v \to \text{Function}\left[s,\ E^{\frac{CI[2]}{2}}\left(1 + \text{ProductLog}\left[E^{-1+E^{-CI[2]}\,(s - C[1])}\right]\right)\right]\right\}$$

The solution of the original equation in variables x and u thus follows by inverting the canonical transformations

```
csol = ((v[s] /. csolex2)[[1]] == v /.
       {v → v[t, w], s → s[t, w]} /. ccceqk11) /.
   {w → w[x, u], t → t[x, u]} /.
  cck1
```

$$E^{c2/2}\left(1 + \text{ProductLog}\left[E^{-1+E^{-c2}\,\left(-\frac{1}{x} - C[1]\right)}\right]\right) == \frac{u}{x}$$

The explicit solution for the original equation thus reads

```
sol = Solve[csol, u] // Simplify
```

$$\left\{\left\{u \to E^{c2/2}\,x\left(1 + \text{ProductLog}\left[E^{-1-\frac{E^{-c2}\,(1+x\,C[1])}{x}}\right]\right)\right\}\right\}$$

where the constants c_2 and $C[1]$ are constants of integration. This example shows that a solution of an ODE is derived if a hybrid algorithm, combining the method of first integrals, the method of canonical variables, and the solution procedure of *Mathematica,* is applied. The solution calculated above is not accessible by DSolve[] or one of the two other methods alone. We can check the solution by inserting the result into the original equation:

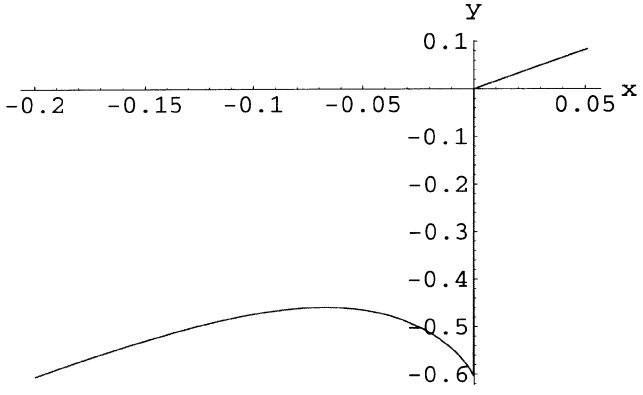
```
firstExample /. (u → Function[x, w] /. (sol[[1, 1]] /. u → w)) //
 Simplify
```

```
0
```

The resulting zero demonstrates that the derived solution satisfies the original equation. This solution is new in the sense that the explicit representation for the

original equation is given. The solution in an implicit representation was given by Ibragimov [1994]. A graphical representation of our result, for fixed constants c_2 and $C[1]$, is shown by

```
Plot[u /. sol /. {c2 → 1, C[1] → 1 / 2},
 {x, -.20, .0510}, PlotStyle → RGBColor[0.996109, 0, 0],
 AxesLabel → {"x", "y"}, PlotRange → All]
```



The example presented in this section illustrates how the combination of different strategies allows the derivation of a solution. In the following section, we will generalize the presented procedures to higher-order ODEs.

4.4.3 Higher-Order Ordinary Differential Equations

Differential equations of higher order arise naturally in physics. For example, third-order ODEs come up in fluid dynamics and fourth-order ODEs in elasticity. For general higher-order equations, there exist hardly any techniques for obtaining explicit symbolic solutions. This means that higher-order ODEs are thus not well studied in the literature. In the following, we will present a symbolic technique for producing explicit solutions independent of the order of the equation. The method described in the preceding sections can, without essential changes, be generalized to the solution of differential equations of higher order.

4.4.3.1 Integrating Factor Method

The essential change in the theory of an integrating factor for higher-order ODEs is the extension of the Lie matrix to higher prolongations. Extending the Lie matrix to higher prolongations is the key step for generalizing the procedure of integrating

factors. Lie called this extension the determination of the multiplier of the differential equation.

The five steps of integration discussed for second-order equations remain the same for higher-order equations. However, the dimension of the Lie matrix changes from a 3×3 matrix to an $(n+1)\times(n+1)$ matrix. Before we discuss the algorithm, let us state the general settings for higher-order equations.

The most general form of an nth-order ODE is given by

$$F(x, u, u', u'', \ldots, u_{(n)}) = 0, \tag{4.65}$$

where $u_{(n)} = \frac{d^n u}{dx^n}$ denotes the nth derivative of u with respect to x. In the following, we assume that equation (4.65) can be solved with respect to the nth derivative. Thus, the actual equation under consideration is

$$u_{(n)} = \omega(x, u, u', \ldots, u_{(n-1)}), \tag{4.66}$$

where ω is a given function of $x, u, u', \ldots, u_{(n-1)}$.

If an nth-order equation admits a finite symmetry of dimension $r \geq n$, then the equation can be integrated by group-theoretic quadrature methods. This, also, can be done in various ways. For a discussion of other procedures, compare Sections 4.4.1 and 4.4.2. One of the group-theoretic algorithms is based on first integrals. The algorithm for an nth-order ODE is summarized as follows:

1. Compute the Lie algebra L_r. A basis for L_r is the set $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r$. The tangent vector fields $\vec{v}_i$ follow from appropriately specifying the group constants.
2. If $r = n$, go to the next step;
   if $r > n$, distinguish any n-dimensional subalgebra L_n of L_r.
   If $r = n - 1$, the order of the equation may be lowered;
   if $r = 0$, the group method is not useful.
3. Calculate the Lie determinants $d\Delta_i$ and Δ, and if possible, determine the n first integrals by integration. The Lie determinants are defined by

$$d\Delta_1 = \det\begin{pmatrix} dx & du & du' & \ldots & du^{(n-1)} \\ \xi_2 & \phi_2 & \phi_2' & \ldots & \phi_2^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u' & u'' & \ldots & \omega \end{pmatrix}, \tag{4.67}$$

and

$$d\Delta_i = \det\begin{pmatrix} \xi_1 & \phi_1 & \phi_1' & \dots & \phi_1^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ dx & du & du' & \dots & du^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u' & u'' & \cdots & \omega \end{pmatrix} \tag{4.68}$$

where i denotes the row of the Lie matrix in which infinitesimals are replaced by differentials. The $n \times n$ Lie determinant Δ reads

$$\Delta = \det\begin{pmatrix} \xi_1 & \phi_1 & \phi_1' & \dots & \phi_1^{(n-1)} \\ \xi_2 & \phi_2 & \phi_2' & \dots & \phi_2^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u' & u'' & \dots & \omega \end{pmatrix}, \tag{4.69}$$

where ξ_i and ϕ_i are the infinitesimals of the vector field $\vec{v}_i$ and ϕ_i' denotes the first and $\phi_i^{(n-1)}$ the $(n-1)$th extension of the infinitesimals ϕ_i. The corresponding first integrals ψ_i are given by

$$\psi_i = \int \frac{d\Delta_i}{\Delta} = c_i, \qquad i = 1, 2, \ldots, n. \tag{4.70}$$

The constants c_i denote the integration constants of the ODE.

4. Solve one of the n integrals with respect to the $(n-1)$th-order derivative and substitute the result into the remaining relations. Repeat this procedure until no derivative remains in the relations.

5. If we can solve the resulting relation with respect to the unknown function u, we have found an explicit solution. Otherwise, our solution is implicit.

This procedure becomes very cumbersome with increasing orders of the equation if done by hand. In principle, the procedure can be applied to any kind of linear or non-linear ODE. How the algorithm works in particular examples is demonstrated below.

Example 1

The first example is a third-order equation listed by Kamke [1977] as No. 7.13:

```
thirdOrderExample = ∂x,x u[x] ∂{x,3} u[x] - α √(1 + β² (∂x,x u[x])²) ;
Map[# == 0&, {thirdOrderExample}] // LieTraditionalForm //
 TableForm
```

$$-\alpha \sqrt{1 + \beta^2\, u_{x,x}^2} + u_{x,x}\, u_{x,x,x} == 0$$

The parameters α and β are real constants. According to Kamke, this third-order ODE is solvable and the solution can only be represented in parametric form. We will show here that an explicit solution of the equation is possible. First, let us check if *Mathematica* can solve the third-order equation.

```
DSolve[thirdOrderExample == 0, u, x]
```

```
DSolve[-α Sqrt[1 + β^2 u''[x]^2] + u''[x] u^(3)[x] == 0, u, x]
```

The above line shows that *Mathematica* is not capable of solving the equation. The question now is: Can we derive the necessary number of symmetries in order to integrate the equation? Deriving the symmetries is the first step in the general algorithm. The calculation of symmetries is carried out by the *MathLie* function Infinitesimals[]

```
infi = Infinitesimals[thirdOrderExample, u, x, {α, β},
  SubstitutionRules → {∂{x,3} u[x]}]
```

```
{phi[1] → Function[{x, u}, k2 + k3 x],
 xi[1] → Function[{x, u}, k1]}
```

The result is a symmetry group of order three. The number of group constants is equal to the order of the equation. This allows us to apply the integrating algorithm discussed above. The specific symmetries are denoted by the group constants *k1*, *k2*, and *k3*. Each of these parameters is related to a vector field $\vec{v}_i$, $i = 1, 2, 3$. Since the number of vector fields is equal to the order of the equation, we can go to step 3 of the algorithm. In the third step, we determine the Lie matrix by inserting the prolongations of the infinitesimals and the equation itself:

```
inf1 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
    {k1 → 1, k2 → 0, k3 → 0} /.
  u → u[x];

inf2 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
    {k1 → 0, k2 → 1, k3 → 0} /.
  u → u[x];

inf3 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
    {k1 → 0, k2 → 0, k3 → 1} /.
  u → u[x];
```

The Lie matrix is derived by

```
Δmatrix = DeltaMatrix[x,
  u, α Sqrt[1 + β^2 (∂x,x u[x])^2] / ∂x,x u[x], 3, {inf1, inf2, inf3}];
TableForm[Δmatrix] // LieTraditionalForm
```

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & x & 1 & 0 \\ 1 & u_x & u_{x,x} & \dfrac{\alpha\sqrt{1+\beta^2\,u_{x,x}^2}}{u_{x,x}} \end{matrix}$$

One of the three first integrals is

```
integ1 = FirstIntegral[x, u, Δmatrix, 1] == c1 // Simplify;
integ1 // LieTraditionalForm
```

$$x - \frac{\sqrt{1+\beta^2\,u_{x,x}^2}}{\alpha\,\beta^2} == c1$$

The result depends on u'' and now allows us to rewrite all terms containing u'' in the Lie matrices. Next, we solve the first integral with respect to u''. Since the integral depends quadratically on u'', we get two solutions:

```
sol1 = Solve[integ1, ∂x,x u[x]]
```

$$\left\{\left\{u''[x] \to -\sqrt{-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2}\right\},\right.$$
$$\left.\left\{u''[x] \to \sqrt{-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2}\right\}\right\}$$

The first of the two solutions is used to replace u'' in the Lie matrix. The reader can easily do the calculation for the second solution by himself:

```
dmat = Δmatrix /. sol1[[1]]
```

$$\left\{\{1,\ 0,\ 0,\ 0\},\ \{0,\ 1,\ 0,\ 0\},\ \{0,\ x,\ 1,\ 0\},\right.$$
$$\left\{1,\ u'[x],\ -\sqrt{-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2},\right.$$
$$\left.\left.-\left(\alpha\sqrt{\left(1+\beta^2\left(-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)\right)}\right) \Big/ \right.\right.$$
$$\left.\left.\left(\sqrt{-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2}\right)\right\}\right\}$$

The simplified Lie matrix is used again to calculate the second first integral of the third-order equation:

```
integ2 = FirstIntegral[x, u, dmat, 3] == c2 // Simplify
```

$$\left(\sqrt{-\frac{1}{\beta^2}+c1^2\,\alpha^2\,\beta^2-2\,c1\,x\,\alpha^2\,\beta^2+x^2\,\alpha^2\,\beta^2}\right.$$
$$(-1+c1^2\,\alpha^2\,\beta^4-2\,c1\,x\,\alpha^2\,\beta^4+x^2\,\alpha^2\,\beta^4)+$$
$$\left.\alpha\,\beta^2\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\;u'[x]\right)\Big/$$
$$\left(\alpha\,\beta^2\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right)==$$
$$c2$$

As expected, we find the integral depending only on first derivatives of u. Since this integral is linear in u', it is uniquely solvable in u'

```
sol2 = Solve[integ2, u'[x]] // Simplify
```

$$\left\{\left\{u'[x]\to c2-\left(\sqrt{-\frac{1}{\beta^2}+c1^2\,\alpha^2\,\beta^2-2\,c1\,x\,\alpha^2\,\beta^2+x^2\,\alpha^2\,\beta^2}\right.\right.$$
$$(-1+c1^2\,\alpha^2\,\beta^4-2\,c1\,x\,\alpha^2\,\beta^4+x^2\,\alpha^2\,\beta^4)\Big)\Big/$$
$$\left.\left.\left(\alpha\,\beta^2\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right)\right\}\right\}$$

The resultant expression contains radicals of quadratic polynomials in x. Inserting this result into the Lie matrix, we are able to eliminate the dependencies on u'. We find

```
dmat1 = dmat /. sol2[[1, 1]] // Simplify
```

$$\left\{\{1,\ 0,\ 0,\ 0\},\ \{0,\ 1,\ 0,\ 0\},\ \{0,\ x,\ 1,\ 0\},\right.$$
$$\left\{1,\ c2-\left(\sqrt{-\frac{1}{\beta^2}+c1^2\,\alpha^2\,\beta^2-2\,c1\,x\,\alpha^2\,\beta^2+x^2\,\alpha^2\,\beta^2}\right.\right.$$
$$(-1+c1^2\,\alpha^2\,\beta^4-2\,c1\,x\,\alpha^2\,\beta^4+x^2\,\alpha^2\,\beta^4)\Big)\Big/$$
$$\left.\left(\alpha\,\beta^2\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right),\right.$$
$$-\sqrt{-\frac{1}{\beta^2}+c1^2\,\alpha^2\,\beta^2-2\,c1\,x\,\alpha^2\,\beta^2+x^2\,\alpha^2\,\beta^2},$$
$$\left.-\frac{\alpha\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}}{\sqrt{-\frac{1}{\beta^2}+c1^2\,\alpha^2\,\beta^2-2\,c1\,x\,\alpha^2\,\beta^2+x^2\,\alpha^2\,\beta^2}}\right\}\right\}$$

The last step of integration is inserting the Lie matrices into the third first integral depending on u and x:

```
integ3 = FirstIntegral[x, u, dmat1, 2] == c3 // Simplify
```

$$\left(-c2\,x^3\,\alpha^4\,\beta^8 + c2\,\alpha\,\beta^2\sqrt{(c1-x)^2\,\alpha^2\,\beta^4} - c2\,x^2\,\alpha^3\,\beta^6\sqrt{(c1-x)^2\,\alpha^2\,\beta^4} + \right.$$

$$\sqrt{-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2} -$$

$$2\,x^2\,\alpha^2\,\beta^4\sqrt{\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2} +$$

$$c1^4\,\alpha^4\,\beta^8\sqrt{\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2} -$$

$$4\,c1^3\,x\,\alpha^4\,\beta^8\sqrt{\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2} +$$

$$x^4\,\alpha^4\,\beta^8\sqrt{\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2} +$$

$$2\,c1\,x\,\alpha^2\,\beta^4\left(c2\,x\,\alpha^2\,\beta^4 + c2\,\alpha\,\beta^2\sqrt{(c1-x)^2\,\alpha^2\,\beta^4} + \right.$$

$$2\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)} -$$

$$\left.2\,x^2\,\alpha^2\,\beta^4\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)}\right) +$$

$$c1^2\,\alpha^2\,\beta^4\left(-c2\,x\,\alpha^2\,\beta^4 - c2\,\alpha\,\beta^2\sqrt{(c1-x)^2\,\alpha^2\,\beta^4} - \right.$$

$$2\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)} +$$

$$\left.6\,x^2\,\alpha^2\,\beta^4\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)}\right) +$$

$$\left.(c1-x)^2\,\alpha^4\,\beta^8\,u[x]\right) \Big/$$

$$((c1-x)^2\,\alpha^4\,\beta^8) ==$$

$$c3$$

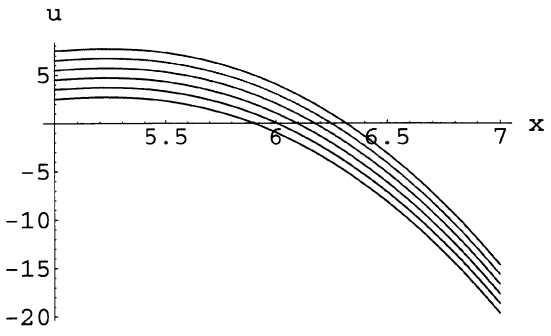
The integral contains the dependent variable u again as a linear variable. In turn, we end up with a unique solution for the Kamke equation 7.13, which is

```
solution = Solve[integ3, u[x]] // Simplify
```

$$\left\{\left\{u[x] \to c3 - \left(c1^4\,\alpha^4\,\beta^8\,\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)} - 4\,c1^3\,x\,\alpha^4\,\beta^8\,\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)} + \sqrt{\left(-\frac{1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)}\,(-1 + x^2\,\alpha^2\,\beta^4)^2 - c2\,\alpha\,\beta^2\left(x^3\,\alpha^3\,\beta^6 - \sqrt{(c1-x)^2\,\alpha^2\,\beta^4} + x^2\,\alpha^2\,\beta^4\,\sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right) + c1^2\,\alpha^2\,\beta^4\left(2\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)}\,(-1 + 3\,x^2\,\alpha^2\,\beta^4) - c2\,\alpha\,\beta^2\left(x\,\alpha\,\beta^2 + \sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right)\right) + 2\,c1\,x\,\alpha^2\,\beta^4\left(-2\sqrt{\left(\frac{-1}{\beta^2} + c1^2\,\alpha^2\,\beta^2 - 2\,c1\,x\,\alpha^2\,\beta^2 + x^2\,\alpha^2\,\beta^2\right)}\,(-1 + x^2\,\alpha^2\,\beta^4) + c2\,\alpha\,\beta^2\left(x\,\alpha\,\beta^2 + \sqrt{(c1-x)^2\,\alpha^2\,\beta^4}\right)\right)\right) \Big/ \left((c1-x)^2\,\alpha^4\,\beta^8\right)\right\}\right\}$$

The solution depends on three constants *c1*, *c2*, and *c3*, all of which are constants of integration. The parameters α and β are the parameters of the original equation. To get a feeling how the solution evolves, we plot the solution for different parameter sets *c1*, *c2*, *c3* at fixed α and β:

```
s1 = Table[
   (u[x] /. solution /. {c1 → 1, c2 → 1/2, c3 → i, α → 1, β → 1/2})[[1]],
   {i, 0, 5}];

Plot[Evaluate[s1],
 {x, 5, 7}, PlotStyle → {RGBColor[0, 0, 0.996109],
   RGBColor[0, 0, 0.62501], RGBColor[0.500008, 0, 0.500008],
   RGBColor[0.500008, 0, 0.996109],
   RGBColor[0.996109, 0, 0.500008],
   RGBColor[0.500008, 0, 0.250004]},
 AxesLabel → {"x", "u"}]
```



The figure represents the real valued solutions of the equation

$$\partial_{x,x}\, u[x]\, \partial_{\{x,3\}}\, u[x] - \alpha \sqrt{1 + \beta^2 (\partial_{x,x}\, u[x])^2} = 0.$$

The different curves represent the solutions for values of c3 $\in \{0, 1, 2, 3, 4, 5\}$ and fixed values for c2 = 1/2, c1 = 1. The parameters of the equation are $\alpha = 1$ and $\beta = 1/2$.

We note that the solution is explicitly represented by a complicated expression containing radicals and polynomials. This result is new, as Kamke only offers a parametric representation of the solution. The example shows that with Lie's procedure, we are able to arrive at a solution for higher-order ODEs. *Mathematica*, by itself, is not yet able to handle this type of equation. □

Example 2

The second example for higher-order equations is a third-order equation. In this case, the equation has no direct physical origin. It is only used for checking the integration procedure:

$$u_{x,x,x} + \mathfrak{Re}\,(u_x\, u_{x,x} - u(x)\, u_{x,x,x}) = 0, \tag{4.71}$$

where $\mathfrak{Re}$ is a positive constant. The equation has some resemblance to hydrodynamic equations if the first term of the equation is replaced by a fourth-order derivative $u_{x,x,x,x}$. However, the fourth-order equation does not possess the necessary number of symmetries to start the integration process. The reader may check this. The equation under consideration is thus

```
thirdOrder = ∂x,x,x u[x] + Re (∂x u[x] ∂x,x u[x] - u[x] ∂x,x,x u[x]);
Map[# == 0&, {thirdOrder}] // LieTraditionalForm // TableForm
```

```
ux,x,x + Re (ux ux,x - u ux,x,x) == 0
```

where $\mathfrak{Re}$ is a real constant. The equation has a minimum of required symmetries given by

```
infi = Infinitesimals[thirdOrder , u, x, {Re},
  SubstitutionRules → {∂{x,3} u[x]}]
```

$$\left\{\text{xi}[1] \to \text{Function}[\{x, u\},\ k1 + k2\, x],\ \text{phi}[1] \to \text{Function}\left[\{x, u\},\ \frac{k3\ (-1 + u\,\mathfrak{Re})}{\mathfrak{Re}}\right]\right\}$$

The calculation below shows that the functions of *MathLie* are not able to find all first integrals in a single run. We need to split the integration into a few steps considering the symmetries given by the vector fields $\vec{v}_i$. The three symmetries related to the vector fields are given by

```
inf1 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
   {k1 → 1, k2 → 0, k3 → 0} /.
  u → u[x];

inf2 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
   {k1 → 0, k2 → 1, k3 → 0} /.
  u → u[x];

inf3 = {{xi[1][x, u]}, {phi[1][x, u]}} /. infi /.
   {k1 → 0, k2 → 0, k3 → 1} /.
  u → u[x];
```

The right-hand side of the third-order equation ω is given by

```
ω = (Solve[thirdOrder == 0, ∂x,x,x u[x]])[[1, 1, 2]];
ω // LieTraditionalForm
```

$$\frac{\mathfrak{Re}\ u_x\ u_{x,x}}{-1 + u\,\mathfrak{Re}}$$

The 4×4 Lie matrix follows by inserting the infinitesimals of the vector fields and ω into the function DeltaMatrix[]:

```
Δmatrix = DeltaMatrix[x, u, ω, 3, {inf1, inf2, inf3}];
TableForm[Δmatrix] // LieTraditionalForm
```

1	0	0	0
x	0	$-u_x$	$-2\,u_{x,x}$
0	$\frac{-1 + u\,\mathfrak{Re}}{\mathfrak{R}}\,e$	u_x	$u_{x,x}$
1	u_x	$u_{x,x}$	$\frac{\mathfrak{Re}\ u_x\ u_{x,x}}{-1 + u\,\mathfrak{Re}}$

The determinant of the Lie matrix is a polynomial in u, u', and u'':

```
Det[Δmatrix] // LieTraditionalForm
```

$$2\,u_x^2\,u_{x,x} - 2\,u\,u_{x,x}^2 + \frac{2\,u_{x,x}^2}{\mathrm{Re}}$$

In order to obtain first integrals for the equation, the second line of the above Δ matrix is replaced by the differentials:

```
integ1 = FirstIntegral[x, u, Δmatrix, 2] == c1 // Simplify
```

$$\frac{1}{2}\,(\mathrm{Log}[-1 + \mathrm{Re}\,u[x]] - \mathrm{Log}[u''[x]]) == c1$$

The remaining two integrals ψ_1 and ψ_3 are not accessible to FirstIntegral[]. The reason for this is that the function Integrate[] of *Mathematica* cannot solve certain types of integrals. However, the result so far derived is helpful to find a solution of the equation. If we look at the first integral ψ_2, we observe that this relation is a second-order ODE. The solution for u'' clearly shows

```
sol1 = Solve[integ1, ∂x,x u[x]] // Simplify
```

$$\{\{u''[x] \to E^{-2\,c1}\,(-1 + \mathrm{Re}\,u[x])\}\}$$

If we transform *Rule* to *Equal*, we get the equation

```
eqh = (sol1 /. Rule → Equal)[[1, 1]]
```

$$u''[x] == E^{-2\,c1}\,(-1 + \mathrm{Re}\,u[x])$$

a second-order equation which is now solvable by DSolve[]:

```
sol2 = DSolve[eqh, u, x] /. c1 → C[3]
```

$$\left\{\left\{u \to \left(\frac{1}{\mathrm{Re}} + E^{-E^{-C[3]}\sqrt{\mathrm{Re}}\,\#1}\,C[1] + E^{E^{-C[3]}\sqrt{\mathrm{Re}}\,\#1}\,C[2]\,\&\right)\right\}\right\}$$

Inserting this solution into the original equation, we can verify that the original equation is satisfied,

```
thirdOrder /. sol2 // Simplify
```

$$\{0\}$$

meaning that the left-hand side of the equation vanishes, and, in turn, equality has been established. Solving the original equation with *Mathematica,* we get

```
DSolve[thirdOrder == 0, u, x]
```

$$\mathrm{DSolve}[u^{(3)}[x] + \mathrm{Re}\,(u'[x]\,u''[x] - u[x]\,u^{(3)}[x]) == 0,\ u,\ x]$$

showing us that *Mathematica* in the present form is not capable of handling third-order equations.

The example demonstrates that we can even find solutions for cases where we know fewer first integrals than the order of the equation. The procedure of integration splits heterogeneously using different tools to solve the reduced equation. This behavior of higher-order ODEs creates some difficulties in the automatic solution process. □

Example 3

The third example considers the fourth-order ODE No. 7.16 of Kamke [1977]. This equation is a non-linear ODE containing second-, third-, and fourth-order derivatives. The problem with such an equation is that no standard procedure in literature offers a way to construct the solution given by Kamke. We will demonstrate that the integrating factor method is very effective for the construction of the solution. The equation No. 7.16 by Kamke reads

```
kamke716 = 3 ∂x,x u[x] ∂{x,4} u[x] - 5 (∂x,x,x u[x])^2 == 0;
kamke716 // LieTraditionalForm
```

$$-5\,u_{x,x,x}^2 + 3\,u_{x,x}\,u_{x,x,x,x} == 0$$

Kamke also lists the solution of the above ODE in implicit form:

$$(u(x) + C_1\,x + C_2)^2 = C_3\,x + C_4, \tag{4.72}$$

where C_1, C_2, C_3, and C_4 are real constants. The following examinations will demonstrate that this simple solution follows from our procedure. The first step of Lie's procedure is the determination of the infinitesimals:

```
infkamke = Infinitesimals[kamke716, u, x,
  SubstitutionRules → {∂{x,4} u[x]}]
```

```
{phi[1] → Function[{x, u}, k3 + k1 u + k4 x],
 xi[1] → Function[{x, u}, k5 + k2 u + k6 x]}
```

The result of this calculation is a symmetry group containing six group parameters k_i. The second step consists in finding a solvable subalgebra of dimension four from these infinitesimals. The determination of all solvable subalgebras of dimension four is carried out with

```
solvable = SolvableAlgebrasOfOrderN[infkamke, {u}, {x}, 4]
```

```
{{{k1 → 1}, {k2 → 1}, {k3 → 1}, {k5 → 1}},
 {{k1 → 1}, {k2 → 1}, {k4 → 1}, {k6 → 1}},
 {{k1 → 1}, {k2 → 1}, {k5 → 1}, {k6 → 1}},
```

```
{{k1 → 1}, {k3 → 1}, {k4 → 1}, {k5 → 1}},
{{k1 → 1}, {k3 → 1}, {k4 → 1}, {k6 → 1}},
{{k1 → 1}, {k3 → 1}, {k5 → 1}, {k6 → 1}},
{{k2 → 1}, {k3 → 1}, {k5 → 1}, {k6 → 1}},
{{k3 → 1}, {k4 → 1}, {k5 → 1}, {k6 → 1}}}
```

From the result, we can choose one of the eight solvable algebras. The related coefficients of the vector fields creating these subalgebras are derived by inserting the above result and assuming the other group constants equal to zero:

```
vectorBasis =
 Map[({{xi[1][x, u]}, {phi[1][x, u]}} /. infkamke /. # /.
      {k1 → 0, k2 → 0, k3 → 0, k4 → 0, k5 → 0, k6 → 0, u → u[x]})&,
  solvable]
```

```
{{{{0}, {u[x]}}, {{u[x]}, {0}}, {{0}, {1}}, {{1}, {0}}},
 {{{0}, {u[x]}}, {{u[x]}, {0}}, {{0}, {x}}, {{x}, {0}}},
 {{{0}, {u[x]}}, {{u[x]}, {0}}, {{1}, {0}}, {{x}, {0}}},
 {{{0}, {u[x]}}, {{0}, {1}}, {{0}, {x}}, {{1}, {0}}},
 {{{0}, {u[x]}}, {{0}, {1}}, {{0}, {x}}, {{x}, {0}}},
 {{{0}, {u[x]}}, {{0}, {1}}, {{1}, {0}}, {{x}, {0}}},
 {{{u[x]}, {0}}, {{0}, {1}}, {{1}, {0}}, {{x}, {0}}},
 {{{0}, {1}}, {{0}, {x}}, {{1}, {0}}, {{x}, {0}}}}
```

Knowing the infinitesimals of the solvable subalgebras, we can proceed to the integration step of the algorithm. The information from the original equation *kamke176* about the right-hand side of the equation is extracted by

```
ω = Solve[kamke716, ∂{x,4} u[x]][[1, 1, 2]];
ω // LieTraditionalForm
```

$$\frac{5\,u_{x,x,x}^2}{3\,u_{x,x}}$$

The Lie matrix of the equation is then calculated for the fourth subalgebra by

```
Δmatrix = DeltaMatrix[x, u, ω, 4, vectorBasis[[4]]];
TableForm[Δmatrix] // LieTraditionalForm
```

0	u	u_x	$u_{x,x}$	$u_{x,x,x}$
0	1	0	0	0
0	x	1	0	0
1	0	0	0	0
1	u_x	$u_{x,x}$	$u_{x,x,x}$	$\frac{5\,u_{x,x,x}^2}{3\,u_{x,x}}$

The determinant of the Δ matrix is

```
Det[Δmatrix] // LieTraditionalForm
```

$$-\frac{2}{3}\,u_{x,x,x}^2$$

Knowing that the determinant of Lie's matrix is a non-vanishing quantity, we can calculate the first integrals of the equation. One of these integrals is

```
integ1 = FirstIntegral[x, u, Δmatrix, 1] == c1 // Simplify
```

$$\frac{1}{2}\left(5\,\text{Log}[u''[x]] - 3\,\text{Log}[u^{(3)}[x]]\right) == c1$$

A second first integral follows by

```
integ2 = FirstIntegral[x, u, Δmatrix, 4] == c2 // Simplify
```

$$x + \frac{3\,u''[x]}{2\,u^{(3)}[x]} == c2$$

At this point of our calculation, we know that equation No. 7.16 by Kamke allows two conserved quantities given by *integ1* and *integ2*. The right-hand sides of these differential expressions *c1* and *c2* are two real constants. The two integrals contain derivatives of third and second order. Since we know that both expressions are conserved, we can use one of these quantities to eliminate higher derivatives. We decide to eliminate the third-order derivative in the first integral *integ1* by

```
s1 = Solve[integ2, ∂x,x,x u[x]] // Simplify
```

$$\left\{\left\{u^{(3)}[x] \to \frac{3\,u''[x]}{2\,c2 - 2\,x}\right\}\right\}$$

The elimination of the third-order derivative in *integ1* gives us

```
integ1Help = integ1 /. s1[[1]]
```

$$\frac{1}{2}\left(5\,\text{Log}[u''[x]] - 3\,\text{Log}\left[\frac{3\,u''[x]}{2\,c2 - 2\,x}\right]\right) == c1$$

representing an integral containing only second-order derivatives. The solution of this expression with respect to the second integral delivers

```
s2 = Solve[integ1Help, ∂x,x u[x]]
```

$$\left\{\left\{u''[x] \to -\frac{3\sqrt{3}\;E^{c1}}{\sqrt{8\,c2^3 - 24\,c2^2\,x + 24\,c2\,x^2 - 8\,x^3}}\right\},\right.$$
$$\left.\left\{u''[x] \to \frac{3\sqrt{3}\;E^{c1}}{\sqrt{8\,c2^3 - 24\,c2^2\,x + 24\,c2\,x^2 - 8\,x^3}}\right\}\right\}$$

The two resulting expressions can be integrated twice to find the solution. However, we apply DSolve[] to the expressions to find the solution. Before we can use DSolve[], we need to transform the rules to equations by

```
eqh = s2 /. Rule → Equal // Flatten
```

$$\left\{u''[x] == -\frac{3\sqrt{3}\ E^{c1}}{\sqrt{8\,c2^3 - 24\,c2^2\,x + 24\,c2\,x^2 - 8\,x^3}}\right.$$

$$\left.u''[x] == \frac{3\sqrt{3}\ E^{c1}}{\sqrt{8\,c2^3 - 24\,c2^2\,x + 24\,c2\,x^2 - 8\,x^3}}\right\}$$

Solving the first equation gives us

```
sol1 = DSolve[eqh[[1]], u, x]
```

$$\left\{\left\{u \to \left(C[1] + C[2]\,\#1 - \frac{3\sqrt{3}\ E^{c1}\sqrt{8\,c2^3 - 24\,c2^2\,\#1 + 24\,c2\,\#1^2 - 8\,\#1^3}}{2\,(-c2 + \#1)}\&\right)\right\}\right\}$$

The second relation for u'' delivers the second solution

```
sol2 = DSolve[eqh[[2]], u, x]
```

$$\left\{\left\{u \to \left(C[1] + C[2]\,\#1 + \frac{3\sqrt{3}\ E^{c1}\sqrt{8\,c2^3 - 24\,c2^2\,\#1 + 24\,c2\,\#1^2 - 8\,\#1^3}}{2\,(-c2 + \#1)}\&\right)\right\}\right\}$$

In conclusion, we find two solutions in an explicit representation. This has to be expected since the solution given by Kamke contains the unknown variable u in quadratic form. The derived solution can be inserted into the original equation to verify that the gained results are correct. For the first solution, we find

```
kamke716 /. sol1 // Simplify
```

```
{True}
```

meaning that the first solution satisfies the equation. The second solution also satisfies the equation

```
kamke716 /. sol2 // Simplify
```

```
{True}
```

At the end, we demonstrated that the integrating factor method is capable of solving a fourth-order equation. □

5

Point Symmetries of Partial Differential Equations

5.1. Introduction

The subject of this section is to discuss the basic tools of Lie's symmetry method in connection with partial differential equations. These tools will support the practical calculations. We will show how the theory of Lie becomes vital again by using computer algebra calculations.

The theory under discussion is the symmetry theory of Lie. This theory is useful for solving partial differential equations in a systematic way. The question Lie had raised more than 100 years ago was how to systematically solve differential equations. He was wondering about the many methods his colleagues used in solving differential equations. Up to the present day, this question of deriving solutions for a given differential equation has been of topical interest for physicists and mathematicians alike. Lie found a solution to this problem by introducing a method which allows the examination of symmetry transformations of equations. Using this method, he was able to find solutions not only for ordinary differential equations as discussed in Chapter 4 but also for non-linear partial differential equations.

From these remarks, we can deduce that Lie's method is capable of handling a large number of equations. The application of this method depends neither on the type of the equation nor on the number of variables involved in the equations. Lie's method

is a general procedure appropriate for any type of differential equation. However, perusing the literature on Lie's method, we observe that this method has rarely been applied, compared with the wealth of differential equations in practical and theoretical problems.

The reason for the widespread rejection of Lie's method during the last hundred years by the community of mathematicians and physicists is that his method demands a huge number of algebraic calculations in order to extract the symmetries of a differential equation. Even for simple equations, the algebraic amount of calculations is large compared to other methods. If someone is genial enough to guess a solution to solve a particular problem, he probably does not have a deeper insight into the solution structure of the equation. However, if he or she is interested in a complete solution of the symmetry problem, the reader is offered the ability to obtain the information needed on an equation by using a symbolic calculation in *MathLie*. This tool allows us to completely solve the symmetry problem either in a non-interactive or an interactive way.

This chapter is organized as follows: In Section 5.2 we review Lie's method using the terminology of today. In Section 5.3, we introduce the invariance condition based on Fréchet derivatives. Section 5.4 discusses some capabilities of the package *MathLie* and presents some examples of how to use *MathLie* to find symmetries. Section 5.5 introduces the term similarity reduction. Section 5.6 is devoted to a number of applications of *MathLie*.

5.2. Lie's Theory Used in *MathLie*

In his work, Lie pointed out that the symmetry of any differential equation is defined as follows:

Definition: Lie symmetry

A Lie (point) symmetry is characterized by an infinitesimal transformation which leaves the given differential equation invariant under the transformation of all independent and dependent variables. ○

The Lie symmetries of differential equations (DEQs) naturally form a group: Since the composition of any two symmetries is also a symmetry, there is an identity transformation; the composition of symmetries is obviously associative; and any symmetry has an inverse. Such groups are called Lie groups and are invertible point transformations of both the dependent and independent variables of the DEQs. The DEQs may depend on continuous parameters. Lie pointed out that this group is of great importance in understanding and constructing solutions of DEQs. Lie

demonstrated that many techniques for finding solutions can be unified and extended by considering symmetry groups. Today, we know several applications of Lie groups in the theory of differential equations (cf. Ibragimov [1985], Bluman and Kumei [1989], Olver [1986], Ovsiannikov [1982], Ibragimov [1994–1996], Baumann [1987]).

To use the symmetry groups in any application, we first need to find the symmetries of the equations. A first approach to finding point symmetries of such systems is to make a general change of all variables and then enforce the new variables to satisfy the same set of DEQs. This approach leads to complicated non-linear systems of DEQs for the functions used in the transformations. Lie demonstrated that such a procedure is unnecessary. He established an efficient method based on an infinitesimal formulation of the problem of finding the symmetry group of a set of DEQs, replacing these highly complicated and in most cases intractable non-linear equations by tractable linear overdetermined systems of partial differential equations. The solution of these so-called infinitesimal determining equations can be used to determine symmetry transformations.

Let us consider the general case of a non-linear system of differential equations for an arbitrary number q of unknown functions u^α which may depend on p independent variables x_i. We denote these sets of variables simply by $u = (u^1, u^2, \ldots, u^q)$ and $x = (x_1, x_2, \ldots, x_p)$, respectively. The general case is given by a system of m non-linear differential equations

$$\Delta^i(x, u_{(k)}) = 0, \qquad i = 1, 2, \ldots, m \tag{5.1}$$

of order k. The term $u_{(k)}$ is understood as the kth derivative of u with respect to x. We note that m, k, p, and q are arbitrary, positive integers. Consider, further, a one-parameter ϵ-Lie group of transformations

$$x^* = \Xi(x, u, \epsilon), \tag{5.2}$$

$$u^* = \Phi(x, u, \epsilon) \tag{5.3}$$

under which (5.1) must be invariant. The star on the variables x and u denote the new variables. Invariance of (5.1) under the action of (5.2) and (5.3) means that any solution $u = \Theta(x)$ of (5.1) maps into some other solution $v = \Psi(x; \epsilon)$ of (5.1). Let $u = \Theta(x)$ be a solution of (5.1). If we replace the dependent and independent variables u and x by v and $x^* = \Xi$, respectively, equations (5.1) become

$$\Delta^i(x^*, v_{(k)}) = 0, \quad i = 1, 2, \ldots, m. \tag{5.4}$$

Then, $v = \Theta(x^*)$ are solutions of (5.4). This implies that if (5.1) and (5.4) have a unique solution, then

$$\Theta(x^*) = \Phi(x, \Theta(x), \epsilon). \tag{5.5}$$

Hence, Θ satisfies the one-parameter functional equation

$$\Theta(\Xi(x, \epsilon)) = \Phi(x, \Theta, \epsilon). \tag{5.6}$$

Expanding equations (5.2) and (5.3) around the identity $\epsilon = 0$, we can generate the following infinitesimal transformations:

$$x_i^* = x_i + \epsilon \xi_i(x, u) + O(\epsilon^2), \qquad i = 1, 2, \ldots, p \tag{5.7}$$

$$u^{*\alpha} = u^\alpha + \epsilon \phi_\alpha(x, u) + O(\epsilon^2), \qquad \alpha = 1, 2, \ldots, q \tag{5.8}$$

where the functions ξ_i and ϕ_α are the infinitesimals of the transformations for the independent and dependent variables, respectively. In order to find the unknown infinitesimals ξ_i and ϕ_α, we need to extend or prolong the transformation group to include the properties of the derivatives. It is an infinitesimal approach which considers the Lie algebra $\mathfrak{L}$ corresponding to the Lie group G. Generalizing the formulas of Chapter 4, the infinitesimal transformation (5.7) and (5.8) can be put into the form

$$\vec{v} = \sum_{i=1}^{p} \xi_i(x, u) \frac{\partial}{\partial x_i} + \sum_{\alpha=1}^{q} \phi_\alpha(x, u) \frac{\partial}{\partial u_\alpha}, \tag{5.9}$$

where $\vec{v}$ represents a linear combination of the vector fields generating $\mathfrak{L}$, which, in turn, is based on the characteristic quantities ξ_i and ϕ_α of the transformation (5.7) and (5.8). The algorithm used in *MathLie* for finding the infinitesimals ξ_i and ϕ_α is described below. We emphasize that the infinitesimals in this simple form only depend on independent and dependent variables. A prolongation of the dependencies to derivatives extends the Lie symmetries to so-called generalized Lie symmetries, which are discussed in Chapter 9. Transformations (5.7) and (5.8), together with the transformations for the first, second, ... derivatives of the u_α's, are called first, second, ... prolongations. Using these various extensions, the infinitesimal criterion for the invariance of (5.1) under the group (5.2) and (5.3) is derivable by

$$\mathrm{pr}^{(k)} \vec{v}\, \Delta \,|_{\Delta=0} = 0, \tag{5.10}$$

where the kth prolongation of the vector field $\vec{v}$ is given by

$$\mathrm{pr}^{(k)} \vec{v} = \vec{v} + \sum_{\alpha=1}^{q} \sum_{J} \phi_\alpha^J(x, u_{(k)}) \frac{\partial}{\partial u_J^\alpha}. \tag{5.11}$$

The second summation extends over all multi-indices $J = (j_1, \ldots, j_l)$ with $1 \le j_l \le p$, $1 \le l \le k$. The kth prolongation coefficients ϕ_α^J are given by

$$\phi_\alpha^J(x, u^{(k)}) = D_J\left(\phi_\alpha - \sum_{i=1}^{p} \xi_i u_i^\alpha\right) + \sum_{i=1}^{p} \xi_i u_{J,i}^\alpha , \tag{5.12}$$

where $u_i^\alpha = \partial u^\alpha / \partial x_i$ and $u_{J,i}^\alpha = \partial u_J^\alpha / \partial x_i$. Thus, the system of differential equations (5.1) is invariant under the transformation of a one-parameter group with the infinitesimal generator (5.9) if the ξ_i's and ϕ_α's are determined from equation (5.10).

So far, we discussed the standard procedure to derive the determining equations for the infinitesimals ξ_i and ϕ_α. This procedure is widely used in the literature (cf. Olver [1986], Ibragimov [1985], Bluman and Kumei [1990], and Ovsiannikov [1982]). From a calculation point of view, the procedure described above is very inefficient and time- and memory-consuming. The main slowing-down step of the procedure is the recursive calculation of the expansion coefficients in equation (5.11). The following section will discuss a more efficient way to derive the determining equations. This procedure is based on the powerful pattern-matching capability of *Mathematica* and uses the Fréchet derivative to represent the invariance condition (5.10).

5.3. Invariance Based on Fréchet Derivatives

The Fréchet derivative can be considered as a generalization of the complete derivative. In this section, we will use this type of derivative to formulate an efficient procedure for the calculation of the invariance condition used in the derivation of the determining equations.

The Fréchet derivative of a support function P with respect to a test function Q was defined in Chapter 2 by

$$\mathcal{D}_P(Q) = \frac{d}{d\epsilon} P(u + \epsilon Q)\,|_{\epsilon=0} . \tag{5.13}$$

The meaning of equation (5.13) is that in the support P, we have to replace the dependent variables and their derivatives by a variation of the original variables. The variation is represented by the variables themselves and by a test function weighted by a parameter ϵ. After the substitution, we differentiate with respect to ϵ and then set $\epsilon = 0$.

This relation defined for an r-dimensional support P and for a q-dimensional test function Q can be implemented very efficiently in *Mathematica*. The implementation was given in Section 3.5.

Let us now briefly discuss the connection between the invariance condition (5.10) and the Fréchet derivative. To calculate the determining equations, we need the prolongation of a vector field $\vec{v}_Q$ applied to the system of differential equations Δ. If we assume that the related characteristic Q depends on the dependent variables and their derivatives, we can write down a relation which connects the prolongation of a differential system with the Fréchet derivative of the system in an evolutionary representation (cf. Olver [1986]). The term evolutionary representation means that we consider infinitesimal transformations independent of the independent variables. The connection between the prolongation and the Fréchet derivative is given by

$$\mathrm{pr}^{(k)}\,\vec{v}_Q(\Delta) = \mathfrak{D}_\Delta(Q), \tag{5.14}$$

where Q is now the support and we have in mind that Δ is the system of partial differential equations. This relation follows from the definition of the prolongation in evolutionary representation:

$$\mathrm{pr}^{(k)}\,\vec{v}_Q = \sum_{\alpha=1}^{p}\sum_{J} D_J\, Q_\alpha\, \frac{\partial}{\partial u_J^\alpha} \tag{5.15}$$

with $Q_\alpha = Q_\alpha(u^{(k)})$ depending only on the derivatives of the dependent variables $k = 0, 1, \ldots$. If, in addition, we use the definition of the Fréchet derivative given by equation (5.13), we can immediately reproduce equation (5.14).

Thus, the prolongation operator is related to the Fréchet derivative. We can utilize this relation to reformulate the invariance condition (5.10). Applying relation (5.14) and the definition of the evolutionary vector field, we are able to replace the invariance condition (5.10) by the relation

$$\left(\mathfrak{D}_\Delta(Q) + \sum_{i=1}^{p} \xi_i\, D_i\, \Delta\right)\Bigg|_{\substack{\Delta=0 \\ Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i\, u_i^\alpha}} = 0. \tag{5.16}$$

At first glance, this expression appears to be very complicated from the calculation point of view. Indeed, it is very cumbersome if one tries to use this formula in a manual calculation. In fact, in a pencil calculation, we first have to replace all dependent variables and their derivatives by the variation of the dependent variables which must be extended up to the kth order in the derivatives. After the substitutions, we have to differentiate the expression obtained with respect to ϵ and afterward set $\epsilon = 0$. These steps contain a lot of work if we do them by hand. However, with *Mathematica*, all the steps are very easy to handle. This is possible because *Mathematica* offers powerful matching procedures already implemented in its kernel to carry out the calculations.

The advantage of this method to calculate the prolongation of a given system of differential equations is not only its fast calculation but also the flexibility in choosing expressions for the characteristics in the calculation which allows an extension to generalized symmetries.

All the steps given above to derive the determining equations are incorporated in the package *MathLie*. The result of the *MathLie* functions is a system of linear homogeneous partial differential equations for the infinitesimals $\xi_i = \xi_i(x, u)$ and $\phi_\alpha = \phi_\alpha(x, u)$, in which x and u are vectors of the independent and dependent variables. These are the so-called determining equations for the symmetries of the system Δ.

At this point of the discussion, we note that equation (5.16) looks very similar to the invariance condition of the non-classical symmetry method. The difference is that the second side condition Q_α is not equal to zero in relation (5.16).

In summary, in this section we discussed the algorithm in mathematical terms to calculate Lie point symmetries. The essential steps of this calculation are as follows:

1. Calculate the prolongation of the system of differential equations up to kth order by

$$\mathrm{pr}^{(k)}\, \vec{v}\, \Delta = 0. \tag{5.17}$$

2. Use the equations themselves to eliminate redundant information of the prolongation

$$\mathrm{pr}^{(k)}\, \vec{v}\, \Delta\,|_{\Delta=0} = 0. \tag{5.18}$$

3. Extract the determining equations from the prolongation by setting the coefficients of the derivatives in the dependent variables equal to zero.
4. Solve the resulting determining equations.

Steps 1 to 3 will be discussed in this section. The fourth step deals with the solution of the determining equations to be discussed in detail in Chapter 10. We note at this stage that the determining equations are always solvable in closed form since they build up an overdetermined system of linear partial differential equations.

5.4. Application of the Theory

In this section, we discuss the application of the theoretical formulas discussed to appropriately calculate Lie point symmetries. We apply the theoretical notions in terms of *MathLie* functions. The main tools discussed here are the prolongation operator and the derivation of the determining equations.

5.4.1 Calculation of Prolongations

Calculation of the prolongation in *MathLie* can easily be carried out by using the formula based on the definition of the Fréchet derivative. The theoretical concept of the prolongation is realized in *MathLie* by the function Prolongation[]. The on-line information on this function reads

```
Information["Prolongation", LongForm → False]

Prolongation[equation_,
  dependent_,independent_,parameters_:{}]
       determines
  the prolongation of an equation or a system of equations.
```

This function expects four different quantities as input. The first argument of the function contains the equations $\Delta = 0$; the second and third arguments specify the dependent and independent variables. The fourth slot contains the parameters of the equation. This input quantity is optional and can be omitted if the equation contains no parameters. The function Prolongation[] applied to an arbitrary partial differential equation $F(x, t, u, u_t) = 0$ allows us to calculate the prolongation for this expression. Let us first define the equation by

```
equat1 = F[x, t, u[x, t], ∂t u[x, t]] == 0

F[x, t, u[x, t], u^(0,1)[x, t]] == 0
```

Applying the function Prolongation[] to this expression, we have to supply the additional two arguments as well. The second and third lists contain the dependent and independent variables. Collecting this information, we can write

```
pr1 = Prolongation[equat1, {u}, {x, t}]

{u^(0,1)[x, t] phi[1]^(0,0,1)[x, t, u[x, t]]
    F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] -
  u^(0,1)[x, t] u^(1,0)[x, t] xi[1]^(0,0,1)[x, t, u[x, t]]
   F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] - u^(0,1)[x, t]^2
   xi[2]^(0,0,1)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] +
  phi[1]^(0,1,0)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] -
  u^(1,0)[x, t] xi[1]^(0,1,0)[x, t, u[x, t]]
   F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] - u^(0,1)[x, t]
   xi[2]^(0,1,0)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] +
  phi[1][x, t, u[x, t]] F^(0,0,1,0)[x, t, u[x, t], u^(0,1)[x, t]] +
  xi[2][x, t, u[x, t]] F^(0,1,0,0)[x, t, u[x, t], u^(0,1)[x, t]] +
  xi[1][x, t, u[x, t]] F^(1,0,0,0)[x, t, u[x, t], u^(0,1)[x, t]]}
```

The infinitesimals in *MathLie* are by default denoted by *xi*[*p*][*x*,*u*[*x*,*t*]] and *phi*[*q*][*x*,*u*[*x*,*t*]], where *xi* stands for the infinitesimals of the independent variables and *phi* for the dependent one. The number in square brackets denotes the first, second, third, etc. variables in the set of independent and dependent variables, respectively. Here, *xi*[1] stands for the first independent variable x and *xi*[2] for the second variable t. In our example, only one dependent variable u is present, so the related infinitesimal is *phi*[1]. The argument of the infinitesimals contains the independent and dependent variables with all the dependencies.

The result of the above calculation represents the first-order prolongation of the general equation $F(x, t, u, u_t) = 0$. Alternatively, we can use a shorthand notation for the function Prolongation[] by the symbol $\left(pr^{k}\,\vec{v}\right)^{\gamma}_{u,x}[\Delta]$. This symbol is part of a palette accompanying *MathLie*. Thus, the more symbolic representation of the prolongation is

```
prolong = (pr^k v)^{}_{{u},{x,t}} [equat1]
```

```
{u^(0,1)[x, t] phi[1]^(0,0,1)[x, t, u[x, t]]
    F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] -
  u^(0,1)[x, t] u^(1,0)[x, t] xi[1]^(0,0,1)[x, t, u[x, t]]
   F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] - u^(0,1)[x, t]^2
   xi[2]^(0,0,1)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] +
  phi[1]^(0,1,0)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] -
  u^(1,0)[x, t] xi[1]^(0,1,0)[x, t, u[x, t]]
   F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] - u^(0,1)[x, t]
   xi[2]^(0,1,0)[x, t, u[x, t]] F^(0,0,0,1)[x, t, u[x, t], u^(0,1)[x, t]] +
  phi[1][x, t, u[x, t]] F^(0,0,1,0)[x, t, u[x, t], u^(0,1)[x, t]] +
  xi[2][x, t, u[x, t]] F^(0,1,0,0)[x, t, u[x, t], u^(0,1)[x, t]] +
  xi[1][x, t, u[x, t]] F^(1,0,0,0)[x, t, u[x, t], u^(0,1)[x, t]]}
```

The result created by this operator is identical to the result derived by Prolongation[]. The function behind the operator $\left(\mathrm{pr}^{k}\,\vec{v}\right)^{\gamma}_{u,x}[\Delta]$ automatically detects the number of variables involved and creates the representation of the infinitesimals. Input quantities needed are just the independent and dependent variables and the equation. To some, the result may look somehow strange. If one likes to have the result represented in a more traditional form, the standard representation of this expression can be transform by means of LieTraditionalForm[]. The result contains Greek letters and indices for the numbering of the infinitesimals. The transformation of the previous result is carried out by the following line:

```
prolong // LieTraditionalForm
```

$$\{F_x\,\xi_1 + F_t\,\xi_2 + F_u\,\phi_1 - F_{u_t}\,u_x\,(\xi_1)_t - F_{u_t}\,u_t\,u_x\,(\xi_1)_u - F_{u_t}\,u_t\,(\xi_2)_t - F_{u_t}\,u_t^2\,(\xi_2)_u + F_{u_t}\,(\phi_1)_t + F_{u_t}\,u_t\,(\phi_1)_u\}$$

To be more specific, let us calculate the prolongation of the Burgers equation as another example. The Burgers equation, $u_t + uu_x - u_{x,x} = 0$, is one of the standard equations treated in non-linear physics. This equation was used by Burgers [1948] as a mathematical model of turbulence. The Burgers equation in the field variable

```
U = u[x, t];
```

reads

```
burgers = ∂t U + U ∂x U - ∂{x,2} U == 0; LieTraditionalForm[burgers]
```

$$u_t + u\,u_x - u_{x,x} == 0$$

The prolongation of this equation follows with the prolongation operator or the function Prolongation[] as

```
(pr^k v⃗)^{}_{u,{x,t}} [burgers] // LieTraditionalForm
```

$$\begin{aligned}&\{u_x\,\phi_1 - u_x\,(\xi_1)_t - u_t\,u_x\,(\xi_1)_u - u\,u_x^2\,(\xi_1)_u - u\,u_x\,(\xi_1)_x - u_t\,(\xi_2)_t - \\ &\quad u_t^2\,(\xi_2)_u - u\,u_t\,u_x\,(\xi_2)_u - u\,u_t\,(\xi_2)_x + (\phi_1)_t + u_t\,(\phi_1)_u + u\,u_x\,(\phi_1)_u + \\ &\quad u\,(\phi_1)_x + 2\,u_x\,(\xi_2)_u\,u_{x,t} + 2\,(\xi_2)_x\,u_{x,t} + 3\,u_x\,(\xi_1)_u\,u_{x,x} + \\ &\quad 2\,(\xi_1)_x\,u_{x,x} + u_t\,(\xi_2)_u\,u_{x,x} - (\phi_1)_u\,u_{x,x} + u_x^3\,(\xi_1)_{u,u} + \\ &\quad 2\,u_x^2\,(\xi_1)_{x,u} + u_x\,(\xi_1)_{x,x} + u_t\,u_x^2\,(\xi_2)_{u,u} + 2\,u_t\,u_x\,(\xi_2)_{x,u} + \\ &\quad u_t\,(\xi_2)_{x,x} - u_x^2\,(\phi_1)_{u,u} - 2\,u_x\,(\phi_1)_{x,u} - (\phi_1)_{x,x}\}\end{aligned}$$

The result contains derivatives of the infinitesimals ξ_1, ξ_2, and ϕ_1 related to the two independent variables and the dependent variable. We note that the numbers of the indices of the infinitesimals are related to the occurrence of the variables in the argument of u. Index 1 is connected to x and 2 denotes the second independent variable t.

Another function more flexible in its specification of the infinitesimals is FrechetProlong[]. This function allows us to supply expressions for infinitesimals. The first three arguments of the function are the same as in Prolongation[]. The difference is that, in general, we have to deliver infinitesimals in the last two arguments in such a way that there exists an expression for each variable. The position in the first list is directly related to the position of the independent variable in the arguments of dependent variables. The order in the second list depends also on the order of dependent variables in the arguments of the infinitesimals. Concerning the names of the infinitesimals, we can arbitrarily choose them. We demonstrate this by calculating the prolongation of the Harry-Dym equation. Originally a pure mathematical object, the Harry-Dym equation today is discussed in connection with physical applications (Kadanoff [1990]). It is a special feature of FrechetProlong[]

that only the left-hand sides of the equations are needed. In case of the Harry-Dym equation

$$u_t - \lambda u^3 u_{x,x,x} = 0 \tag{5.19}$$

the left-hand side of the equation is expected in a list:

```
harryDym = {∂t U - λ U^3 ∂{x,3} U}; harryDym // LTF
```

```
ut - u^3 λ ux,x,x == 0
```

The Harry-Dym equation is a third-order non-linear partial differential equation in u. λ is a real parameter in this equation. The prolongation in the infinitesimals ξ, τ, and Φ follows by

```
pharryDym = FrechetProlong[harryDym, {u}, {x, t},
  {ξ[x, t, U], τ[x, t, U]}, {Φ[x, t, U]}];
pharryDym // LieTraditionalForm
```

```
{-ux ξt - ut ux ξu - ut τt - ut^2 τu + Φt + ut Φu + 3 u^3 λ τu ux,t ux,x +
  3 u^3 λ ξu ux,x^2 + 6 u^3 λ ux^2 ux,x ξu,u + 9 u^3 λ ux ux,x ξx,u + 3 u^3 λ ux,x ξx,x +
  3 u^3 λ ux^2 ux,t τu,u + 3 u^3 λ ut ux ux,x τu,u + 6 u^3 λ ux ux,t τx,u +
  3 u^3 λ ut ux,x τx,u + 3 u^3 λ ux,t τx,x - 3 u^3 λ ux ux,x Φu,u - 3 u^3 λ ux,x Φx,u +
  3 u^3 λ ux τu ux,x,t + 3 u^3 λ τx ux,x,t - 3 u^2 λ Φ ux,x,x + 4 u^3 λ ux ξu ux,x,x +
  3 u^3 λ ξx ux,x,x + u^3 λ ut τu ux,x,x - u^3 λ Φu ux,x,x + u^3 λ ux^4 ξu,u,u +
  3 u^3 λ ux^3 ξx,u,u + 3 u^3 λ ux^2 ξx,x,u + u^3 λ ux ξx,x,x + u^3 λ ut ux^3 τu,u,u +
  3 u^3 λ ut ux^2 τx,u,u + 3 u^3 λ ut ux τx,x,u + u^3 λ ut τx,x,x -
  u^3 λ ux^3 Φu,u,u - 3 u^3 λ ux^2 Φx,u,u - 3 u^3 λ ux Φx,x,u - u^3 λ Φx,x,x}
```

This representation of the prolongation contains the infinitesimals in a more indicative form connecting the name of the independent variables with the names for the infinitesimals in Greek. However, the direct access by subscripts is lost. We see that the use of names for the infinitesimals is by no means restricted.

Another problem frequently encountered in the calculation of infinitesimal transformations is the partial knowledge of the infinitesimals. We illustrate this kind of calculation for the heat equation. The heat equation

$$u_t - u_{x,x} = 0 \tag{5.20}$$

is a second-order partial differential equation used in the description of temperature changes in solid and fluid media (Bluman and Kumei [1989]). The left-hand side of the equation for the scaled temperature field u depending on the temporal and spatial coordinate reads

```
heat = {∂t U - ∂{x,2} U}; heat // LTF
```

```
ut - ux,x == 0
```

If we know a partial representation of the infinitesimals, we can use this information to define the infinitesimals for the function FrechetProlong[]. For the heat equation, let us assume that the infinitesimals are given by linear functions in u. The infinitesimals for the independent and dependent variables are thus

```
indepInfinitesimals = {f[x, t] U + g[x, t], h[t]}
```

```
{g[x, t] + f[x, t] u[x, t], h[t]}
```

```
dependentInfinitesimals = {k[t] U}
```

```
{k[t] u[x, t]}
```

This representation of the infinitesimals contains incomplete information about the final form and thus restricts the solution manifold for the ξ_i and ϕ_α. Inserting this form for the infinitesimals in FrechetProlong[], we end up with a special representation of the prolongation:

```
pheat = FrechetProlong[heat, {u}, {x, t},
  indepInfinitesimals, dependentInfinitesimals];
LieTraditionalForm[pheat]
```

$$\{u\,k_t + k\,u_t - h_t\,u_t - u\,f_t\,u_x - g_t\,u_x - f\,u_t\,u_x + 2\,f_x\,u_x^2 + u\,u_x\,f_{x,x} + u_x\,g_{x,x} - k\,u_{x,x} + 2\,u\,f_x\,u_{x,x} + 2\,g_x\,u_{x,x} + 3\,f\,u_x\,u_{x,x}\}$$

The result gained is an expression containing functions f, g, h, and k. If we know these arbitrary functions, we can check the invariance of the equation directly. It is sufficient to know a subgroup of the complete group to check the invariance. It is well known that the heat equation is invariant with respect to translations (Bluman and Kumei [1989]). As we know from Chapter 2, these symmetries are represented in infinitesimal form by

```
indepndInfinitesimls = {k1, k2}
```

```
{k1, k2}
```

```
dependentInfinitesimals = {k3}
```

```
{k3}
```

The group constants *k1*, *k2*, and *k3* are real constants. Inserting these infinitesimals in our function FrechetProlong[], we find

```
pheats = FrechetProlong[heat, {u}, {x, t},
  indepndInfinitesimls, dependentInfinitesimals]
```

```
{0}
```

The result reveals that the heat equation is invariant with respect to translations. It is now easy to manually check other types of symmetries for the heat equation. We only have to specify the infinitesimals in the function FrechetProlong[]. Let us assume another type of invariance to be given by a rotation of the independent variables and a translation of the dependent variable. The check of these hypothetical infinitesimals

```
pheatss = FrechetProlong[heat, {u}, {x, t},
  {-k1 t, k1 x}, {k2}]; pheatss // LTF
```

$$k1\, u_x + 2\, k1\, u_{x,t} == 0$$

shows that the heat equation is not invariant with respect to rotations in the independent variables.

The function FrechetProlong[] can be used not only to derive the prolongation of an equation but also to calculate the expansion coefficients of the prolongation in general. For example, if we need the general representation of the first coefficient of the prolongation related to variable t, we construct this term by

```
firstExtension = FrechetProlong[{∂t u[x, t]}, {u}, {x, t},
  {xi[1][x, t, u[x, t]], xi[2][x, t, u[x, t]]},
  {phi[1][x, t, u[x, t]]}];
firstExtension // LieTraditionalForm
```

$$\{-u_x\,(\xi_1)_t - u_t\,u_x\,(\xi_1)_u - u_t\,(\xi_2)_t - u_t^2\,(\xi_2)_u + (\phi_1)_t + u_t\,(\phi_1)_u\}$$

The result represents the general formula for the first extension with respect to t. This sort of expression is tabulated in the book of Bluman and Cole [1974] and is now available to any order or number of variables. The general expression of the second prolongation with respect to x follows from

```
secondExtension = FrechetProlong[{∂{x,2} u[x, t]},
  {u}, {x, t},
  {xi[1][x, t, u[x, t]], xi[2][x, t, u[x, t]]},
  {phi[1][x, t, u[x, t]]}];
secondExtension // LieTraditionalForm
```

$$\begin{aligned}\{&-2\,u_x\,(\xi_2)_u\,u_{x,t} - 2\,(\xi_2)_x\,u_{x,t} - 3\,u_x\,(\xi_1)_u\,u_{x,x} - \\ &2\,(\xi_1)_x\,u_{x,x} - u_t\,(\xi_2)_u\,u_{x,x} + (\phi_1)_u\,u_{x,x} - u_x^3\,(\xi_1)_{u,u} - \\ &2\,u_x^2\,(\xi_1)_{x,u} - u_x\,(\xi_1)_{x,x} - u_t\,u_x^2\,(\xi_2)_{u,u} - 2\,u_t\,u_x\,(\xi_2)_{x,u} - \\ &u_t\,(\xi_2)_{x,x} + u_x^2\,(\phi_1)_{u,u} + 2\,u_x\,(\phi_1)_{x,u} + (\phi_1)_{x,x}\}\end{aligned}$$

Comparing the results with expressions given in Bluman and Cole [1974], it is obvious that the formulas are identical. The reader may calculate, for example, the fifth expansion coefficient related to terms $u_{x,x,t,t,t}$.

The function FrechetProlong[] is also capable of calculating the prolongation of a general expression containing derivatives. Let us demonstrate this behavior by examining the general partial differential equation of second order given by the relation

$$F(x, t, u, u_x, u_t, u_{x,x}) = 0. \tag{5.21}$$

The left-hand side of this general second-order equation in *Mathematica* reads

```
pde2 = {F[x, t, U, ∂x U, ∂t U, ∂{x,2} U]}
```

```
{F[x, t, u[x, t], u^(1,0)[x, t], u^(0,1)[x, t], u^(2,0)[x, t]]}
```

The prolongation formula for this general PDE of second-order follows from

```
PrintDf[FrechetProlong[pde2, {u}, {x, t},
    {xi[1][x, t, u[x, t]], xi[2][x, t, u[x, t]]},
    {phi[1][x, t, u[x, t]]}] // LieTraditionalForm]
```

$$\begin{aligned}
&\{F_x\,\xi_1 + F_t\,\xi_2 + F_u\,\phi_1 - F_{u_t}\,u_x\,(\xi_1)_t - F_{u_t}\,u_t\,u_x\,(\xi_1)_u - F_{u_x}\,u_x^2\,(\xi_1)_u - \\
&F_{u_x}\,u_x\,(\xi_1)_x - F_{u_t}\,u_t\,(\xi_2)_t - F_{u_t}\,u_t^2\,(\xi_2)_u - F_{u_x}\,u_t\,u_x\,(\xi_2)_u - \\
&F_{u_x}\,u_t\,(\xi_2)_x + F_{u_t}\,(\phi_1)_t + F_{u_t}\,u_t\,(\phi_1)_u + F_{u_x}\,u_x\,(\phi_1)_u + F_{u_x}\,(\phi_1)_x - \\
&2\,F_{u_{x,x}}\,u_x\,(\xi_2)_u\,u_{x,t} - 2\,F_{u_{x,x}}\,(\xi_2)_x\,u_{x,t} - 3\,F_{u_{x,x}}\,u_x\,(\xi_1)_u\,u_{x,x} - \\
&2\,F_{u_{x,x}}\,(\xi_1)_x\,u_{x,x} - F_{u_{x,x}}\,u_t\,(\xi_2)_u\,u_{x,x} + F_{u_{x,x}}\,(\phi_1)_u\,u_{x,x} - \\
&F_{u_{x,x}}\,u_x^3\,(\xi_1)_{u,u} - 2\,F_{u_{x,x}}\,u_x^2\,(\xi_1)_{x,u} - F_{u_{x,x}}\,u_x\,(\xi_1)_{x,x} - \\
&F_{u_{x,x}}\,u_t\,u_x^2\,(\xi_2)_{u,u} - 2\,F_{u_{x,x}}\,u_t\,u_x\,(\xi_2)_{x,u} - F_{u_{x,x}}\,u_t\,(\xi_2)_{x,x} + \\
&F_{u_{x,x}}\,u_x^2\,(\phi_1)_{u,u} + 2\,F_{u_{x,x}}\,u_x\,(\phi_1)_{x,u} + F_{u_{x,x}}\,(\phi_1)_{x,x}\}
\end{aligned}$$

The result of this calculation was converted to a more readable form in index notation by the function LieTraditionalForm[]. This function reduces the standard *Mathematica* output to a shorter representation by deleting the arguments of any derivative and using the variables of differentiation as an index. The expressions free of any derivatives remain unchanged.

So far, we discussed some applications of the functions Prolongation[] and FrechetProlong[], allowing us to derive the prolongation of a differential equation. As we know from the theoretical considerations, the prolongation of a differential equation is the basis for the derivation of determining equations. In the following section, we will discuss a function of *MathLie* which is instrumental in the derivation of determining equations.

5.4.2 Derivation of Determining Equations

Determining equations for infinitesimals are the result of invariance condition (5.10). The package *MathLie* provides a function allowing us to derive determining

equations for a given system of differential equations. The name of the function is DeterminingEquations[]. The on-line description of the function

```
Information["DeterminingEquations", LongForm → False]

DeterminingEquations[equations_List, dependvar_List,
  independvar_List,substitutionTerms_List,parameters_List:
  {}] calculates the determining equations for a given
  system of equations. The function uses the Frechet
  derivative to calculate the prolongation.
```

tells us that we need five input arguments. The first argument contains the left-hand side of the equation $\Delta = 0$. The second and third arguments are lists for the dependent and independent variables. The fourth list contains terms for which the equation $\Delta = 0$ is solved. The solutions with respect to these terms are used as side conditions in the invariance relation (5.10). If the equations under examination contain parameters, we can feed in these symbols in the last list. This list can be suppressed if no parameters are contained in the PDE. The function DeterminingEquations[] uses the function FrechetProlong[] to calculate the kth prolongation of the equations. After the calculation of the prolongation, the side conditions are applied to the result of the function FrechetProlong[]. This step reduces the redundant information in the manifold of the equation. Upon application of the side conditions, the determining equations are extracted as coefficients of the derivatives of the dependent variables. Since the infinitesimals themselves are independent of derivatives, we find the determining equations as a set of coupled PDEs.

Application of the function DeterminingEquations[] is demonstrated by the heat equation. The determining equations of the infinitesimals for the heat equation follow from

```
detheat = DeterminingEquations[heat, {u}, {x, t},
  {∂t U}]; detheat // LTF

(ξ1)u == 0
(ξ2)u == 0
(ϕ1)u,u == 0
(ξ2)x == 0
-(ξ1)t + (ξ1)x,x - 2 (ϕ1)x,u == 0
(ϕ1)t - (ϕ1)x,x == 0
2 (ξ1)x - (ξ2)t == 0
```

The result of this calculation *detheat* consists of a list containing the left-hand sides of the seven determining equations. We transformed these expressions to a system of equations by LTF[] adding zero to the right-hand side, and the result is displayed in a

table. The seven equations contain the unknown functions ξ_1, ξ_2, and ϕ_1. The function DeterminingEquations[] automatically implants these names for the infinitesimals. The unknown functions ξ_1, ξ_2, and ϕ_1 depend on the independent variables x and t and on the dependent variable u. The symmetries of the heat equation are determined by this set of equations.

Taking a closer look at these equations, we realize that they are linear but coupled. However, the main observation is that they are linear. Linearity of the equations is a general feature of the determining equations for point symmetries. This feature is of great advantage in solving the equations. Another general property of the determining equations is that this set of equations is always overdetermined. This means that, in general, there exist more equations than unknown functions. This fact helps a lot in the derivation of the solution.

In the present case of the heat equation, we find seven equations for three unknowns ξ_1, ξ_2, and ϕ_1. Another example demonstrating these two general properties again is the general second-order partial differential equation

```
Clear[F]

gheat = {∂t U + F[∂{x,2} U]}; gheat // LTF
```

$$F + u_t == 0$$

representing one of many generalizations of the heat equation. If we apply the function DeterminingEquations[] to this equation, we find

```
detgheat = DeterminingEquations[gheat, {u}, {x, t}, {∂t U}];
detgheat // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\phi_1)_t == 0$$
$$(\xi_1)_t == 0$$
$$(\xi_2)_x == 0$$
$$(\phi_1)_{x,x} == 0$$
$$-(\xi_1)_{x,x} + 2(\phi_1)_{x,u} == 0$$
$$-2(\xi_1)_x + (\phi_1)_u == 0$$
$$(\xi_2)_t - (\phi_1)_u == 0$$

Looking at this system of 10 equations, we recognize the same 2 properties as in the example for the heat equation linearity and a larger number of equations than unknown functions. The equations are linear in the infinitesimals independent of the form of general function F, and they are overdetermined. This general behavior does

not change if we examine non-linear equations like the Burgers equation or the Harry-Dym equation. The determining equations for these two models follow by

```
DeterminingEquations[{burgers[[1]]}, {u}, {x, t}, {∂t U}] // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$
$$\phi_1 - (\xi_1)_t - u\,(\xi_1)_x + u\,(\xi_2)_t + (\xi_1)_{x,x} - 2\,(\phi_1)_{x,u} == 0$$
$$(\phi_1)_t + u\,(\phi_1)_x - (\phi_1)_{x,x} == 0$$
$$2\,(\xi_1)_x - (\xi_2)_t == 0$$

which are six determining equations in the case of the Burgers equation

```
DeterminingEquations[harryDym, {u}, {x, t}, {∂t U}, {λ}] // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$
$$-(\xi_1)_t + u^3\,\lambda\,(\xi_1)_{x,x,x} - 3\,u^3\,\lambda\,(\phi_1)_{x,x,u} == 0$$
$$(\phi_1)_t - u^3\,\lambda\,(\phi_1)_{x,x,x} == 0$$
$$-(\xi_1)_{x,x} + (\phi_1)_{x,u} == 0$$
$$3\,\phi_1 - 3\,u\,(\xi_1)_x + u\,(\xi_2)_t == 0$$

and eight determining equations for the Harry-Dym equation. Since the Harry-Dym equation contains a parameter λ, we have to tell the function DeterminingEquations[] that λ is a variable in the equation which does not depend on the independent variables. The last argument of the function DeterminingEquations[] contains this information.

Up to now, we discussed equations containing only a single dependent variable. The following example examines a system of two equations. The physical background of these equations is the flow in a polytropic gas. Following Ibragimov [1985], we can write down the equations of motion for polytropic gas in two spatial dimensions and one temporal dimension. In polar coordinates, the radial and angular velocity fields are

```
Vr = vr[r, θ, t];
```

```
Vθ = vθ[r, θ, t];
```

The depth h of the fluid above a flat bottom is given by

```
H = h[r, θ, t];
```

The equations of motion for this fluid are given by

```
poly = {∂t Vr + Vr ∂r Vr + (Vθ ∂θ Vr)/r + ∂r H,
  ∂t Vθ + Vr ∂r Vθ + (Vθ ∂θ Vθ)/r + (∂θ H)/r,
  ∂t H + Vr ∂r H + (Vθ ∂θ H)/r + H ((∂r (Vr r))/r + (∂θ Vθ)/r)}; poly // LTF
```

$$h_r + v_r\,(v_r)_r + (v_r)_t + \frac{v_\theta\,(v_r)_\theta}{r} == 0$$

$$\frac{h_\theta}{r} + v_r\,(v_\theta)_r + (v_\theta)_t + \frac{v_\theta\,(v_\theta)_\theta}{r} == 0$$

$$h_t + h_r\,v_r + \frac{h_\theta\,v_\theta}{r} + h\left(\frac{v_r + r\,(v_r)_r}{r} + \frac{(v_\theta)_\theta}{r}\right) == 0$$

This system consists of three equations for the dependent variables *vr*, *vθ*, and *h*. The independent variables are the radius *r* and the angle *θ*. The determining equations for the polytropic gas follow by specifying the knowledge on all the variables and the equations in the function DeterminingEquations[]. The equations do not depend on a parameter, so the parameter list is empty.

```
DeterminingEquations[poly,
  {vr, vθ, h}, {r, θ, t}, {∂t Vr, ∂t Vθ, ∂t H}, {}] //
 LTF
```

$$(\xi_1)_h == 0$$

$$(\xi_2)_h == 0$$

$$(\xi_3)_h == 0$$

$$(\phi_3)_{v_\theta} == 0$$

$$(\xi_1)_{v_\theta} == 0$$

$$(\xi_2)_{v_\theta} == 0$$

$$(\xi_3)_{v_\theta} == 0$$

$$(\phi_3)_{v_r} == 0$$

$$(\xi_1)_{v_r} == 0$$

$$(\xi_2)_{v_r} == 0$$

$$(\xi_3)_{v_r} == 0$$

$$-v_\theta\,\xi_1 + r\,\phi_2 - r^2\,v_r\,(\xi_2)_r - r^2\,(\xi_2)_t - r\,v_\theta\,(\xi_2)_\theta + r\,v_r\,v_\theta\,(\xi_3)_r + r\,v_\theta\,(\xi_3)_t + h\,(\xi_3)_\theta + v_\theta^2\,(\xi_3)_\theta + h\,r\,(\phi_2)_h == 0$$

$$-v_\theta\,\xi_1 + r\,\phi_2 - r^2\,v_r\,(\xi_2)_r - r^2\,(\xi_2)_t - r\,v_\theta\,(\xi_2)_\theta + r\,v_r\,v_\theta\,(\xi_3)_r + r\,v_\theta\,(\xi_3)_t + h\,(\xi_3)_\theta + v_\theta^2\,(\xi_3)_\theta - h\,r\,(\phi_2)_h == 0$$

$$-v_\theta\,\xi_1 + r\,\phi_2 - r^2\,v_r\,(\xi_2)_r - r^2\,(\xi_2)_t - r\,v_\theta\,(\xi_2)_\theta + r\,v_r\,v_\theta\,(\xi_3)_r + r\,v_\theta\,(\xi_3)_t + v_\theta^2\,(\xi_3)_\theta == 0$$

$$-r\,(\xi_2)_r + v_\theta\,(\xi_3)_r + (\phi_2)_{v_r} == 0$$

$$r\,(\xi_2)_r - v_\theta\,(\xi_3)_r + (\phi_1)_{v_\theta} == 0$$

$$-h\,v_r\,\xi_1 + h\,r\,\phi_1 + r\,v_r\,\phi_3 + h\,r\,v_r^2\,(\xi_3)_r + h\,r\,v_r\,(\xi_3)_t + h\,v_r\,v_\theta\,(\xi_3)_\theta + h\,r^2\,(\phi_1)_r + h\,r\,(\phi_2)_\theta - h\,r\,v_r\,(\phi_3)_h + r^2\,v_r\,(\phi_3)_r + r^2\,(\phi_3)_t + r\,v_\theta\,(\phi_3)_\theta == 0$$

$$r\,\phi_1 - r\,v_r\,(\xi_1)_r - r\,(\xi_1)_t - v_\theta\,(\xi_1)_\theta + h\,r\,(\xi_3)_r + r\,v_r^2\,(\xi_3)_r + r\,v_r\,(\xi_3)_t + v_r\,v_\theta\,(\xi_3)_\theta + h\,r\,(\phi_1)_h == 0$$

$$r\,\phi_1 - r\,v_r\,(\xi_1)_r - r\,(\xi_1)_t - v_\theta\,(\xi_1)_\theta + h\,r\,(\xi_3)_r + r\,v_r^2\,(\xi_3)_r + r\,v_r\,(\xi_3)_t + v_r\,v_\theta\,(\xi_3)_\theta - h\,r\,(\phi_1)_h == 0$$

$$r\,\phi_1 - r\,v_r\,(\xi_1)_r - r\,(\xi_1)_t - v_\theta\,(\xi_1)_\theta + r\,v_r^2\,(\xi_3)_r + r\,v_r\,(\xi_3)_t + v_r\,v_\theta\,(\xi_3)_\theta == 0$$

$$r\,\phi_3 - h\,r\,(\xi_1)_r + 2\,h\,r\,v_r\,(\xi_3)_r + h\,r\,(\xi_3)_t + h\,v_\theta\,(\xi_3)_\theta + h\,r\,(\phi_1)_{v_r} - h\,r\,(\phi_3)_h == 0$$

$$h\,v_r\,(\xi_3)_r - h\,v_r\,(\phi_1)_h + r\,v_r\,(\phi_1)_r + r\,(\phi_1)_t + v_\theta\,(\phi_1)_\theta + r\,(\phi_3)_r == 0$$

$$-h\,\xi_1 + r\,\phi_3 - h\,r\,(\xi_2)_\theta + h\,r\,v_r\,(\xi_3)_r + h\,r\,(\xi_3)_t + 2\,h\,v_\theta\,(\xi_3)_\theta + h\,r\,(\phi_2)_{v_\theta} - h\,r\,(\phi_3)_h == 0$$

$$-r\,(\xi_1)_r + 2\,r\,v_r\,(\xi_3)_r + r\,(\xi_3)_t + v_\theta\,(\xi_3)_\theta - r\,(\phi_1)_{v_r} + r\,(\phi_3)_h == 0$$

$$-\xi_1 - r\,(\xi_2)_\theta + r\,v_r\,(\xi_3)_r + r\,(\xi_3)_t + 2\,v_\theta\,(\xi_3)_\theta - r\,(\phi_2)_{v_\theta} + r\,(\phi_3)_h == 0$$

$$-(\xi_3)_r + (\phi_1)_h == 0$$

$$h\,v_r\,(\xi_3)_\theta - h\,r\,v_r\,(\phi_2)_h + r^2\,v_r\,(\phi_2)_r + r^2\,(\phi_2)_t + r\,v_\theta\,(\phi_2)_\theta + r\,(\phi_3)_\theta == 0$$

$$-(\xi_1)_\theta + v_r\,(\xi_3)_\theta + r\,(\phi_1)_{v_\theta} == 0$$

$$(\xi_1)_\theta - v_r\,(\xi_3)_\theta + r\,(\phi_2)_{v_r} == 0$$

$$-(\xi_3)_\theta + r\,(\phi_2)_h == 0$$

The resulting 29 equations are again linear. The symmetries of polytropic gas follow by solving this set of overdetermined equations. For polytropic gas equations, we not only have to provide three dependent variables but also three terms in the list for the substitutions. Compared with the examples discussed above, the typing is a little bit tedious, but the gain of the calculation is greater than this little hazel. To simplify the input of the information *MathLie* provides a template for the function DeterminingEquations[] which looks like $\mathcal{D}et\mathrm{Eq}_{u,x}^{\delta,\gamma}[\Delta]$. This operator has the same functionality as function DeterminingEquations[] itself. In fact it derives the determining equations using the function DeterminingEquations[]. The following example shows the location of the input variables for the Harry-Dym equation.

```
detHarryDym = DetEq[{∂t v},{λ}][{u},{x,t}] [harryDym] ; detHarryDym // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$
$$-(\xi_1)_t + u^3\,\lambda\,(\xi_1)_{x,x,x} - 3\,u^3\,\lambda\,(\phi_1)_{x,x,u} == 0$$
$$(\phi_1)_t - u^3\,\lambda\,(\phi_1)_{x,x,x} == 0$$
$$-(\xi_1)_{x,x} + (\phi_1)_{x,u} == 0$$
$$3\,\phi_1 - 3\,u\,(\xi_1)_x + u\,(\xi_2)_t == 0$$

So far, we have been able to calculate the determining equations for a given system of partial differential equations. The question arises of how to solve these equations. The following section will discuss an interactive procedure to construct solutions for the determining equations. In Chapter 10, we will discuss procedures allowing the automatic solution of the determining equations.

5.4.3 Interactive Solution of Determining Equations

In the above discussions, we found a lot of equations determining the infinitesimal transformations of the different models. We realized that a common property of all these systems of equations was their linearity. Since determining equations are linear, contrary to the equations we started from (compare with the Burgers or Harry-Dym equations), we expect that the linear equations can be solved more easily. How to tackle this problem interactively by *MathLie* is the content of this section. We will show you a way which is similar to the automatic procedure of solving equations.

Let us demonstrate the interactive solution steps for the heat equation. The seven determining equations for the heat equation are

```
detheat // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$
$$-(\xi_1)_t + (\xi_1)_{x,x} - 2\,(\phi_1)_{x,u} == 0$$
$$(\phi_1)_t - (\phi_1)_{x,x} == 0$$
$$2\,(\xi_1)_x - (\xi_2)_t == 0$$

Before discussing the solution of these equations, let us introduce some simplification. During the solution steps of these equations, the variables x, t, and u are taking the role of independent variables. Taking this behavior into account in our calculations, we can remove the dependencies in u. In *Mathematica*, we just replace the dependent variable by the variable itself:

```
detheat = detheat /. u[x, t] → u
```

```
{xi[1]^(0,0,1)[x, t, u], xi[2]^(0,0,1)[x, t, u], phi[1]^(0,0,2)[x, t, u],
 xi[2]^(1,0,0)[x, t, u], -xi[1]^(0,1,0)[x, t, u] -
  2 phi[1]^(1,0,1)[x, t, u] + xi[1]^(2,0,0)[x, t, u],
 phi[1]^(0,1,0)[x, t, u] - phi[1]^(2,0,0)[x, t, u],
 -xi[2]^(0,1,0)[x, t, u] + 2 xi[1]^(1,0,0)[x, t, u]}
```

This simplifies a little the representation of the equations but does not solve them. If we look at the first four equations, we observe that the infinitesimals ξ_1, ξ_2, and ϕ_1 are reduced to special presolutions resulting especially from single terms. For example, the first two equations state that the infinitesimals for the independent variables are independent of the dependent variable u. The fourth equation says that, in addition, ξ_2 is independent of x. The third equation in the list *detheat* suggests that the infinitesimal ϕ_1 is linear in u. All this information can be collected in rules allowing us to simplify the determining equations:

```
infini1 = {xi[1] → Function[{x, t, u}, xi[1][x, t]],
   xi[2] → Function[{x, t, u}, xi[2][t]],
   phi[1] → Function[{x, t, u}, f1[x, t] u + f2[x, t]]}
```

```
{xi[1] → Function[{x, t, u}, xi[1][x, t]],
 xi[2] → Function[{x, t, u}, xi[2][t]],
 phi[1] → Function[{x, t, u}, f1[x, t] u + f2[x, t]]}
```

where *f1* and *f2* are two arbitrary functions depending on x and t. Inserting this primal representation of the infinitesimals into the determining equations, we end up with the following system:

```
detheat1 = detheat /. infini1; detheat1 // LTF
```

```
True
True
True
True
-2 f1_x - (ξ1)_t + (ξ1)_x,x == 0
u f1_t + f2_t - u f1_x,x - f2_x,x == 0
2 (ξ1)_x - (ξ2)_t == 0
```

The seven equations reduce to three equations for the unknown functions ξ_1, ξ_2, $f1$, and $f2$. Considering the last equation of this set by differentiating with respect to x,

```
∂x Last[detheat1] // LTF
```

```
2 (ξ1)_x,x == 0
```

we realize that a single term remains. This term defines a partial differential equation of second order for ξ_1. The solution of this equation is given by a linear function in x. Thus, we define

```
infini2 = {xi[1] → Function[{x, t}, g1[t] x + g2[t]]}
```

```
{xi[1] → Function[{x, t}, g1[t] x + g2[t]]}
```

Inserting this partial solution again into the reduced determining equations *detheat1*, we can simplify the determining equations a second time. The resulting equations read

```
detheat2 = detheat1 /. infini2; detheat2 // LTF
```

```
True
True
True
True
-2 f1_x - x g1_t - g2_t == 0
u f1_t + f2_t - u f1_x,x - f2_x,x == 0
2 g1 - (ξ2)_t == 0
```

An integration of the fifth element of the list *detheat2* with respect to x gives us

```
integ = ∫detheat2[[5]] dx - g3[t]; integ // LTF
```

$$-2\,f_1 - g_3 - \frac{x^2\,g1_t}{2} - x\,g2_t == 0$$

meaning that function $f1$ can be expressed by

```
sol = Flatten[Solve[integ == 0, f1[x, t]] /. f1[x, t] → w]
```

$$\left\{w \to \frac{1}{4}\left(-2\,g3[t] - x^2\,g1'[t] - 2\,x\,g2'[t]\right)\right\}$$

which is converted in a pure function by

```
infini3 = f1 → Function[{x, t}, w] /. sol
```

$$f1 \to \text{Function}\left[\{x, t\}, \frac{1}{4}\left(-2\,g3[t] - x^2\,g1'[t] - 2\,x\,g2'[t]\right)\right]$$

Inserting the result again in the reduced set of determining equations, we find

```
detheat3 = Simplify[detheat2 /. infini3]; detheat3 // LTF
```

```
True
True
True
True
True
```

$$f2_t + \frac{u\,g1_t}{2} - f2_{x,x} - \frac{1}{4}u\,(2\,g3_t + x\,(x\,g1_{t,t} + 2\,g2_{t,t})) == 0$$

$$2\,g_1 - (\xi_2)_t == 0$$

Apart from *xi*[2], the equations contain only relations for the auxiliary functions *g1*, *g2*, *g3*, and *f2*. Extracting the coefficients of u, $u\,x$, and $u\,x^2$, we get the following set of equations:

```
equat = Table[Coefficient[detheat3, u x^i], {i, 2, 1, -1}];
equat // LieTraditionalForm
```

$$\left\{\left\{0, 0, 0, 0, 0, -\frac{g1_{t,t}}{4}, 0\right\}, \left\{0, 0, 0, 0, 0, -\frac{g2_{t,t}}{2}, 0\right\}\right\}$$

```
AppendTo[equat, Coefficient[detheat3 /. x → 0, u]];

AppendTo[equat, (detheat3 /. u → 0)[[6]]];
equat // LieTraditionalForm
```

$$\left\{\left\{0, 0, 0, 0, 0, -\frac{g1_{t,t}}{4}, 0\right\}, \left\{0, 0, 0, 0, 0, -\frac{g2_{t,t}}{2}, 0\right\},\right.$$
$$\left.\left\{0, 0, 0, 0, 0, \frac{g1_t}{2} - \frac{g3_t}{2}, 0\right\}, f2_t - f2_{x,x}\right\}$$

```
h1 = DeleteCases[Flatten[equat], 0]; h1 // LTF
```

$$-\frac{g1_{t,t}}{4} == 0$$

$$-\frac{g2_{t,t}}{2} == 0$$

$$\frac{g1_t}{2} - \frac{g3_t}{2} == 0$$

$$f2_t - f2_{x,x} == 0$$

The final set of determining equations is solved step by step using the function DSolve[]. By renaming the constants of integration, we prevent misinterpretations of the group parameters. Function *g1* follows from the first equation of the set *h1* by

```
s1 = DSolve[h1[[1]] == 0, g1, t] /. {C[1] → k1, C[2] → k2}

{{g1 → (k1 + k2 #1&)}}
```

The second auxiliary function *g2* is

```
s2 = DSolve[h1[[2]] == 0, g2, t] /. {C[1] → k3, C[2] → k4}

{{g2 → (k3 + k4 #1&)}}
```

The last of the auxiliary functions *g3* reads

```
s3 = DSolve[h1[[3]] == 0 /. s1, g3, t] /. {C[1] → k5}

{{g3 → (k5 + k2 #1&)}}
```

Knowing the representations of functions *g1*, *g2*, and *g3*, we can integrate the remaining equation of the determining system

```
s4 = DSolve[(Last[detheat3] /. s1)[[1]] == 0, xi[2], t] /.
  C[1] → k6

{{xi[2] → (k6 + 2 k1 #1 + k2 #1^2&)}}
```

Knowledge of the auxiliary functions allows us to write down the solutions for the infinitesimals:

```
infinitesimals = Flatten[{xi[1][x, t, u],
   xi[2][x, t, u], phi[1][x, t, u]} /. infini1 /.
   infini2 /. infini3 /. s1 /. s2 /. s3 /. s4]

{k3 + k4 t + (k1 + k2 t) x, k6 + 2 k1 t + k2 t^2,
 1/4 u (-2 (k5 + k2 t) - 2 k4 x - k2 x^2) + f2[x, t]}
```

The result shows that the infinitesimals depend on six parameters *k1–k6*, which are the group parameters of the symmetry group. In addition to these six parameters, the infinitesimals contain the auxiliary function *f2* which satisfies the heat equation. The heat equation remains as a last condition in variable *h1*. The symmetry represented by *f2* is related to an infinite dimensional group. This infinite dimensional group is characteristic for linear partial differential equations. The subgroups determined by one of the parameters *k1–k6* are related to translations, scalings, and Galilean boosts.

We demonstrated by the above calculation how the linear coupled system of partial differential equations is solved by using simple integration steps. The package *MathLie* offers a function to carry out all these simple steps in one shot. The name of the related *MathLie* function is Infinitesimals[]. This function allows the automatic derivation of the infinitesimal transformations. The shorthand description of the function shows us the information needed to carry out the calculation:

```
Information["Infinitesimals", LongForm → False]
```

```
Infinitesimals[equations_,
  dependentVariables_, independentVariables_,parameters_,
  options___] The function Infinitesimals calculates the
  point symmetries of a given system of equations.
  The results of the calculation are not saved in a
  file. They are available in a pure function representation.
```

The application of this function to the heat equation reads

```
Infinitesimals[heat, {u}, {x, t}, {}]
```

$$\left\{\left\{\text{xi}[1] \to \text{Function}[\{x, t, u\}, k5 - 2\,k2\,t + k6\,x + k4\,t\,x],\ \text{xi}[2] \to \text{Function}[\{x, t, u\}, k3 + t\,(2\,k6 + k4\,t)],\ \text{phi}[1] \to \text{Function}\left[\{x, t, u\}, u\left(k1 - \frac{k4\,t}{2} + k2\,x - \frac{k4\,x^2}{4}\right) + \text{free}[1][x, t]\right]\right\},\ \left\{-\text{free}[1]^{(0,1)}[x, t] + \text{free}[1]^{(2,0)}[x, t]\right\}\right\}$$

The result of the function Infinitesimals[] consists of a nested list. The first part of the list contains the infinitesimals; the second, a list of remaining equations. The remaining equations of the second part are equations which are not solved by the *MathLie* package. The function *free[1]* occurring in the remaining equation is also part of the infinitesimals. This result has to be expected since the original equation is a linear equation which has to satisfy the superposition principle reflected by the occurrence of the auxiliary function *free[1]*. The first list of the result circumscribes the infinitesimals in a pure function representation. This cast allows direct substitution of the infinitesimals into any equation containing the infinitesimals or their derivatives. The result found for the heat equation represents a six-dimensional finite group which is isomorphic to the finite group derived by our manual calculations.

Having the function Infinitesimals[] available, we are able to determine the point symmetries of the other equations discussed above. For example, the Harry-Dym equation allows the symmetries

```
symharryDym = Infinitesimals[
  ∂t u[x, t] - λ u[x, t]^3 ∂x,x,x u[x, t] == 0, {u}, {x, t}, {λ}]

{xi[1] → Function[{x, t, u}, k3 + x (k4 + k5 x)],
 xi[2] → Function[{x, t, u}, k1 + k2 t],
 phi[1] → Function[{x, t, u}, u (-k2/3 + k4 + 2 k5 x)]}
```

The result shows that the Harry-Dym equation allows a five-dimensional finite group independent of the parameter λ. Group parameters *k3* and *k1* represent the invariance of the Harry-Dym equation with respect to translations. The scaling symmetries are determined by the parameters *k4* and *k2*. The remaining parameter, *k5*, represents a non-standard conformal transformation.

Another example for a calculation of point symmetries is given by the two-dimensional polytropic gas discussed above. The symmetries follow by

```
sympoly = Infinitesimals[poly, {vr, vθ, h}, {r, θ, t}]

{xi[1] → Function[{r, θ, t, vr, vθ, h}, (k2 + k3) r],
 xi[2] → Function[{r, θ, t, vr, vθ, h}, -k4],
 xi[3] → Function[{r, θ, t, vr, vθ, h}, k1 + k2 t],
 phi[3] → Function[{r, θ, t, vr, vθ, h}, 2 h k3],
 phi[1] → Function[{r, θ, t, vr, vθ, h}, k3 vr],
 phi[2] → Function[{r, θ, t, vr, vθ, h}, k3 vθ]}
```

The resulting point symmetries are given by a four-dimensional group. The symmetry transformations are translations in the time coordinate and the angular direction. The related group parameters are *k1* and *k4*. In addition to the translations, the polytropic gas enables a scaling of the radial and temporal coordinate denoted by *k2*. Group parameter *k3* represents a special type of scaling.

The function Infinitesimals[] can not only explicitly treat given equations but is also capable of analyzing equations of a general form, like

$$u_t - F(x, u, u_x, u_{x,x}) = 0, \tag{5.22}$$

where F is an arbitrary function depending on a set of independent variables x and a set of dependent variables u. An example of such an equation for one dependent and two independent variables is given by the general second-order equation

```
geneq = {∂t U - F[x, U, ∂x U, ∂x,x U]}

{-F[x, u[x, t], u^(1,0)[x, t], u^(2,0)[x, t]] + u^(0,1)[x, t]}
```

The corresponding infinitesimals for this type of equation follow by

```
symgeneq = Infinitesimals[geneq, {u}, {x, t}]

{phi[1] → Function[{x, t, u}, 0], xi[1] → Function[{x, t, u}, 0],
 xi[2] → Function[{x, t, u}, k1]}
```

representing a one-dimensional symmetry group. Parameter *k1* characterizes the translations under which the general evolution equation is invariant. Let us now specify the auxiliary function F in a more explicit form. For example, let us assume that F is replaced by a function f independent of u and x. We further assume that the second derivative in u with respect to x is created by a derivative of f with respect to x. Taking into account all these assumptions, we end up with an expression for a general nonlinear diffusion equation:

```
geneqe = {∂t U - ∂x f[∂x U]}; geneqe // LTF

ut - fux ux,x == 0
```

The symmetries of this general non-linear diffusion equation are

```
symgeneqe = Infinitesimals[geneqe, {u}, {x, t}]

{xi[1] → Function[{x, t, u}, k2 + k4 x/2],
 xi[2] → Function[{x, t, u}, k3 + k4 t],
 phi[1] → Function[{x, t, u}, k1 + k4 u/2]}
```

The four group parameters *k1*, *k2*, *k3*, and *k4* represent the translation and scaling symmetries of the equation. If we further assume that f is given by a power of the derivatives u_x, we get the equation

```
geneqe = {∂t U - ∂x (∂x U)^μ}; geneqe // LTF

ut - μ ux^(-1+μ) ux,x == 0
```

where μ is a real parameter. The infinitesimals of this non-linear heat equation read

```
symgeneqe = Infinitesimals[geneqe, {u}, {x, t}, {μ}]

{xi[1] → Function[{x, t, u}, k4 + k5 x],
 xi[2] → Function[{x, t, u}, k1 + t (-k3 (-1 + μ) + k5 (1 + μ))],
 phi[1] → Function[{x, t, u}, k2 + k3 u]}
```

The five-dimensional group contains translations and scaling symmetries. In addition the symmetry group depends on the parameter μ.

The function Infinitesimals[] is also available in a shorthand operator notation. The related operator template is $\mathcal{PS}^{\gamma}_{u,x}[\Delta]$. As input quantities, this operator needs the

independent and dependent variables, the equations, and the parameters. The example of the Harry-Dym equation reduces to

```
PS[{λ}, {{u}, {x, t}}] [harryDym]

{xi[1] → Function[{x, t, u}, k3 + x (k4 + k5 x)],
 xi[2] → Function[{x, t, u}, k1 + k2 t],
 phi[1] → Function[{x, t, u}, u (-k2/3 + k4 + 2 k5 x)]}
```

5.4.4 Data Basis of Symmetries

The package *MathLie* offers a few functions allowing the creation of a database for differential equations. The database consists of different files containing information on the specific equation. Each file stores information on the equation itself and results gained by the application of different functions. The merits of such a database are the consecutive collection of information on symmetries, on algebraic properties, on transformation properties, and on solutions for the equation under consideration. Since the information is stored on disk, it can always be retrieved from there.

The basic element of the database is a file containing information on each individual equation. Such a file is created by the function LieEquations[]. The file contains information on the independent and dependent variables of the equation and the equation itself. It also contains information on possible parameters of the equation. Also included in the file are the expressions for which the equations are solved and applied as side conditions in the determining equations. After the solution of the determining equations, the file contains information on the infinitesimals. Sometimes, it is helpful to have the source or name of an equation available. This information is contained in two different global variables called *Title* and *Source*. The entire information needed to carry out a symmetry analysis can be created by exerting the function LieEquations[]. An example will show us how to facilitate this function.

Example 1

Let us again discuss the heat equation. The first step in adding information to the database is the creation of the related file. For the heat equation, we create the file *heat.dgl*. The function LieEquations[] will do the necessary job:

```
LieEquations["heat.dgl", {∂t u[x, t] - d ∂{x,2} u[x, t]},
 {u}, {x, t}, {d}, {"Heat equation"},
 {{"G. Baumann"}, {"Ulm 1997"}}]
```

The file with the name *heat.dgl* is created by this function in the current directory. The file contains the equation, which is given as second argument in a list. The third and fourth arguments of LieEquations[] are lists containing the dependent and independent variables. The fifth argument is a list containing the parameters of the equation—in our example the diffusion constant *d*. The sixth and seventh arguments of LieEquations[] are lists containing strings. The sixth list carries the name of the equation, here *Heat equation,* and the seventh list consists of sublists specifying for example the source of the equation. The file created by LieEquations[] contains this information and more. We can print the contents of the file by

```
!! heat.dgl

Title = {"Heat equation"}
Source = {{"G. Baumann"}, {"Ulm 1997"}}
IndepVar = {x, t}
DependVar = {u}
EqList = {Derivative[0, 1][u][x, t] -
          d*Derivative[2, 0][u][x, t]}
SubsList = {Derivative[0, 1][u][x, t]}
ParameterS = {d}
ListXi = {}
ListPhi = {}
```

The above printout shows that the file contains variables like *Title*, *Source*, *IndepVar*, *DependVar*, *EqList*, *SubsList*, *ParameterS*, *ListXi*, and *ListPhi*. These variables are global variables in *MathLie* and are used extensively in the package. Be sure not to use these names in your calculations. If you set up this file, it provides you with the information needed in symmetry calculations. For example, if you have to derive the determining equations for the heat equation, you can use the function Lie[] of the package *MathLie*. The function Lie[] delivers in its simpler applications, results similar to those obtained by the program of Champagne et al. [1991].

The function Lie[] needs one argument and a set of options. The options of Lie[] influence the properties of this function.

Let us calculate the determining equations for the heat equation. We start the calculation simply by

```
detheat = Lie["heat.dgl"]; detheat // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$

$$-(\xi_1)_t + d\,(\xi_1)_{x,x} - 2\,d\,(\phi_1)_{x,u} == 0$$
$$(\phi_1)_t - d\,(\phi_1)_{x,x} == 0$$
$$-2\,(\xi_1)_x + (\xi_2)_t == 0$$

Function Lie[] calculates the prolongation of the equation using the Fréchet derivative. After extraction of coefficients and a simplification of these equations, the function returns the determining equations collected in a list. This list can be used to manipulate the equations. For example, you can manually solve them as discussed in Section 5.4.3. As mentioned above, the behavior of the function Lie[] is controlled by different options. One of these options is called ScreenPrint which is set to False by default and thus suppresses all printing. If we want to see how the calculation proceeds, we can set the option TraceSteps→True, which is another option of Lie[]. These two options are helpful in checking the calculation if one is curious about the steps operated by Lie[]. The options also help in locating some errors that occurred during the calculation. The silent calculation done for the heat equation then looks like

```
Lie["heat.dgl", TraceSteps → True, ScreenPrint → True]

 +-------------------------------------------+
  Welcome to MathLie™
  for the calculation of the symmetry group
  by G. Baumann, © 1992 - 1999
 +-------------------------------------------+
Loading previous calculations
   of the determining equations from  HEAT.DEQ
Loading data from  HEAT.DGL

Title of the equations :
-----------------------------------------------------
Heat equation
Source of the equations:
G. Baumann
Ulm 1997
Equations of motion:
-----------------------------------------------------
Equation No. 1
u_t - d u_x,x == 0
-----------------------------------------------------
Substitution No. 1
u_t → d u_x,x
-----------------------------------------------------
We are using all the equation(s) checking the infinitesimals
of the given system consisting of 1 equation(s) in total.
```

```
We find 7 determining equations after simplification.
A list of the determining equations follows.
Equation No. : 1
-------------------------------------------------------
(xi[1])_u == 0
-------------------------------------------------------
Equation No. : 2
-------------------------------------------------------
(xi[2])_u == 0
-------------------------------------------------------
Equation No. : 3
-------------------------------------------------------
(phi[1])_u,u == 0
-------------------------------------------------------
Equation No. : 4
-------------------------------------------------------
(xi[2])_x == 0
-------------------------------------------------------
Equation No. : 5
-------------------------------------------------------
-2 d (phi[1])_x,u - (xi[1])_t + d (xi[1])_x,x == 0
-------------------------------------------------------
Equation No. : 6
-------------------------------------------------------
(phi[1])_t - d (phi[1])_x,x == 0
-------------------------------------------------------
Equation No. : 7
-------------------------------------------------------
-2 (xi[1])_x + (xi[2])_t == 0
-------------------------------------------------------
We
  have to treat 7 determining equations after simplification.

Results are collected in the list FinalResult.

{xi[1]^(0,0,1)[x, t, u],
 xi[2]^(0,0,1)[x, t, u[x, t]], phi[1]^(0,0,2)[x, t, u[x, t]],
 xi[2]^(1,0,0)[x, t, u[x, t]], -xi[1]^(0,1,0)[x, t, u[x, t]] -
  2 d phi[1]^(1,0,1)[x, t, u[x, t]] + d xi[1]^(2,0,0)[x, t, u[x, t]],
 phi[1]^(0,1,0)[x, t, u[x, t]] - d phi[1]^(2,0,0)[x, t, u[x, t]],
 xi[2]^(0,1,0)[x, t, u[x, t]] - 2 xi[1]^(1,0,0)[x, t, u[x, t]]}
```

The results of the calculation are again the determining equations. If you look at the beginning of the calculation, you will realize that Lie[] opens a file called *heat.deq* containing the determining equations. This file was created by Lie[] at the end of the

first run for the specific equation. The file contains the complete set of determining equations, thus keeping information on one equation in two different files. The first file with extension *.dgl* contains the essential information on the equation itself. The second file with extension *.deq* is used as source of derived information for the determining equations. Whenever you call Lie[] or derivatives of the function Lie[] be aware that both files contain information on the same equation. □

Let us now solve the determining equations. This can be done by using the function LieSolve[]. This function allows the solution of the determining equations by exerting the same procedures as the function Infinitesimals[]. The difference is that the information gained is saved in the file currently opened. In the case of the heat equation, we get

```
solheat = LieSolve["heat.dgl"]
```

$$\left\{\left\{\text{xi}[1] \to \text{Function}[\{x, t, u\}, k5 - 2\,d\,k2\,t + k6\,x + k4\,t\,x],\ \text{xi}[2] \to \text{Function}[\{x, t, u\}, k3 + t\,(2\,k6 + k4\,t)],\ \text{phi}[1] \to \text{Function}\left[\{x, t, u\}, u\left(k1 - \frac{k4\,t}{2} + k2\,x - \frac{k4\,x^2}{4\,d}\right) + \text{free}[1][x, t]\right]\right\},\right.$$
$$\left.\left\{-\frac{\text{free}[1]^{(0,1)}[x, t]}{d} + \text{free}[1]^{(2,0)}[x, t]\right\}\right\}$$

The result of this calculation is a list containing the infinitesimals and the remaining equations. The remaining equations are usually not solvable by the procedures used by LieSolve[]. If we look at the first part of this list, we realize that the infinitesimals *xi[1]*, *xi[2]*, and *phi[1]* for the heat equation are given in a pure function representation. This representation allows us to use the results for the infinitesimals in further calculations. The second part of the resulting list contains the unsolved equations, which in the present case is given by the original equation. We have to expect that the arbitrary function *free[1]* must satisfy the heat equation since we analyzed a linear partial differential equation. As noted, the function LieSolve[] saves the derived infinitesimals in the file *heat.dgl*. The information on the structure of the infinitesimals is contained in the lists *ListXi* and *ListPhi*. We can check the contents of *heat.dgl* again by

```
!! heat.dgl

Title = {Heat equation}
Source = {{G. Baumann}, {Ulm 1997}}
IndepVar = {x, t}
DependVar = {u}
EqList =
 {Derivative[0, 1][u][x, t] - d*Derivative[2, 0][u][x, t]}
SubsList = {Derivative[0, 1][u][x, t]}
ParameterS = {d}
```

```
ListXi = {}
ListPhi = {}
ListXi =
 {k5 - 2*d*k2*t + k6*x + k4*t*x, k3 + t*(2*k6 + k4*t)}
ListPhi = {(k1 - (k4*t)/2 + k2*x - (k4*x^2)/(4*d))*
    u[x, t] + free[1][x, t]}
ListEquations =
 {-(Derivative[0, 1][free[1]][x, t]/d) +
   Derivative[2, 0][free[1]][x, t]}
```

In addition to the two augmented lists *ListXi* and *ListPhi*, the file contains a new variable *ListEquations* storing the unsolved equations. After the application of Lie[] and LieSolve[] to the heat equation, we created an information basis on the symmetries of the equation. The symmetries of the heat equation are presented by constants *k1* to *k6* and by the arbitrary function *free[1]*. We can independently check the gained information to verify the invariance of the heat equation. Again, we apply function Lie[] in connection with the option *Info→True*. The application of Lie[] to *heat.dgl* in connection with the option *Info→True* gives us

```
Lie["heat.dgl", Info → True] // LTF
```

$$(\mathcal{F}_1)_t - d\,(\mathcal{F}_1)_{x,x} == 0$$

The function Lie[] takes the information on the infinitesimals and carries out the calculation of a symmetry analysis. In the last step of the calculation, the infinitesimals are inserted into the determining equations. The result shows that the original equation must be satisfied by the arbitrary function $\mathcal{F}_1 = free[1]$. It is essential for this example that the check reproduces the heat equation; otherwise, there would exist an error in the calculation. We note that for other equations the result may be an empty list, especially if the equation is a non-linear one. We also note that the check of the results is completely independent of the solution procedure used by LieSolve[]. Thus, we have an independent tool allowing us to examine the results of LieSolve[].

Example 2

Let us demonstrate the whole procedure of deriving the symmetries for another example. Closely related to the heat equation is the so-called non-linear filtration equation (Ibragimov [1994]). The equation of motion is given for a field $u = u(x, t)$ by

$$u_t - k(u_x)\,u_{x,x} = 0, \tag{5.23}$$

where $k(u_x)$ is an arbitrary function of u_x. In *Mathematica*'s notation, the left-hand side of the equation reads

```
filtration = {∂t u[x, t] - k[∂x u[x, t]] ∂{x,2} u[x, t]};
filtration // LTF
```

$$u_t - k\,u_{x,x} == 0$$

We suppressed the right-hand side of the equation and collected the left-hand side in a list. The file containing the information on this equation is created by

```
LieEquations["filtra.dgl", filtration, {u}, {x, t}, {},
 {"Nonlinear Filtration Equation"},
 {{"I.Sh. Akhatov, R.K. Gazizov and N.H. Ibragimov"},
  {"Group classification of nonlinear filtration equations"},
  {"Soviet Math. Dokl. 35, 384, 1987"},
  {"The equation is listed in the CRC Handbook
    of Lie Group Analysis of Differential Equations"},
  {"Vol. 1, Chapter 10.3, p. 129"},
  {"Ed. N.H. Ibragimov"},
  {"Boca Raton 1994"}}]
```

The symmetries of this equation follow by

```
solfilt = LieSolve["filtra.dgl"]
```

$$\left\{\left\{\text{xi}[1] \to \text{Function}\left[\{x, t, u\}, k2 + \frac{k4\,x}{2}\right],\right.\right.$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u\}, k3 + k4\,t],$$
$$\left.\left.\text{phi}[1] \to \text{Function}\left[\{x, t, u\}, k1 + \frac{k4\,u}{2}\right]\right\},\right.$$
$$\left.\{\}\right\}$$

The result is isomorphic to the result given by Ibragimov [1994] representing translations and scaling. Since the filtration equation is a non-linear PDE, we actually do not expect that arbitrary functions occur in the infinitesimals. However, there exists quite a number of non-linear examples for which the symmetry group is of infinite order and thus contains arbitrary functions. These functions are usually restricted by one or more PDEs which are different from the original equation.

If we know the symmetries, we can use another function of *MathLie,* called LieStructureForm[], to derive the algebraic properties of the related Lie algebra. We get the structural properties of the algebra and the transformation properties by

```
LieStructureForm["filtra.dgl"]
```

```
We will check the infinitesimals and calculate algebraic
as well as group properties
 +-------------------------------------------+
  Welcome to MathLie™
  for the calculation of the symmetry group
  by G . Baumann, © 1992 - 1999
```

```
 +-----------------------------------------------+
Loading
   previous calculations of the determining equations from
 FILTRA.DEQ
Loading data from  FILTRA.DGL
Title of the equations:
------------------------------------------------------------
Nonlinear Filtration Equation
Source of the equations:
------------------------------------------------------------
I.Sh. Akhatov, R.K. Gazizov and N.H. Ibragimov
Group classificatin of nonlinear filtration equations
Soviet Math. Dokl. 35, 384, 1987
The equation is listed in the CRC Handbook
  of Lie Group Analysis of Differential Equations
Vol. 1, Chapter 10.3, p. 129
Ed. N.H. Ibragimov
Boca Raton 1994
Equations of motion:
------------------------------------------------------------
Equation No. 1
u_t - k[u_x] u_x,x == 0
------------------------------------------------------------
Substitution No. 1
u_t → k[u_x] u_x,x
------------------------------------------------------------
Infinitesimals :
  Xi1 = k2 + (k4 x)/2
  Xi2 = k3 + k4 t
  Phi1 = k1 + 1/2 k4 u[x, t]
------------------------------------------------------------
We are using all the equation(s) checking the infinitesimals
of the given system consisting of 1 equation(s) in total.
We have to treat 0
  determining equations after simplification.
Calculation of the commutator table of the Lie algebra.
Basis of the Lie algebra.
------------------------------------------------------------
v1 == ∂_u
v2 == ∂_x
v3 == ∂_t
v4 == t ∂_t + (u ∂_u)/2 + (x ∂_x)/2
------------------------------------------------------------
```

Structure of the Lie algebra.

Ideal 0 : $\{v_1, v_2, v_3, v_4\}$
Ideal 1 : $\{v_1, v_2, v_3\}$
Ideal 2 : {solvable algebra}

Casimier(s) of the Lie algebra.

{}

Commutator table of the Lie algebra.

$$\begin{pmatrix} 0 & 0 & 0 & -\frac{v_1}{2} \\ 0 & 0 & 0 & -\frac{v_2}{2} \\ 0 & 0 & 0 & -v_3 \\ \frac{v_1}{2} & \frac{v_2}{2} & v_3 & 0 \end{pmatrix}$$

The commutator table is stored in list LieTable.
The nonzero structure constants are

$$C_{1,4,1} = -\frac{1}{2}$$

$$C_{2,4,2} = -\frac{1}{2}$$

$$C_{3,4,3} = -1$$

$$C_{4,1,1} = \frac{1}{2}$$

$$C_{4,2,2} = \frac{1}{2}$$

$$C_{4,3,3} = 1$$

Structure constants are contained in the list LieStructure.
Metric of the structure constants

$$g\_\{ij\} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{2} \end{pmatrix}$$

Determinant of the metric
det(g) = 0

Calculate the symmetry groups of the transformations

$$Xi(1) = k2 + \frac{k4\,x}{2}$$

$$Xi(2) = k3 + k4\,t$$

$$Phi(1) = k1 + \frac{1}{2}\,k4\,u[x, t]$$

```
-----------------------------------------------------
Group order = 4
Number of classes = 15
-----------------------------------------------------
The subgroups and the related transformations read
-----------------------------------------------------
{k1 → 0, k2 → 0, k3 → 0, k4 → 1}
x[s] == {x[s], t[s], u[s]}
t[s] == {x[s] → (E^(s/2) C[3]&), t[s] → (E^s C[1]&), u[s] → (E^(s/2) C[2]&)}
u[s] == {{}}
-----------------------------------------------------
{k1 → 0, k2 → 0, k3 → 1, k4 → 0}
x[s] == {x[s], t[s], u[s]}
t[s] == {x[s] → (C[3]&), t[s] → (C[1] + s&), u[s] → (C[2]&)}
u[s] == {{}}
-----------------------------------------------------
{k1 → 0, k2 → 0, k3 → 1, k4 → 1}
x[s] == {x[s], t[s], u[s]}
t[s] ==
 {x[s] → (E^(s/2) C[3]&), t[s] → (-1 + E^s C[1]&), u[s] → (E^(s/2) C[2]&)}
u[s] == {{}}
-----------------------------------------------------
 To save space the rest of the output was deleted .
-----------------------------------------------------
There exist no transformations creating new solutions.
Results are collected in the list FinalResult.
The infinitesimals are contained in Result2.
{}
```

The function actually does not return any result for further use. However, the function prints out the result shown on the screen. As a first result, we get the algebraic properties based on the vector fields of the symmetry group. The commutator table of the vector fields is followed by a number of group transformations based on subgroups of the symmetry. The subgroup is specified by replacement rules setting the values of the group parameters k_i. Transformations for the dependent and independent variables are given in a most general form. If the classification of the group allows a transformation of known solutions to other solutions of the equation, a graphical representation of this transformation is created. □

Example 3

One of the frequently used partial differential equations in the description of non-linear phenomena in physics is the Burgers equation. Writing down the equation in *Mathematica*, let us first define the field u by

```
U = u[x, t];
```

The left-hand side of the Burgers equation is given by

```
burgers = {∂t U + 2 U ∂x U - ∂{x,2} U}; burgers // LTF
```

$$u_t + 2\,u\,u_x - u_{x,x} == 0$$

The physical background and the important solutions are discussed by Lighthill [1956] and Crighton [1979].

The well-known Cole-Hopf transformation converts the Burgers equation to the linear diffusion equation. Hence, the solution for the former can be explicitly obtained by solving the linear problem of the diffusion equation. Our intention here is to find the symmetries of the Burgers equation for the standard representation given above.

Derivation of the symmetries as a part of our database system presumes that first we have to create the information file for the Burgers equation. The representation of the Burgers equation given above was taken from

```
source = {{"M.J. Ablowitz and P.A. Clarkson"},
  {"Solitons,
     Nonlinear Evolution Equations and Inverse Scattering"},
  {"Cambridge University Press, 1991"},
  {"first equation on page 34"}}
```

```
{{M.J. Ablowitz and P.A. Clarkson}, {Solitons, Nonlinear
    Evolution Equations and Inverse Scattering},
 {Cambridge University Press, 1991},
 {first equation on page 34}}
```

We set the title of the equation to

```
title = {"Burgers equation"}
```

```
{Burgers equation}
```

The file *burgers.dgl* is then created by

```
LieEquations["burgers.dgl", burgers, {u}, {x, t}, {},
 title, source]
```

The derivation of the determining equations is pursued by

```
burgersDetEquations = Lie["burgers.dgl", Statistics → True];
burgersDetEquations // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\phi_1)_{u,u} == 0$$
$$(\xi_2)_x == 0$$
$$2\,\phi_1 - (\xi_1)_t - 2\,u\,(\xi_1)_x + 2\,u\,(\xi_2)_t + (\xi_1)_{x,x} - 2\,(\phi_1)_{x,u} == 0$$
$$(\phi_1)_t + 2\,u\,(\phi_1)_x - (\phi_1)_{x,x} == 0$$
$$2\,(\xi_1)_x - (\xi_2)_t == 0$$

Again, the result of this calculation is a system of linear coupled partial differential equations. These equations are the basis for the determination of the symmetries. Before we attempt to solve these equations, we discuss another representation of the Burgers equation which follows from the above if we replace the dependent variable u by the gradient of a field v. We define this substitution by

```
subst = u → Function[{x, t}, ∂x v[x, t]]
```

$$u \to \text{Function}[\{x, t\}, \partial_x v[x, t]]$$

and apply this transformation to the Burgers equation

```
burgersx = burgers /. subst; burgersx // LTF
```

$$v_{x,t} + 2\,v_x\,v_{x,x} - v_{x,x,x} == 0$$

The result is a third-order non-linear PDE for the field v. Integrating this equation with respect to x, we obtain a potential representation of the Burgers equation:

```
pburgers = ∫ burgersx[[1]] dx; pburgers // LTF
```

$$v_t + v_x^2 - v_{x,x} == 0$$

This sort of equation was examined by Olver [1986].

In the following, we will show how group properties of the potential representation differ from the original representation of the Burgers equation. Let us again extend our database by a new file containing the information on the potential Burgers equation. We create a file for the potential Burgers equation by

```
LieEquations["burgersp.dgl", pburgers, {v}, {x, t}, {},
 {"Burgers equation"}, {{"G. Baumann"},
  {"Burgers equation in potential form"}, {"Ulm 1997"}}]
```

The corresponding determining equations follow by

```
Lie["burgersp.dgl"] // LTF
```

$$(\xi_1)_v == 0$$
$$(\xi_2)_v == 0$$
$$(\xi_2)_x == 0$$
$$-(\xi_1)_t + 2(\phi_1)_x + (\xi_1)_{x,x} - 2(\phi_1)_{x,v} == 0$$
$$(\phi_1)_t - (\phi_1)_{x,x} == 0$$
$$2(\xi_1)_x - (\xi_2)_t == 0$$
$$-2(\xi_1)_x + (\xi_2)_t + (\phi_1)_v - (\phi_1)_{v,v} == 0$$

A comparison of the results shows that in both calculations, the number of equations are the same. However, the seven equations differ in their forms. Consequently, the structure of the symmetries is different. We can illustrate this by using the function LieSolve[] of the package *MathLie*. The function LieSolve[] provides us with the representation of the symmetries in infinitesimal form. The infinitesimals of the Burgers equation thus follow by

```
iburgers = LieSolve["burgers.dgl"]
```

$$\{\{\text{xi}[1] \to \text{Function}[\{x, t, u\}, k2 + k4\,t + (k3 + k5\,t)\,x],$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u\}, k1 + t\,(2\,k3 + k5\,t)],$$
$$\text{phi}[1] \to \text{Function}\left[\{x, t, u\}, \frac{1}{2}(k4 - 2\,(k3 + k5\,t)\,u + k5\,x)\right]\},$$
$$\{\}\}$$

The result is a list containing the infinitesimals in a pure function representation. The first two elements *xi[1]* and *xi[2]* of the list represent the infinitesimals for the independent variables x and t, respectively. The third element, *phi[1]*, is the infinitesimal for the dependent variable. Among the five-dimensional symmetry group are symmetries of translation and scaling. The infinitesimals for the potential Burgers equation follow by a similar calculation from

```
iburgersp = LieSolve["burgersp.dgl"]
```

$$\{\{\text{xi}[1] \to \text{Function}[\{x, t, v\}, k2 + 2\,k5\,t + (k3 + 4\,k6\,t)\,x],$$
$$\text{xi}[2] \to \text{Function}[\{x, t, v\}, k1 + 2\,t\,(k3 + 2\,k6\,t)],\ \text{phi}[1] \to$$
$$\text{Function}[\{x, t, v\}, k4 + 2\,k6\,t + k5\,x + k6\,x^2 + E^v\,\text{free}[1][x, t]]\},$$
$$\{-\text{free}[1]^{(0,1)}[x, t] + \text{free}[1]^{(2,0)}[x, t]\}\}$$

Comparing both results, we recognize that the symmetry structure of the original Burgers equation and the potential representation of the Burgers equation are the same for the infinitesimals of the independent variables. The structure of the dependent variables is completely different. In the case of the Burgers equation, we find a finite dimensional representation of the symmetry group, whereas in the potential representation, we get an infinite dimensional group characterized by the

arbitrary function *free[1][x,t]*. This arbitrary function has to satisfy the heat equation as an additional equation. The last list of the result contains this equation. The discrete part of the symmetry group of the potential Burgers equation is also extended by one degree to a six-dimensional symmetry group. □

In physical applications, however, the model equations often turn out to be more complicated than the Burgers equation, involving as they do, a geometrical expansion term, a non-linear damping term, or a more general convection term. In the following, we will examine the influence of this change on the symmetries if the convection term in the original equation is altered. The convective extension of the Burgers equation is obtained by replacing u_x with $u^2 u_x$. This quadratic extension is sometimes called *generalized Burgers equation.*

```
burgersc = {∂t U + U^2 ∂x U - ∂{x,2} U}; burgersc // LTF
```

$$u_t + u^2\, u_x - u_{x,x} == 0$$

For a quick overview of the changes in the symmetries, we take advantage of the interactive function Infinitesimals[]. Application of this function to the equation can also be used to create a database if the related notebook is considered the basic part of the data system. The advantage of this view is that the complete information on an equation is collected in one file and that background information which is not necessary for a symmetry analysis can also be collected in the notebook. The determination of the symmetries for this generalized equation is carried out by

```
iburgersc = Infinitesimals[burgersc, {u}, {x, t}]
```

$$\left\{\text{xi}[1] \to \text{Function}\left[\{x, t, u\}, k1 + \frac{k3\, x}{2}\right],\right.$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u\}, k2 + k3\, t],$$
$$\left.\text{phi}[1] \to \text{Function}\left[\{x, t, u\}, -\frac{k3\, u}{4}\right]\right\}$$

The result shows that the symmetry group of the generalized Burgers equation is reduced from a six-dimensional to a three-dimensional group containing only translations in the independent variables and a scaling.

We recognize now that classification of the group of differential equations is an easy task. For example, the group structure of the Burgers equation with generalized convection term $u^m u_x$ is analyzed by

```
Infinitesimals[{∂t U + U^m ∂x U - ∂{x,2} U}, {u}, {x, t}, {m}]
```

$$\left\{\text{xi}[1] \to \text{Function}\left[\{x, t, u\}, k1 + \frac{k3\, x}{2}\right],\right.$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u\}, k2 + k3\, t],$$
$$\left.\text{phi}[1] \to \text{Function}\left[\{x, t, u\}, -\frac{k3\, u}{2\, m}\right]\right\}$$

where m is an arbitrary constant. The result demonstrates that for an arbitrary m, the symmetry group is of the same structure as for the generalized Burgers equation with quadratic convection term. The difference between the two results is the dependence of *phi[1]* on the parameter m. This result displays that the function Infinitesimals[] is capable of handling not only the specific models of the Burgers equation but also a model depending on an arbitrary parameter.

Concluding this section, we can state that *MathLie* can be used to collect information on symmetries in different ways: first, creation of a file system containing information essential for each equation; second, by creating notebooks that allow detailed discussion of the equation. Both methods will be used in the remaining parts of this book.

5.5. Similarity Reduction of Partial Differential Equations

A similarity reduction of a differential equation is closely connected with the invariance of the equation. In Chapter 3, we discussed the group invariance of ordinary differential equations. The invariance condition in connection with ordinary differential equations was beneficial for the reduction of the order or the integration of ODEs by quadratures. In this section, we will discuss the invariance condition of a partial differential equation to reduce the equation to an ODE or to a PDE in less independent coordinates. The first case occurs for an equation with two independent variables, whereas a general reduction to another PDE follows in cases with more than two independent variables. The reduction procedure generates a similarity representation of the original equation. Since we reduce the number of independent variables, a similarity reduction results into a representation which has some advantages compared with the original equation.

This procedure works independent of the nature of the PDE, linear or nonlinear. It is also independent of the order of the PDE and it does not need information on boundary conditions. The benefit of the reduction is a simpler equation allowing an analytic or numeric solution. Examples of similarity solutions are widely encountered in mechanics, hydrodynamics, in the general theory of relativity, etc. However, the solutions derived in these fields are mainly guesswork and lack an algorithmic procedure. We are going to show here that a similarity reduction is an algorithmic procedure delivering a large number of ansätze discussed in the literature.

The main idea behind a similarity reduction is the term invariance. The discussion of invariance is the first step in this section. To simplify the theoretical representation, we restrict the examination to cases with two independent variables. A generalization to more independent variables is straightforward.

Let us consider the general PDE in two independent variables and one dependent variable in the form

$$\Delta(x, t, u, u_x, u_t, u_{x,x}, \ldots) = 0, \tag{5.24}$$

which is invariant under the one-parameter Lie group of transformations

$$x^* = X(x, t, u; \epsilon), \tag{5.25}$$

$$t^* = T(x, t, u; \epsilon), \tag{5.26}$$

$$u^* = U(x, t, u; \epsilon). \tag{5.27}$$

Let us further assume that $u = \Theta(x, t)$ is a solution of equation (5.24). Inserting this solution into the transformations (5.25)–(5.27), we can write

$$x^* = X(x, t, \theta(x, t); \epsilon), \tag{5.28}$$

$$t^* = T(x, t, \theta(x, t); \epsilon), \tag{5.29}$$

$$u^* = U(x, t, \theta(x, t); \epsilon), \tag{5.30}$$

stating that u^* is also a solution of the transformed PDE. The use of these facts allows us to define the invariance in the following way.

Definition: Invariance

A PDE is invariant under a one-parameter Lie group transformation if $u^* = U(x, t, \Theta(x, t); \epsilon)$ also satisfies the transformed PDE whenever $u = \Theta(x, t)$ is a solution of the original PDE. ○

If we additionally require that we can solve this problem uniquely, we end up with the functional equation

$$\Theta(X(x, t, \Theta(x, t); \epsilon), T(x, t, \theta(x, t); \epsilon)) = U(x, t, \Theta(x, t); \epsilon). \tag{5.31}$$

The solution of this functional equation can be found by introducing the infinitesimal representation of the transformations (5.28)–(5.30). We will carry out the calculation by replacing the transformations (5.28)–(5.30) by their infinitesimal representations

```
Clear[T, X, U];
itrafo = {X → Function[{x, t, u, ϵ}, x + ϵ xi[1][x, t, u]],
  T → Function[{x, t, u, ϵ}, t + ϵ xi[2][x, t, u]],
  U → Function[{x, t, u, ϵ}, u + ϵ phi[1][x, t, u]]}
```

```
{X → Function[{x, t, u, ϵ}, x + ϵ xi[1][x, t, u]],
 T → Function[{x, t, u, ϵ}, t + ϵ xi[2][x, t, u]],
 U → Function[{x, t, u, ϵ}, u + ϵ phi[1][x, t, u]]}
```

Inserting the infinitesimal representation of the transformations into the functional equation (5.31), we get the simplification

```
fun = Θ[X[x, t, Θ, ϵ], T[x, t, Θ, ϵ]] == U[x, t, Θ, ϵ] /. itrafo

Θ[x + ϵ xi[1][x, t, Θ], t + ϵ xi[2][x, t, Θ]] == Θ + ϵ phi[1][x, t, Θ]
```

The parameter ϵ is the group parameter of the transformation. On the left-hand side this expression contains the solution θ for which we are looking. Actually, the above expressions contain θ in a functional form, too. However, this representation depends on the infinitesimals parameter ϵ allowing us a Taylor expansion around the identity $\epsilon = 0$. If we additionally subtract the right-hand side from the left-hand side and extract the terms of lowest order in ϵ, we get the result

```
Coefficient[
   (Series[fun[[1]], {ϵ, 0, 1}] /. Θ[x, t] → Θ) - fun[[2]], ϵ] == 0 //
 LieTraditionalForm

Θx ξ1 + Θt ξ2 - ϕ1 == 0
```

This first-order PDE is called the invariant surface condition. The problem we face is the solution of this first-order partial differential equation. Actually, this is not a problem if we use *Mathematica*. *Mathematica* offers a package integrated in the function DSolve[] allowing the integration of first-order PDEs. This package uses the fact that a first-order PDE is closely related to a set of ordinary differential equations. The connection between the first-order PDE and the system of ODEs can be demonstrated by the following reasoning.

Let us assume that an arbitrary surface in the space with coordinates x, t, and u is given by

$$u = \Theta(x, t). \tag{5.32}$$

Later, we will identify this surface as the solution surface of the first-order PDE. A curve embedded in this surface can be described by a set of parametric coordinates. Let s be the parameter along the curve; then the curve itself is given by the triple

$$(x = x(s), t = t(s), u = u(s)). \tag{5.33}$$

The tangent vector to this curve is

$$\vec{v} = \vec{e}_x \frac{dx}{ds} + \vec{e}_t \frac{dt}{ds} + \vec{e}_u \frac{du}{ds}, \tag{5.34}$$

where $\vec{e}_x$, $\vec{e}_t$, and $\vec{e}_u$ are the unit vectors in the directions of the coordinate axes. The normal direction on the surface $H = u - \Theta(x, t) = \text{const}.$ is determined by the gradient divided by the magnitude of the gradient

$$\vec{n} = \frac{\nabla H}{|\nabla H|} = \frac{1}{|\nabla H|}(-\vec{e}_x \Theta_x - \vec{e}_t \Theta_t + \vec{e}_u). \tag{5.35}$$

The condition $\vec{n}\,\vec{v} = 0$ assures that the curve $(x(s), t(s), u(s))$ is part of the surface H. This condition, however, is nothing more than

$$\Theta_x \frac{dx}{ds} + \Theta_t \frac{dt}{ds} = \frac{du}{ds}. \tag{5.36}$$

If we define a family of curves by

$$\frac{dx}{ds} = \xi_1(x, t, u), \tag{5.37}$$

$$\frac{dt}{ds} = \xi_2(x, t, u), \tag{5.38}$$

$$\frac{du}{ds} = \phi_1(x, t, u), \tag{5.39}$$

called the characteristic differential equations, we can rewrite the surface condition $\vec{n}\,\vec{v} = 0$ in a partial differential equation of first order:

$$\xi_1 \Theta_x + \xi_2 \Theta_t = \phi_1, \tag{5.40}$$

which is equivalent with the equation derived from the functional relation (5.31) for Θ. Consequently, each one-parameter family of characteristic curves generates a surface which defines an integral surface $u - \Theta(x, t) = \text{const}.$ and each such integral surface is generated by a one-parameter family of characteristic curves.

Example 1

Let us consider the first-order partial differential equation

```
pde1 = x ∂x Θ[x, t] + t ∂t Θ[x, t] == Θ[x, t];
pde1 // LieTraditionalForm
```

$$t\,\Theta_t + x\,\Theta_x == \Theta$$

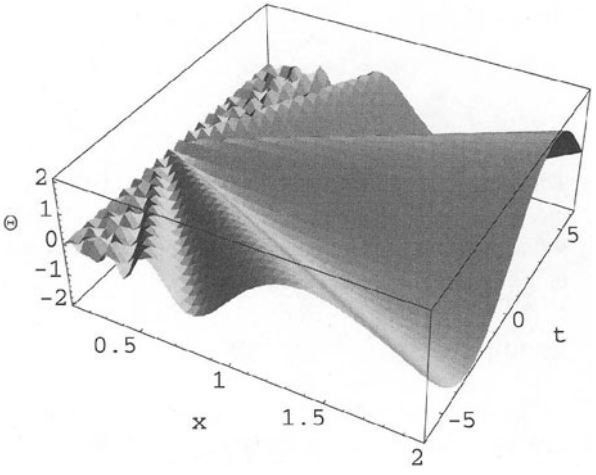
corresponding to a scaling symmetry with infinitesimals $\xi_1 = x$ and $\xi_2 = t$. The first-order PDE is solved by

```
sol = DSolve[pde1, Θ[x, t], {x, t}]
```

$$\left\{\left\{\Theta[x, t] \to x\, C[1]\left[\frac{t}{x}\right]\right\}\right\}$$

where *C[1][t/x]* is an arbitrary function. A graphical representation of this solution with *C[1]* replaced by Sin[] is given by

```
Plot3D[(Θ[x, t] /. (sol /. C[1] → Sin))[[1]],
 {x, 0.1, 2}, {t, -2π, 2π},
 AxesLabel → {"x", "t", "Θ"},
 PlotPoints → 40, Mesh → False]
```



□

Actually, the family of solutions of the characteristic differential equations can be represented in a parametric form by

$$x = x(s, r), \quad t = t(s, r), \quad u = u(s, r), \tag{5.41}$$

where s is the parameter along a characteristic curve and r is the parameter identifying a characteristic curve equal to a certain constant on a characteristic. The essential point of these considerations is that the solution of a first-order partial differential equation is represented by a family of surfaces $u - \Theta(x, t) = \text{const}$.

Now, if $F(x, t, \Theta) = 0$ defines a surface satisfying the first-order partial differential equation, then this surface is an invariant of the one-parameter Lie group transformation. This is obvious from the condition

$$\hat{v}\, F(x, t, \Theta) = 0, \tag{5.42}$$

where $\vec{v}$ is the vector field of the transformation given by

$$\vec{v} = \xi_1 \partial_x + \xi_2 \partial_t + \phi_1 \partial_\Theta \,. \tag{5.43}$$

The result is that the vector field $\vec{v}$ applied on the surface $F(x, t, \Theta)$ delivers the determining equation for the surface. These facts are summarized in the following theorem.

Theorem: Invariance condition

The function $F(x, t, \Theta)$ is an invariant of a one-parameter Lie group transformation if the condition

$$\vec{v}\, F = 0 \tag{5.44}$$

is satisfied. ○

This condition always results into a first-order partial differential equation independent of the number of dependent and independent variables. The equation is solvable by applying the method of characteristics. We demonstrate the application of the theorem by an example.

Example 2

Let us apply the theorem for a subgroup of the heat equation. The symmetry we will examine is connected with the scaling symmetry

```
xi[1][x, t, u] = x;
xi[2][x, t, u] = 2 t;
phi[1][x, t, u] = c u;
```

where c is an arbitrary parameter. The invariance condition (5.44) now reads

```
invar = xi[1][x, t, u] ∂x F[x, t, u] +
  xi[2][x, t, u] ∂t F[x, t, u] +
  phi[1][x, t, u] ∂u F[x, t, u] == 0; invar // LTF
```

$$2\,t\,F_t + c\,u\,F_u + x\,F_x == 0$$

The solution of this PDE follows by

```
solh = DSolve[invar, F[x, t, u], {x, t, u}]
```

$$\left\{\left\{F[x,\ t,\ u] \to C[1]\left[\frac{t}{x^2},\ u\,x^{-c}\right]\right\}\right\}$$

representing the general solution of the first-order partial differential equation. The arbitrary function *C[1]* depends on two invariants given by $I_1 = t/x^2$ and $I_2 = u/x^c$. These two invariants allow the reduction of the heat equation to an ordinary differential equation. The reduction procedure itself is based on the following theorem:

Theorem: Invariant representation

Let the equation $\Delta = 0$ be invariant under a one-parameter group $\mathbf{G}$ and let the infinitesimals ξ_i and ϕ_α, $i = 1, 2, \ldots, p$ and $\alpha = 1, 2, \ldots, q$, be non-vanishing functions on the solution surface H of the equation. Then, the surface H can be represented by equations of the form

$$\Phi_k(I_1(x, u), \ldots, I_{p-1}) = 0, \qquad k = 1, 2, \ldots, q, \tag{5.45}$$

where $I_1, \ldots, I_{p-1}$ define a basis of invariants of the group $\mathbf{G}$. ○

The use of the invariants allows us to reduce the original equation. Let us demonstrate the reduction process by the example of the heat equation. The left-hand side of the equation of motion reads

```
heat = {∂t u[x, t] - ∂{x,2} u[x, t]}; heat // LTF
```

$$u_t - u_{x,x} == 0$$

The two integrals obtained by integrating the characteristic equation are $I_1 = t/x^2$ and $I_2 = u/x^c$. The first integral combines the independent variables in a unique variable called the similarity variable $\zeta = t/x^2$. The second invariant, I_2, combines the dependent variable and one independent variable to the similarity representation of the solution $u(x, t) = x^c F(\zeta)$, where $F(\zeta) = I_2$. These two relations allow us to define the following rules:

```
rules = {t → ζ x^2, u → Function[{x, t}, x^c F[t/x^2]]}
```

$$\left\{t \to x^2\, \zeta,\ u \to \text{Function}\left[\{x, t\},\ x^c\, F\left[\frac{t}{x^2}\right]\right]\right\}$$

Applying these two rules to the heat equation, we get

```
rheat = heat /. rules; rheat // LTF
```

$$-(-1 + c)\, c\, F\, x^{-2+c} + x^{-2+c}\, F_\zeta + 4\, c\, x^{-2+c}\, \zeta\, F_\zeta - x^c \left(\frac{6\, \zeta\, F_\zeta}{x^2} + \frac{4\, \zeta^2\, F_{\zeta,\zeta}}{x^2}\right) == 0$$

representing an ordinary differential equation for F depending only on ζ. The common factor x^{c-2} can be eliminated by division:

```
rheat = Expand[rheat / x^(c-2)]; rheat // LTF
```

$$c\,F - c^2\,F + F_\zeta - 6\,\zeta\,F_\zeta + 4\,c\,\zeta\,F_\zeta - 4\,\zeta^2\,F_{\zeta,\zeta} == 0$$

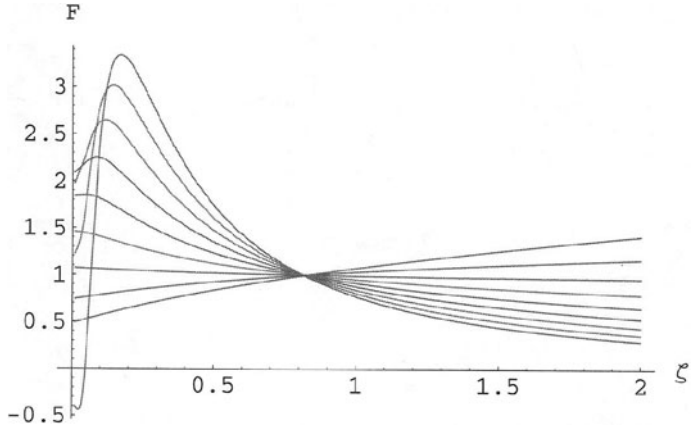
Thus, we reduced the original PDE to an ODE by utilizing the invariants of the group. The merit of this reduction is that the derived ODE is easier to solve than the original PDE. Another advantage is that we can use the solution procedures discussed in Section 4.4, allowing us to solve the reduced equation. However, in the present case, we utilize the capabilities of DSolve[]. The solution of the ODE follows by

```
sheat = DSolve[rheat[[1]] == 0, F, ζ]
```

$$\left\{\left\{F \to \left(C[1]\,\text{Hypergeometric1F1}\left[\frac{1}{2} - \frac{c}{2}, \frac{3}{2}, -\frac{1}{4\,\#1}\right]\left(\frac{1}{\#1}\right)^{\frac{1}{2}-\frac{c}{2}} + C[2]\,\text{Hypergeometric1F1}\left[-\frac{c}{2}, \frac{1}{2}, -\frac{1}{4\,\#1}\right]\left(\frac{1}{\#1}\right)^{-c/2}\&\right)\right\}\right\}$$

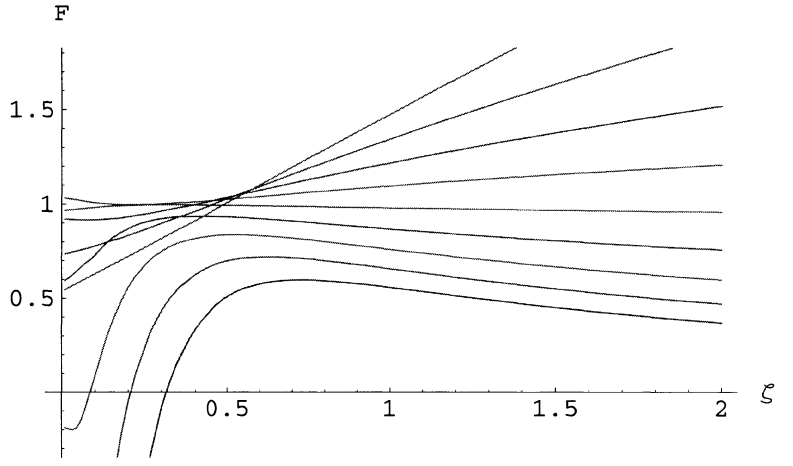
The resulting solution is a combination of special functions containing the group parameter c. Since we started from a second-order ordinary differential equation, we end up with a solution containing two constants of integration *C[1]* and *C[2]*. The two solutions are graphically shown in the following figure. The different curves represent different values of the group parameter c.

```
Plot[Evaluate[Table[F[ζ] /. sheat[[1, 1]],
   {c, -2.1, 2.1, .5}] /. {C[1] → 1, C[2] → 0}],
 {ζ, 0.01, 2}, AxesLabel → {"ζ", "F"},
 PlotStyle → Table[Hue[i/10], {i, 1, 8}]]
```



The second solution with $C[1] = 0$ and $C[2] = 1$ has the graph

```
Plot[Evaluate[Table[F[ζ] /. sheat[[1, 1]],
    {c, -2.1, 2.1, .5}] /. {C[1] → 0, C[2] → 1}],
 {ζ, 0.01, 2}, AxesLabel → {"ζ", "F"},
 PlotStyle → Table[Hue[i/10], {i, 1, 8}]]
```



The symbolic solution for the heat equation in the original coordinates x and t is derived from the similarity solution by inverting the transformations:

```
backrules = {ζ → t/x^2, F → Function[ζ, F[ζ] x^c]}
```

$$\left\{\zeta \to \frac{t}{x^2},\ F \to \text{Function}[\zeta,\ F[\zeta]\ x^c]\right\}$$

The actual solution follows then by the resubstitution of the similarity representation:

```
solution = F[ζ] /. sheat /. backrules
```

$$\left\{\left(\frac{x^2}{t}\right)^{\frac{1}{2}-\frac{c}{2}} C[1]\ \text{Hypergeometric1F1}\left[\frac{1}{2}-\frac{c}{2},\ \frac{3}{2},\ -\frac{x^2}{4t}\right] + \left(\frac{x^2}{t}\right)^{-c/2} C[2]\ \text{Hypergeometric1F1}\left[-\frac{c}{2},\ \frac{1}{2},\ -\frac{x^2}{4t}\right]\right\}$$

In conclusion, we can say that a solution of a partial differential equation in two independent variables can be constructed by two invariants of the group. One of these two invariants becomes the new independent variable $\zeta = \zeta(x, t)$, the so-called similarity variable, and the other invariant plays the role of a dependent variable $F(\zeta)$. The similarity representation of the solution is given by the relation

$$\Theta = H(x, t, F(\zeta)) \tag{5.46}$$

with the dependence of H on x and t and the arbitrary function $F(\zeta)$ known explicitly. The substitution of this similarity representation into the original equation results in an ordinary differential equation for $F(\zeta)$. □

Now we know the fundamental steps to reduce a PDE to an ODE if the symmetries of the PDE are given. The package *MathLie* offers a function called LieReduction[], which allows us to reduce the number of independent variables by applying the above considerations. The function LieReduction[] can be used to automatically derive the similarity representation of the heat equation and others. The information we need to set up the calculation are the equation, the dependent and independent variables, and two lists for the infinitesimals. The first list of the infinitesimals carries the information on the independent variables and the second carries the group structure of the dependent variables. The following example shows how the reduction for the scaling symmetry is derived for the heat equation:

```
red1 = LieReduction[{∂t u[x, t] - ∂{x,2} u[x, t]}, {u},
  {x, t}, {x, 2 t}, {c u}]; LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$\frac{t}{x^2} - \zeta_1 == 0$$

$$u\, x^{-c} - F_1 == 0$$

$$-x^c \left(-c\, F_1 + c^2\, F_1 - (F_1)_{\zeta_1} + 6\, \zeta_1\, (F_1)_{\zeta_1} - 4\, c\, \zeta_1\, (F_1)_{\zeta_1} + 4\, \zeta_1^2\, (F_1)_{\zeta_1,\zeta_1}\right) == 0$$

The output of the function is a list containing the similarity variables as equations at the first position. The second part of the list contains the similarity representation of the solution. The third and last part of the resulting list contains the reduction of the original equation. Another example for the application of this function is given by the potential Burgers equation. We already determined the symmetries of the potential Burgers equation in Section 5.4.4. Using the results from there for the special group with *k6=1* and *ki=0* for *i*≠6, we find

```
red1 = LieReduction[{∂t u[x, t] + (∂x u[x, t])^2 - ∂{x,2} u[x, t]},
  {u}, {x, t}, {4 t x, 4 t^2}, {2 t + x^2}];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$\frac{t}{x} - \zeta_1 == 0$$

$$\frac{1}{4}\left(4\, u - \frac{x^2}{t} - 2\, \mathrm{Log}[x]\right) - F_1 == 0$$

$$\zeta_1^2\, (3 - 12\, F1_{\zeta_1}\, \zeta_1 + 4\, F1_{\zeta_1}^2\, \zeta_1^2 - 4\, \zeta_1^2\, F1_{\zeta_1,\zeta_1}) == 0$$

The result of the reduction is a non-linear second-order ordinary differential equation which we can solve by using the function DSolve[]:

```
solburgersp = DSolve[red1[[3, 1]] == 0, F1, zeta1]
```

$$\left\{\left\{F1 \to \left(C[2] + \frac{3\,\text{Log}[\#1]}{2} - \text{Log}[-1 + \#1\,C[1]]\&\right)\right\}\right\}$$

The final solution of the potential Burgers equation follows by using the second part of the similarity reduction. Inserting the solution for *F1* into the similarity representation, we get a representation of the solution in x and t coordinates,

```
solution = Flatten[red1[[2]] /. solburgersp]
```

$$\left\{\frac{1}{4}\left(4\,u - \frac{x^2}{t} - 2\,\text{Log}[x]\right) == C[2] + \frac{3}{2}\,\text{Log}\left[\frac{t}{x}\right] - \text{Log}\left[-1 + \frac{t\,C[1]}{x}\right]\right\}$$

Solving this relation with respect to u, we end up with the explicit solution of the potential Burgers equation in the form

```
solf = Simplify[Solve[solution[[1]], u]]
```

$$\left\{\left\{u \to \frac{1}{4}\left(\frac{x^2}{t} + 4\,C[2] + 6\,\text{Log}\left[\frac{t}{x}\right] + 2\,\text{Log}[x] - 4\,\text{Log}\left[-1 + \frac{t\,C[1]}{x}\right]\right)\right\}\right\}$$

The solution of the chosen subgroup contains two constants of integration which have to be chosen in such a way that the initial and boundary conditions are satisfied.

Example 3

Another example of two coupled diffusion equations demonstrates the application of the function LieReduction[] on a system of equations. The (1+1)-dimensional version of the equations of motion for the density u and the density v are given by

```
cdiffu = {∂t u[x, t] - ∂x,x v[x, t],
  ∂t v[x, t] - ∂x,x u[x, t]}; cdiffu // LTF
```

$$u_t - v_{x,x} == 0$$
$$v_t - u_{x,x} == 0$$

The two equations allow a large symmetry group. The infinitesimals of this group follow by applying the function Infinitesimals[]. The result reads

```
icdiffu = Infinitesimals[cdiffu, {u, v},
  {x, t}, {}, SubstitutionRules → {∂t u[x, t], ∂t v[x, t]}];
icdiffu // LTF
```

$$-(\mathcal{F}_2)_t + (\mathcal{F}_1)_{x,x} == 0$$

$$-(\mathcal{F}_1)_t + (\mathcal{F}_2)_{x,x} == 0$$

$$\xi_1 == k6 + k7\,x - 2\,t\,(k3 + k5\,x)$$

$$\xi_2 == k1 + 2\,t\,(k7 - k5\,t)$$

$$\phi_1 == (k4 + k5\,t)\,u + v\left(k2 + k3\,x + \frac{k5\,x^2}{2}\right) + \mathcal{F}_1$$

$$\phi_2 == (k4 + k5\,t)\,v + u\left(k2 + k3\,x + \frac{k5\,x^2}{2}\right) + \mathcal{F}_2$$

The two functions *free[1]* and *free[2]* satisfy the original equations and generate the infinite dimensional part of the group. The finite part of the group is represented by a seven-dimensional symmetry group. From this group, we select the subgroup with *k4* = *k7* = *1* representing a scaling symmetry of the coupled equations:

```
xinfi = {xi[1][x, t, u, v], xi[2][x, t, u, v]} /.
  icdiffu[[1]] /.
  {k1 → 0, k2 → 0, k3 → 0, k4 → 1, k5 → 0, k6 → 0, k7 → 1}
```

```
{x, 2 t}
```

```
uinfi = {phi[1][x, t, u, v], phi[2][x, t, u, v]} /. icdiffu[[1]] /.
  {k1 → 0, k2 → 0, k3 → 0, k4 → 1, k5 → 0, k6 → 0, k7 → 1,
   free[1][x, t] → 0, free[2][x, t] → 0}
```

```
{u, v}
```

The reduction of the coupled diffusion equations for the scaling group follows then by

```
rcdiffu = LieReduction[cdiffu, {u, v}, {x, t}, {x, 2 t}, {u, v}];
LieTraditionalForm[rcdiffu] /. zeta1 → ζ1 // TableForm
```

$$\frac{t}{x^2} == \zeta_1$$

$$\frac{u}{x} == F_1 \qquad\qquad \frac{v}{x} == F_2$$

$$(F_1)_{\zeta_1} - 2\,\zeta_1\,(F_2)_{\zeta_1} - 4\,\zeta_1^2\,(F_2)_{\zeta_1,\zeta_1} == 0 \qquad -2\,\zeta_1\,(F_1)_{\zeta_1} + (F_2)_{\zeta_1} - 4\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1} == 0$$

The derived coupled set of ordinary differential equations in $F_1 = F1$ and $F_2 = F1$ is not solved by *Mathematica*. This is obvious from

```
DSolve[rcdiffu[[3]], {F1, F2}, zeta1]
```

```
DSolve[{F1'[zeta1] - 2 zeta1 F2'[zeta1] - 4 zeta1^2 F2''[zeta1] == 0,
  -2 zeta1 F1'[zeta1] + F2'[zeta1] - 4 zeta1^2 F1''[zeta1] == 0},
 {F1, F2}, zeta1]
```

However, this is not the end of the story. *Mathematica* offers an alternate way: the numerical solution of the equation. The function NDSolve[] is capable of handling this task. This function is beneficial in solving the reduced system of equations. In determining a numerical solution, it is mandatory that the equations be free of any parameters and that the initial conditions be added to the equations:

```
eqat = rcdiffu[[3]]
```

```
{F1'[zeta1] - 2 zeta1 F2'[zeta1] - 4 zeta1^2 F2''[zeta1] == 0,
 -2 zeta1 F1'[zeta1] + F2'[zeta1] - 4 zeta1^2 F1''[zeta1] == 0}
```

For the initial conditions, we choose the following relations:

```
initials =
 {F1[1] == .1, F2[1] == .2, F1'[1] == -.1, F2'[1] == -0.5};
```

setting the function values at *zeta1=1* and the derivatives to certain constants. These four equations are joined with the list of equations *eqat*:

```
eqat = Join[eqat, initials]
```

```
{F1'[zeta1] - 2 zeta1 F2'[zeta1] - 4 zeta1^2 F2''[zeta1] == 0,
 -2 zeta1 F1'[zeta1] + F2'[zeta1] - 4 zeta1^2 F1''[zeta1] == 0,
 F1[1] == 0.1, F2[1] == 0.2, F1'[1] == -0.1, F2'[1] == -0.5}
```

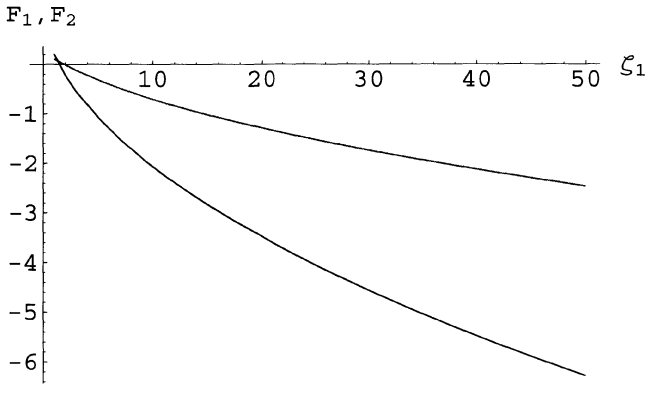
The complete list of equations and initial conditions is now used in the numerical integration for *zeta1* in the range $1 \leq zeta1 \leq 50$.

```
nsolve = NDSolve[eqat, {F1, F2}, {zeta1, 1, 50}]
```

```
{{F1 → InterpolatingFunction[{{1., 50.}}, <>],
  F2 → InterpolatingFunction[{{1., 50.}}, <>]}}
```

The numerical solution is represented by an interpolating function. We can use the representations to plot the solution:

```
Plot[Evaluate[{F1[x], F2[x]} /. nsolve], {x, 1, 50},
 PlotStyle → {RGBColor[1.000, 0.000, 0.000],
   RGBColor[0.000, 0.000, 1.000]},
 AxesLabel → {"ζ1", "F1,F2"}]
```



The utilization of the numerical capabilities of *Mathematica* allows us to examine the solution of the reduced equations for a limited range in *zeta1* and for a special choice of initial values. This numerical representation of the solution compared with an analytic solution is far from being complete. An analytic solution, if we found one, is valid for all initial conditions and unlimited in the range of the independent variable. So numerical solution can only show us the behavior for a specific case of initial conditions. □

Example 4

Another problem also handled by the function LieReduction[] is the reduction of partial differential equations in more than two independent variables. Such a case is given by the Karpman-Belashov equation (KB) (Karpman and Belashov [1991])

$$6u_x^2 + u_{x,t} + 6u\,u_{x,x} - u_{y,y} - \mu\,u_{x,x,x} - \epsilon\,u_{x,x,x,x} - \lambda\,u_{x,x,x,x,x,x} = 0. \tag{5.47}$$

The KB equation contains the Zabolotskaya-Khoklov equation (ZK) with $\epsilon = 0$ and $\lambda = 0$, and the Kadomtsev-Petviashvili equation (KP) with $\mu = 0$ and $\lambda = 0$. The KB-equation is used to model two-dimensional solitons and wave packages in weakly dispersive and dissipative media. Karpman and Belashov studied this type of equation numerically. We will examine here the algebraic properties of the equation. Especially, we will examine the ZK equation and the corresponding analytical solutions. First, let us determine the symmetries of the KB equation

```
karp = {∂x (∂t u[x, y, t] + 6 u[x, y, t] ∂x u[x, y, t] -
     μ ∂{x,2} u[x, y, t] - ϵ ∂{x,3} u[x, y, t] -
      λ ∂{x,5} u[x, y, t]) - ∂{y,2} u[x, y, t]}; karp // LTF
```

$$6\,u_x^2 + u_{x,t} + 6\,u\,u_{x,x} - u_{y,y} - \mu\,u_{x,x,x} - \epsilon\,u_{x,x,x,x} - \lambda\,u_{x,x,x,x,x,x} == 0$$

The three parameters λ, μ, and ϵ are real constants. The infinitesimals of the KB equation follow by

```
ikarp = Infinitesimals[karp, {u}, {x, y, t}, {μ, ϵ, λ}];
ikarp // LTF
```

$$\phi_1 == \frac{1}{12}\,(2\,(\mathcal{F}_2)_t + y\,(\mathcal{F}_1)_{t,t})$$

$$\xi_1 == \mathcal{F}_2 + \frac{1}{2}\,y\,(\mathcal{F}_1)_t$$

$$\xi_2 == \mathcal{F}_1$$

$$\xi_3 == k1$$

The result of the calculation is an infinite dimensional symmetry group determined by two arbitrary functions $\mathcal{F}_1 = free[1]$ and $\mathcal{F}_2 = free[2]$. These two functions determine the transformation of the y, x, and u coordinates. We also observe that the group is free of any system parameter λ, ϵ, and μ. The two arbitrary functions *free*[*1*] and *free*[2] do not have to satisfy any other equations. Thus, we can choose them individually. If we assume, for example, that these functions are given by two constants *free*[1][*t*] = *k2* and *free*[2][*t*] = *k3*, we immanently select from the infinite dimensional group those subgroups which describe the invariance of the equation with respect to translations in the independent variables. At the other hand, this special subgroup creates the following reduction:

```
rkarp = LieReduction[karp, {u}, {x, y, t}, {k1, k2, k3},
  {0}]; LTF[Flatten[rkarp]] /. {zeta1 → ζ1, zeta2 → ζ2}
```

$$-\frac{-k1\,t + k3\,x}{k1} - \zeta_1 == 0$$

$$-\frac{k2\,x - k1\,y}{k1} - \zeta_2 == 0$$

$$u - F_1 == 0$$

$$\begin{aligned}
&6\,k1^4\,k3^2\,(F_1)^2_{\zeta_1} + 12\,k1^4\,k2\,k3\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_2} + 6\,k1^4\,k2^2\,(F_1)^2_{\zeta_2} - \\
&\quad k1^5\,k3\,(F_1)_{\zeta_1,\zeta_1} + 6\,k1^4\,k3^2\,F_1\,(F_1)_{\zeta_1,\zeta_1} - k1^5\,k2\,(F_1)_{\zeta_1,\zeta_2} + \\
&\quad 12\,k1^4\,k2\,k3\,F_1\,(F_1)_{\zeta_1,\zeta_2} - k1^6\,(F_1)_{\zeta_2,\zeta_2} + 6\,k1^4\,k2^2\,F_1\,(F_1)_{\zeta_2,\zeta_2} + \\
&\quad k1^3\,k3^3\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} + 3\,k1^3\,k2\,k3^2\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_2} + \\
&\quad 3\,k1^3\,k2^2\,k3\,\mu\,(F_1)_{\zeta_1,\zeta_2,\zeta_2} + k1^3\,k2^3\,\mu\,(F_1)_{\zeta_2,\zeta_2,\zeta_2} - \\
&\quad k1^2\,k3^4\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1} - 4\,k1^2\,k2\,k3^3\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_2} - \\
&\quad 6\,k1^2\,k2^2\,k3^2\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_2,\zeta_2} - 4\,k1^2\,k2^3\,k3\,\epsilon\,(F_1)_{\zeta_1,\zeta_2,\zeta_2,\zeta_2} - \\
&\quad k1^2\,k2^4\,\epsilon\,(F_1)_{\zeta_2,\zeta_2,\zeta_2,\zeta_2} - k3^6\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1} - \\
&\quad 6\,k2\,k3^5\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_2} - 15\,k2^2\,k3^4\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_2,\zeta_2} - \\
&\quad 20\,k2^3\,k3^3\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_2,\zeta_2,\zeta_2} - 15\,k2^4\,k3^2\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_2,\zeta_2,\zeta_2,\zeta_2} - \\
&\quad 6\,k2^5\,k3\,\lambda\,(F_1)_{\zeta_1,\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2} - k2^6\,\lambda\,(F_1)_{\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2} == \\
&0
\end{aligned}$$

The result is a non-linear partial differential equation of sixth order for the similarity function *F1*. The similarity function depends on the two similarity variables *zeta1* and *zeta2*. Thus, we reduced a (2+1)-dimensional PDE to a (1+1)-dimensional PDE. Both similarity variables *zeta1* and *zeta2* are invariants of the KP equation. The reduced equation is again a candidate for Lie's method. For the sake of simplicity, we choose the group constants *k1*, *k2*, and *k3* in a suited way by $k1 = k2 = 1$, and $k3 = v$. The corresponding reduction thus simplifies to

```
rkarp1 = LieReduction[karp, {u}, {x, y, t}, {1, 1, v},
  {0}]; LTF[Flatten[rkarp1]] /. {zeta1 → ζ1, zeta2 → ζ2}
```

$$
\begin{aligned}
&t - v\,x - \zeta_1 == 0\\
&-x + y - \zeta_2 == 0\\
&u - F_1 == 0\\
&6\,v^2\,(F_1)_{\zeta_1}^2 + 12\,v\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_2} +\\
&\quad 6\,(F_1)_{\zeta_2}^2 - v\,(F_1)_{\zeta_1,\zeta_1} + 6\,v^2\,F_1\,(F_1)_{\zeta_1,\zeta_1} - (F_1)_{\zeta_1,\zeta_2} +\\
&\quad 12\,v\,F_1\,(F_1)_{\zeta_1,\zeta_2} - (F_1)_{\zeta_2,\zeta_2} + 6\,F_1\,(F_1)_{\zeta_2,\zeta_2} + v^3\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} +\\
&\quad 3\,v^2\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_2} + 3\,v\,\mu\,(F_1)_{\zeta_1,\zeta_2,\zeta_2} + \mu\,(F_1)_{\zeta_2,\zeta_2,\zeta_2} -\\
&\quad v^4\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1} - 4\,v^3\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_2} - 6\,v^2\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_2,\zeta_2} -\\
&\quad 4\,v\,\epsilon\,(F_1)_{\zeta_1,\zeta_2,\zeta_2,\zeta_2} - \epsilon\,(F_1)_{\zeta_2,\zeta_2,\zeta_2,\zeta_2} - v^6\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1} -\\
&\quad 6\,v^5\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_2} - 15\,v^4\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1,\zeta_2,\zeta_2} -\\
&\quad 20\,v^3\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_2,\zeta_2,\zeta_2} - 15\,v^2\,\lambda\,(F_1)_{\zeta_1,\zeta_1,\zeta_2,\zeta_2,\zeta_2,\zeta_2} -\\
&\quad 6\,v\,\lambda\,(F_1)_{\zeta_1,\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2} - \lambda\,(F_1)_{\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2,\zeta_2} ==\\
&0
\end{aligned}
$$

It is obvious that the reduction of the KB equation is a sixth-order non-linear PDE in 1+1 variables. Thus, the resulting equation is nearly as complicated as the original equation. To simplify things, let us examine models which follow from the KB equation. If we change the parameters in the KB equation in an appropriate way, we get a simplified equation. One example is the reduction of the KB equation by

```
karps1 = karp /. λ → 0; karps1 // LTF
```

$$6\,u_x^2 + u_{x,t} + 6\,u\,u_{x,x} - u_{y,y} - \mu\,u_{x,x,x} - \epsilon\,u_{x,x,x,x} == 0$$

This kind of equation is called a reduced KB equation (rKB) in the following. The infinitesimals of this equation are calculated by

```
infkarps1 = Infinitesimals[karps1, u, {x, y, t}, {μ, ϵ}];
infkarps1 // LTF
```

$$\phi_1 == \frac{1}{12}\,(2\,(\mathcal{F}_2)_t + y\,(\mathcal{F}_1)_{t,t})$$

$$\xi_1 == \mathcal{F}_2 + \frac{1}{2}\,y\,(\mathcal{F}_1)_t$$

$$\xi_2 == \mathcal{F}_1$$

$$\xi_3 == k1$$

Again, we find a group which is determined by two arbitrary functions $\mathcal{F}_1 = free[1]$ and $\mathcal{F}_2 = free[2]$. Both functions depend only on t and are not restricted by any side conditions. It is obvious that the symmetries of the model with $\lambda = 0$ allow the same transformations as the complete model.

The following reduction of the $\lambda = 0$ model assumes that $free[2] = 1 = free[1]$ and $k1 = c$, with c a real constant:

```
rkarps1 = LieReduction[karps1, {u}, {x, y, t}, {1, 1, c},
  {0}]; LTF[Flatten[rkarps1]] /. {zeta1 → ζ1, zeta2 → ζ2}
```

$$t - c\,x - \zeta_1 == 0$$

$$-x + y - \zeta_2 == 0$$

$$u - F_1 == 0$$

$$6\,c^2\,(F_1)^2_{\zeta_1} + 12\,c\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_2} +$$
$$6\,(F_1)^2_{\zeta_2} - c\,(F_1)_{\zeta_1,\zeta_1} + 6\,c^2\,F_1\,(F_1)_{\zeta_1,\zeta_1} -$$
$$(F_1)_{\zeta_1,\zeta_2} + 12\,c\,F_1\,(F_1)_{\zeta_1,\zeta_2} - (F_1)_{\zeta_2,\zeta_2} + 6\,F_1\,(F_1)_{\zeta_2,\zeta_2} +$$
$$c^3\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} + 3\,c^2\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_2} + 3\,c\,\mu\,(F_1)_{\zeta_1,\zeta_2,\zeta_2} +$$
$$\mu\,(F_1)_{\zeta_2,\zeta_2,\zeta_2} - c^4\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1} - 4\,c^3\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_2} -$$
$$6\,c^2\,\epsilon\,(F_1)_{\zeta_1,\zeta_1,\zeta_2,\zeta_2} - 4\,c\,\epsilon\,(F_1)_{\zeta_1,\zeta_2,\zeta_2,\zeta_2} - \epsilon\,(F_1)_{\zeta_2,\zeta_2,\zeta_2,\zeta_2} ==$$
$$0$$

The result is a fourth-order non-linear PDE in the similarity variables *zeta1*, *zeta2*, and *F1*. The reduction is as complicated as the reduction of the full KB equation.

Another simplification of the original KB equation follows if we set ϵ and λ equal to zero:

```
karps2 = karp /. {λ → 0, ϵ → 0}; karps2 // LTF
```

$$6\,u_x^2 + u_{x,t} + 6\,u\,u_{x,x} - u_{y,y} - \mu\,u_{x,x,x} == 0$$

The resulting equation is known as the ZK equation. The symmetries of the ZK equation are determined by

```
infkarps2 = Infinitesimals[karps2, u, {x, y, t}, {μ}];
infkarps2 // LTF
```

$$\phi_1 == \frac{1}{12}\,(-6\,k2\,u + 2\,(\mathcal{F}_2)_t + y\,(\mathcal{F}_1)_{t,t})$$
$$\xi_1 == \frac{1}{2}\,(k2\,x + 2\,\mathcal{F}_2 + y\,(\mathcal{F}_1)_t)$$
$$\xi_2 == \frac{3\,k2\,y}{4} + \mathcal{F}_1$$
$$\xi_3 == k1 + k2\,t$$

For this model, we again find an infinite symmetry group depending on two arbitrary functions. Contrary to the models examined above, the discrete part of the symmetry group increases by one group parameter. The reduction of this model for a scaling group with $k2 = 1$, $k1 = 0$, and the arbitrary functions set equal to zero follows by

```
rkarps21 = LieReduction[karps2, {u}, {x, y, t}, {x/2, 3 y/4, t},
  {-u/2}] // PowerExpand;
LTF[Flatten[rkarps21]] /. {zeta1 → ζ1, zeta2 → ζ2}
```

$$\frac{t}{x^2} - \zeta_1 == 0$$
$$\frac{y}{x^{3/2}} - \zeta_2 == 0$$
$$u\,x - F_1 == 0$$
$$\zeta_1^{7/4}\,(48\,I\,t^{1/4}\,\mu\,F_1 + 144\,I\,t^{1/4}\,F_1^2 - 24\,I\,t^{1/4}\,(F_1)_{\zeta_1} +$$
$$432\,I\,t^{1/4}\,\mu\,\zeta_1\,(F_1)_{\zeta_1} + 672\,I\,t^{1/4}\,F_1\,\zeta_1\,(F_1)_{\zeta_1} +$$
$$192\,I\,t^{1/4}\,\zeta_1^2\,(F_1)_{\zeta_1}^2 + 267\,I\,t^{1/4}\,\mu\,\zeta_2\,(F_1)_{\zeta_2} +$$
$$468\,I\,t^{1/4}\,F_1\,\zeta_2\,(F_1)_{\zeta_2} + 288\,I\,t^{1/4}\,\zeta_1\,\zeta_2\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_2} +$$
$$108\,I\,t^{1/4}\,\zeta_2^2\,(F_1)_{\zeta_2}^2 - 16\,I\,t^{1/4}\,\zeta_1\,(F_1)_{\zeta_1,\zeta_1} +$$
$$384\,I\,t^{1/4}\,\mu\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1} + 192\,I\,t^{1/4}\,F_1\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1} -$$
$$12\,I\,t^{1/4}\,\zeta_2\,(F_1)_{\zeta_1,\zeta_2} + 540\,I\,t^{1/4}\,\mu\,\zeta_1\,\zeta_2\,(F_1)_{\zeta_1,\zeta_2} +$$
$$288\,I\,t^{1/4}\,F_1\,\zeta_1\,\zeta_2\,(F_1)_{\zeta_1,\zeta_2} - 8\,I\,t^{1/4}\,(F_1)_{\zeta_2,\zeta_2} +$$
$$189\,I\,t^{1/4}\,\mu\,\zeta_2^2\,(F_1)_{\zeta_2,\zeta_2} + 108\,I\,t^{1/4}\,F_1\,\zeta_2^2\,(F_1)_{\zeta_2,\zeta_2} +$$
$$64\,I\,t^{1/4}\,\mu\,\zeta_1^3\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} + 144\,I\,t^{1/4}\,\mu\,\zeta_1^2\,\zeta_2\,(F_1)_{\zeta_1,\zeta_1,\zeta_2} +$$
$$108\,I\,t^{1/4}\,\mu\,\zeta_1\,\zeta_2^2\,(F_1)_{\zeta_1,\zeta_2,\zeta_2} + 27\,I\,t^{1/4}\,\mu\,\zeta_2^3\,(F_1)_{\zeta_2,\zeta_2,\zeta_2}) ==$$
$$0$$

representing a third-order non-linear PDE. Another reduction follows for translations as symmetry transformations:

```
rkarps22 = LieReduction[karps2, {u}, {x, y, t}, {1, 1, c},
  {0}]; LTF[Flatten[rkarps22]] /. {zeta1 → ζ1, zeta2 → ζ2}
```

$$t - c\,x - \zeta_1 == 0$$
$$-x + y - \zeta_2 == 0$$
$$u - F_1 == 0$$
$$6\,c^2\,(F_1)^2_{\zeta_1} + 12\,c\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_2} +$$
$$6\,(F_1)^2_{\zeta_2} - c\,(F_1)_{\zeta_1,\zeta_1} + 6\,c^2\,F_1\,(F_1)_{\zeta_1,\zeta_1} - (F_1)_{\zeta_1,\zeta_2} +$$
$$12\,c\,F_1\,(F_1)_{\zeta_1,\zeta_2} - (F_1)_{\zeta_2,\zeta_2} + 6\,F_1\,(F_1)_{\zeta_2,\zeta_2} + c^3\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} +$$
$$3\,c^2\,\mu\,(F_1)_{\zeta_1,\zeta_1,\zeta_2} + 3\,c\,\mu\,(F_1)_{\zeta_1,\zeta_2,\zeta_2} + \mu\,(F_1)_{\zeta_2,\zeta_2,\zeta_2} ==$$
$$0$$

We use this similarity representation to apply Lie's procedure again. The symmetries of the reduction follow by

```
irkarps22 = Infinitesimals[rkarps22[[3, 1]], F1, {zeta1, zeta2},
  {c, μ}, SubstitutionRules → {∂{zeta1,3} F1[zeta1, zeta2]}];
irkarps22 // LTF
```

$$\phi_1 == \frac{(1 - 4\,c + 24\,c^2\,F1)\,k3}{-6 + 12\,c}$$
$$\xi_1 == \frac{(-1 + 2\,c)\,k1 - c\,k3\,(zeta1 + 4\,c\,zeta1 - c\,zeta2)}{-1 + 2\,c}$$
$$\xi_2 == \frac{(-1 + 2\,c)\,k2 + k3\,((-1 + 2\,c)\,zeta1 + (1 - 6\,c)\,c\,zeta2)}{-1 + 2\,c}$$

The result is a finite symmetry group of order three, allowing us a further reduction. Before we execute this step, we rename the variables to simplify the equations:

```
eqv = {rkarps22[[3, 1, 1]]} /. {F1 → H, zeta1 → ζ1, zeta2 → ζ2};
eqv // LTF
```

$$6\,c^2\,H^2_{\zeta1} + 12\,c\,H_{\zeta1}\,H_{\zeta2} + 6\,H^2_{\zeta2} - c\,H_{\zeta1,\zeta1} +$$
$$6\,c^2\,H\,H_{\zeta1,\zeta1} - H_{\zeta1,\zeta2} + 12\,c\,H\,H_{\zeta1,\zeta2} - H_{\zeta2,\zeta2} + 6\,H\,H_{\zeta2,\zeta2} +$$
$$c^3\,\mu\,H_{\zeta1,\zeta1,\zeta1} + 3\,c^2\,\mu\,H_{\zeta1,\zeta1,\zeta2} + 3\,c\,\mu\,H_{\zeta1,\zeta2,\zeta2} + \mu\,H_{\zeta2,\zeta2,\zeta2} ==$$
$$0$$

For the non-linear third-order PDE in the second similarity representation, we select the subgroup of translations to carry out another reduction. The infinitesimals for this case are given by

```
inf1 = {xi[1][zeta1, zeta2, F1], xi[2][zeta1, zeta2, F1]} /.
    irkarps22 /. {k1 → 1, k2 → v, k3 → 0} /.
  {F1 → H, zeta1 → ζ1, zeta2 → ζ2}
```

```
{1, v}
```

```
inf2 = {phi[1][zeta1, zeta2, F1]} /. irkarps22 /.
   {k1 → 1, k2 → v, k3 → 0} /.
  {F1 → H, zeta1 → ζ1, zeta2 → ζ2}
```

```
{0}
```

The corresponding reduction follows from

```
rkarps221 = LieReduction[eqv, {H}, {ζ1, ζ2}, inf1, inf2];
LTF[Flatten[rkarps221]] /. {zeta1 → ζ}
```

$$-\zeta - v\,\zeta 1 + \zeta 2 == 0$$

$$H - F_1 == 0$$

$$6\,(F_1)_\zeta^2 - 12\,c\,v\,(F_1)_\zeta^2 + 6\,c^2\,v^2\,(F_1)_\zeta^2 -$$
$$(F_1)_{\zeta,\zeta} + v\,(F_1)_{\zeta,\zeta} - c\,v^2\,(F_1)_{\zeta,\zeta} + 6\,F_1\,(F_1)_{\zeta,\zeta} -$$
$$12\,c\,v\,F_1\,(F_1)_{\zeta,\zeta} + 6\,c^2\,v^2\,F_1\,(F_1)_{\zeta,\zeta} + \mu\,(F_1)_{\zeta,\zeta,\zeta} -$$
$$3\,c\,v\,\mu\,(F_1)_{\zeta,\zeta,\zeta} + 3\,c^2\,v^2\,\mu\,(F_1)_{\zeta,\zeta,\zeta} - c^3\,v^3\,\mu\,(F_1)_{\zeta,\zeta,\zeta} ==$$
$$0$$

The result is a third-order non-linear ODE which allows a two-dimensional discrete symmetry group depending on group parameters c and v of the preceding reductions:

```
irkarps221 = Infinitesimals[rkarps221[[3, 1]], F1, {zeta1},
  {c, μ, v}, SubstitutionRules → {∂{zeta1,3} F1[zeta1]}];
irkarps221 // LTF
```

$$\phi_1 == \frac{k2\,(-1 + v - c\,v^2 + 6\,F1\,(-1 + c\,v)^2)}{6\,(-1 + c\,v)^2}$$

$$\xi_1 == k1 - k2\,zeta1$$

The result indicates that the third-order ODE is at least reducible to a second-order ODE which, in fact, is possible by an integration with respect to *zeta1*:

```
firstIntegral = ∫rkarps221[[3, 1, 1]] dzeta1 == c1;
LTF[firstIntegral] /. zeta1 → ζ1
```

$$-c1 + (-1 + v - c\,v^2 + 6\,F_1 - 12\,c\,v\,F_1 + 6\,c^2\,v^2\,F_1)\,(F_1)_{\zeta_1}$$
$$- (-1 + c\,v)^3\,\mu\,(F_1)_{\zeta_1,\zeta_1} == 0$$

The resulting second-order ODE is solved by

```
sol1 = DSolve[firstIntegral, F1, zeta1]
```

$$\{\{F1 \to \Big(\big((1 - v + c\,v^2)\,\text{AiryBi}\big[(13 - 2\,c\,v^3 + c^2\,v^4 + 12\,c1\,\#1 - 2\,v$$
$$(1 + 12\,c\,(1 + c1\,\#1)) + v^2\,(1 + 2\,c + 12\,c^2\,(1 + c1\,\#1)))\big/$$
$$\left(4\,3^{2/3}\,(c1\,(-1 + c\,v)^2)^{2/3}\right)\big] -$$
$$2\,3^{1/3}\,(c1\,(-1 + c\,v)^2)^{1/3}\,\text{AiryBiPrime}\Big[$$
$$(13 - 2\,c\,v^3 + c^2\,v^4 + 12\,c1\,\#1 - 2\,v\,(1 + 12\,c\,(1 + c1\,\#1)) +$$
$$v^2\,(1 + 2\,c + 12\,c^2\,(1 + c1\,\#1)))\big/$$
$$\left(4\,3^{2/3}\,(c1\,(-1 + c\,v)^2)^{2/3}\right)\Big] +$$

```
(AiryAi[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] - v AiryAi[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] + c v^2 AiryAi[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] -
   2 3^(1/3) (c1 (-1 + c v)^2)^(1/3) AiryAiPrime[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))])
  C[1]) /
(6 (-1 + c v)^2 (AiryBi[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] + AiryAi[
     (13 - 2 c v^3 + c^2 v^4 + 12 c1 #1 - 2 v (1 + 12 c (1 + c1 #1)) +
        v^2 (1 + 2 c + 12 c^2 (1 + c1 #1))) /
      (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))]
  C[1])) &)}}
```

The result is a complicated expression containing special functions of the Airy type. However, the solution simplifies if we set the integration constant *c1* equal to zero.

```
firstIntegral = firstIntegral /. c1 → 0;
LTF[firstIntegral] /. zeta1 → ζ1
```

$$(-1 + v - c\,v^2 + 6\,F_1 - 12\,c\,v\,F_1 + 6\,c^2\,v^2\,F_1)\,(F_1)_{\zeta_1} - (-1 + c\,v)^3\,\mu\,(F_1)_{\zeta_1,\zeta_1} == 0$$

The solution now reads

```
sol2 = DSolve[firstIntegral, F1, zeta1]
```

$$\{\{F1 \to \left(\frac{1}{6}\left(\frac{1}{(1 - c\,v)^2} - \frac{v}{1 - c\,v} + \frac{1}{(1 - c\,v)^2}\left(\sqrt{(-1 + 2\,v - v^2 - 2\,c\,v^2 + 2\,c\,v^3 - c^2\,v^4 - 12\,C[2] + 24\,c\,v\,C[2] - 12\,c^2\,v^2\,C[2])}\right.\right.\right.$$

```
Tan[
```

$$\frac{1}{6}\left(-\left(3\,\#1\,\sqrt{(-1+2\,v-v^2-2\,c\,v^2+2\,c\,v^3-c^2\,v^4-12\,C[2]+24\,c\,v\,C[2]-12\,c^2\,v^2\,C[2])}\right)\Big/\left((1-c\,v)^3\,\mu\right)+\frac{1}{(1-c\,v)^2}\left(C[1]\,\sqrt{(-1+2\,v-v^2-2\,c\,v^2+2\,c\,v^3-c^2\,v^4-12\,C[2]+24\,c\,v\,C[2]-12\,c^2\,v^2\,C[2])}\right)\right)\&\Big)\Big\}\Big\}$$

containing only the Tan[] function. At this stage of the calculation, we derived a solution for a special subgroup of all the possible symmetries of the original equation. This special function will help us to represent the solution in the original variables. To get the representation of the solution in x, y, t, and u, we have to invert all the similarity transformations carried out above:

```
sol = u → Function[{x, y, t}, $r] /. $r →
   (F1[t - c x, -x + y] /. (F1 → Function[{zeta1, zeta2}, $w] /.
      $w → (((H == F1[-v ζ1 + ζ2] /. sol1) /.
            {H → F1, ζ1 → zeta1, ζ2 → zeta2}) /.
         {Equal[a_, b_]} -> b)))
```

```
u → Function[{x, y, t}, ((1 - v + c v^2)
     AiryBi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] -
     2 3^(1/3) (c1 (-1 + c v)^2)^(1/3)
      AiryBiPrime[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] +
     (AiryAi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] -
      v AiryAi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] +
      c v^2 AiryAi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
```

```
v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
          (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] - 2 3^(1/3) (c1 (-1 + c v)^2)^(1/3)
        AiryAiPrime[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
             2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
             v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
           (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))]]
     C[1]) /
   (6 (-1 + c v)^2
     (AiryBi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))] +
      AiryAi[(13 - 2 c v^3 + c^2 v^4 + 12 c1 (-x - v (t - c x) + y) -
          2 v (1 + 12 c (1 + c1 (-x - v (t - c x) + y))) +
          v^2 (1 + 2 c + 12 c^2 (1 + c1 (-x - v (t - c x) + y)))) /
        (4 3^(2/3) (c1 (-1 + c v)^2)^(2/3))]
      C[1]))]
```

The second type of solution allowed by the reduced KB equation was derived for c1 = 0. In original variables, this solution reads

```
sols = u → Function[{x, y, t}, $r] /. $r →
   (F1[t - c x, -x + y] /. (F1 → Function[{zeta1, zeta2}, $w] /.
      $w → (((H == F1[-v ζ1 + ζ2] /. sol2) /.
           {H → F1, ζ1 → zeta1, ζ2 → zeta2}) /.
        {Equal[a_, b_]} -> b)))
```

```
u → Function[{x, y, t},
  1/6 (1/(1 - c v)^2 - v/(1 - c v) + 1/(1 - c v)^2 (√(-1 + 2 v - v^2 - 2 c v^2 +
        2 c v^3 - c^2 v^4 - 12 C[2] + 24 c v C[2] - 12 c^2 v^2 C[2])
      Tan[1/6 (-1/((1 - c v)^3 μ) (3 (-x - v (t - c x) + y)
          √(-1 + 2 v - v^2 - 2 c v^2 + 2 c v^3 - c^2 v^4 -
            12 C[2] + 24 c v C[2] - 12 c^2 v^2 C[2])) +
        1/(1 - c v)^2
         (C[1] √(-1 + 2 v - v^2 - 2 c v^2 + 2 c v^3 - c^2 v^4 - 12 C[2] +
            24 c v C[2] - 12 c^2 v^2 C[2])))]]]]
```

We can inspect the ZK equation by this solution:

```
karp /. {λ → 0, ϵ → 0} /. sols // Simplify
```

```
{0}
```

The result is that the given solution satisfies the ZK equation. To get an impression on how the complicated symbolic solutions behave in the x, y, t space, let us plot the solutions for different times over the (x, y)-plane. We choose the parameters for both plots as

```
parameters = {c → 2, v → 1, C[1] → 0, C[2] → 1, μ → 2, c1 → 1};
```

The two functions are created by

```
pl = u[x, y, t] /. sol /. parameters;
```

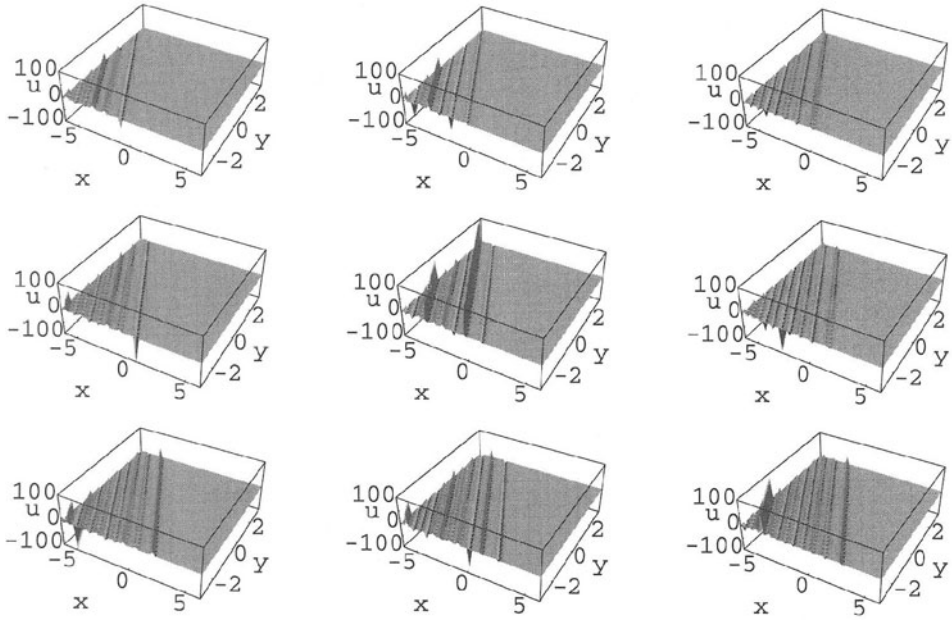
for the general case with c1 = 1

```
pls = u[x, y, t] /. sols /. parameters
```

$$\frac{1}{6}\,(2 + 4\,\mathrm{Tanh}[t - x - y])$$

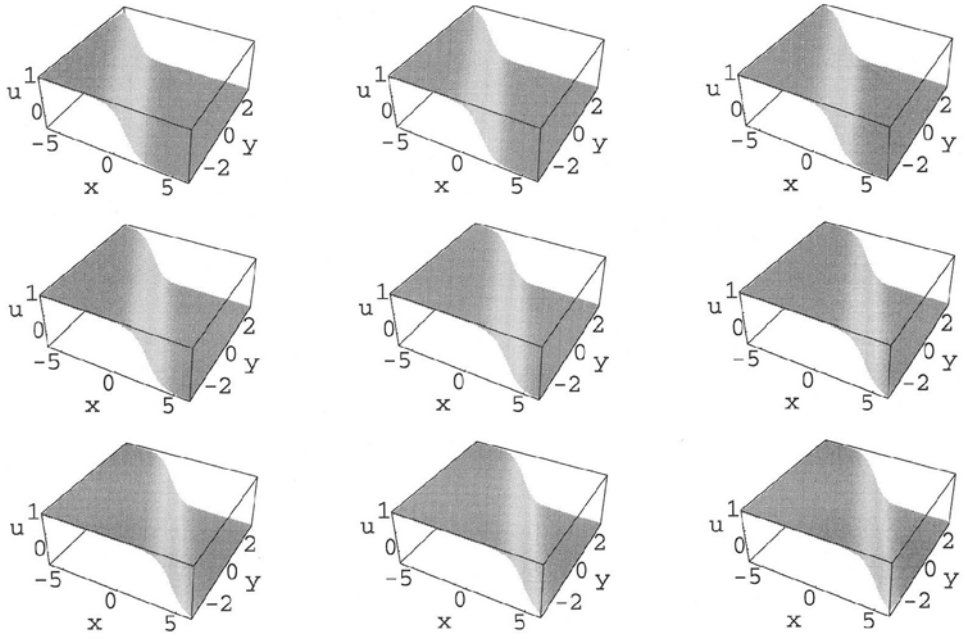
The two special solutions serve to create an animation for $-2 \le t \le 2$ in steps of $\delta t = 1/4$. The general solution looks like a bunch of crests moving from the left to the right:

```
Do[Plot3D[Evaluate[pl /. t → τ],
  {x, -2 π, 2 π}, {y, -π, π}, AxesLabel → {"x", "y", "u"},
  PlotRange → {{-2 π, 2 π}, {-π, π}, {-100, 100}},
  PlotPoints → 60, Mesh → False],
 {τ, -2, 2, .25}]
```



In the animation of these pictures, we observe that a train of waves moves from the left to the right. The simpler solution of the ZK equation is given by a Tanh[] function representing a propagating step from the left to the right:

```
Do[Plot3D[pls /. t → τ,
  {x, -2 π, 2 π}, {y, -π, π}, AxesLabel → {"x", "y", "u"},
  PlotRange → All, PlotPoints → 40, Mesh → False],
 {τ, -2, 2, .25}]
```



The animation shows that the step is stable and does not disperse. The solution is a soliton which propagates in a firm form. □

This section was concerned with the reduction of the original equations (PDEs) either to an ODE or to a PDE with less independent variables. We demonstrated that the reductions are instrumental to find symbolic solutions. Even if we fail to write down an analytic solution, we can utilize the numerical capabilities of *Mathematica* to solve the reduction. For PDEs in more than two independent variables, we can apply the functions of *MathLie* several times to find the reductions and even the solutions. The following sections will demonstrate the application of the functions of *MathLie* to physical problems.

5.6. Working Examples

This section contains some working examples to show the application of the package *MathLie*. We discuss the necessary steps for solving some physical and mathematical problems. The first example deals with the diffusion equation applied to the problem of thermal oscillations on a surface. A second example discusses the application of *MathLie* in the derivation of symmetries for a model describing a single flux line in a superconductor. Several applications from hydrodynamics demonstrate the engineering challenge of the symmetry method. The first atomic explosion serves to demonstrate the extraordinary success of symmetry analysis for estimating unknown quantities from a movie. The formation of droplets is an example currently under discussion in industrial applications.

5.6.1 The Diffusion Equation

The diffusion equation is one of the extensively discussed examples in symmetry analysis. Since the beginning of the theory, this equation was used to demonstrate the usefulness of the symmetry method to derive solutions. Lie himself used the diffusion equation as an example to illustrate the capabilities of the method. In his work, the diffusion equation is one of the equations comprehensively discussed. The diffusion equation is used by many other authors to introduce the method and show how the symmetry method can be extended in different ways. The diffusion equation here is chosen as a reminiscence to the tremendous work of Lie. The example of the diffusion equation illustrates how *MathLie* can be exerted to derive the symmetries of this equation. We also will show how the gained information can be employed to solve a particular problem. The problem we discuss is a boundary value problem concerned with the seasonal oscillations of the temperature on the surface of the earth.

5.6.2 The Earthworm's New Year Problem

This problem is concerned with the propagation of damped temperature waves into the earth due to annual temperature variation. This example is discussed extensively by Bluman and Cole [1974]. Imagine a worm has to decide when he has to celebrate New Year. The only indicator of seasonal changes he has is the variation of temperature. Let us assume that the worm will celebrate New Year when he measures the lowest temperature in the year. The propagation of the seasonal temperature wave the worm must follow is described by a diffusion process. The measured quantity is the temperature denoted by the variable

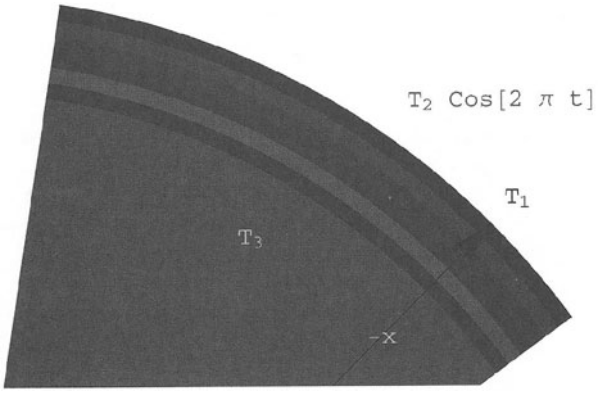
```
U = u[x, t];
```

The field u measures the temperature variation toward the center of the earth ($x > 0$) at a certain time ($t > 0$). The diffusion process of heat is governed by the equation

```
diffusion = {∂t U - ∂{x,2} U}; diffusion // LTF
```

$$u_t - u_{x,x} == 0$$

A sketch of the physical situation we have in mind is given in the following figure:



The constant temperatures T_1 and T_3 describe the temperature on the surface and in the interior of the earth at a certain depth. The seasonal changes of the temperature near the surface are assumed to be $T_2 \cos(2\pi t)$ and are due to radiation changes of the sun. So, besides the diffusion equation, the solution of the problem has to satisfy additionally boundary conditions. In deriving the solution of this boundary value problem, we have to solve the diffusion equation for the given boundary conditions on the surface and in the center. Examining the symmetries of the diffusion equation allows us to decide how the general equation is transformed and what types of transformation are allowed to find the solution. We use the functions LieEquations[] and LieSolve[] of the package *MathLie* to create another example in our data basis. We start with the creation of the information file and carry out the symmetry analysis by applying the function LieSolve[]. After the derivation of the symmetries, we will consider the boundary values and check the invariance of these conditions under the symmetry transformations. In a last step, we will reduce the partial differential equation to an ordinary differential equation. The solution of this ODE will provide us with the information the worm needs to decide when the turn of the year happens.

5.6.2.1 Symmetry Analysis

The first step in an automatic symmetry analysis is the collection of all the information available on the equation. The package *MathLie* offers the function LieEquations[] to collect this information. If you solve a differential equation with pencil and paper, you need to know the equation itself. You also have to know the names of the dependent variables and the independent variables. Sometimes, the equation contains some parameters which you can also save by using the function LieEquations[]. If you assemble a greater number of equations in a database, it is good practice to supply the files with information on the problem and on the sources from which the equation comes. In *MathLie*, all this information is stored in a single file whose name is used by LieEquation[] as first argument. The information we need to handle in the example for the earthworm's New Year problem is given below:

```
LieEquations["diffuw.dgl", diffusion, {u}, {x, t}, {},
 {"The earthworm's New Year problem"},
 {{"G.W. Bluman and J.D. Cole"},
  {"Similarity Methods for Differential Equations"},
  {"Springer, New York, 1974"}, {"pp. 233"}}]
```

After the completion of these lines, the function LieEquations[] created a file containing the information necessary for the symmetry analysis. The content of the file *diffuw.dgl* looks like

```
! ! diffuw.dgl

Title = {The earthworm's New Year problem}
Source = {{G.W. Bluman and J.D. Cole},
 {Similarity Methods for Differential Equations},
 {Springer, New York, 1974}, {pp. 233}}
IndepVar = {x, t}
DependVar = {u}
EqList =
 {Derivative[0, 1][u][x, t] - Derivative[2, 0][u][x, t]}
SubsList = {Derivative[0, 1][u][x, t]}
ParameterS = {}
ListXi = {}
ListPhi = {}
```

The symmetries of the diffusion equation are calculated by applying the function LieSolve[] to the collected information. The function LieSolve[] exerts the invariance condition based on the prolongation formalism. The prolongation of the equation is calculated by using the Fréchet derivative. Knowing the prolongation of the equation, the coefficients of the derivatives of u are extracted. The redundant information

contained in these equations is eliminated in the next step by inserting the diffusion equation itself into the prolongation. After the extraction of the coefficients of the derivatives, a system of determining equations for the infinitesimals results. The determining equations are linear but coupled. In the next step, a general canonical representation of these equations is calculated by LieSolve[]. The last calculation step of LieSolve[] solves the general canonical form. The result of this sequence of steps is an explicit representation of the infinitesimals for the diffusion equation:

```
LieSolve["diffuw.dgl"] // LTF
```

$$-(\mathcal{F}_1)_t + (\mathcal{F}_1)_{x,x} == 0$$

$$\xi_1 == k5 - 2\,k2\,t + k6\,x + k4\,t\,x$$

$$\xi_2 == k3 + t\,(2\,k6 + k4\,t)$$

$$\phi_1 == u\left(k1 - \frac{k4\,t}{2} + k2\,x - \frac{k4\,x^2}{4}\right) + \mathcal{F}_1$$

The result is an infinite dimensional Lie group containing a six-dimensional discrete subgroup. The group parameters are denoted by *k1–k6*. The discrete symmetries serve to construct similarity solutions.

5.6.2.2. Similarity Solution

Knowing the infinitesimals of the diffusion equation, we are ready to solve the boundary value problem for the earthworm's problem. The additional condition that the symmetries have to satisfy are confined in the side conditions on the surface and in the center of the earth. These boundary conditions are

```
b1 = v[0, t] == T2 Exp[I 2 π t]
```

$$v[0, t] == E^{2\,I\,\pi\,t}\,T_2$$

```
b2 = v[∞, t] == 0
```

$$v[\infty, t] == 0$$

Using this complex-valued representation for the temperature, $u = \mathrm{Re}(v)$, we can check the invariance of the boundary conditions by using the infinitesimals. The invariance of $x = 0$ implies that

```
(ListXi[[1]] /. x → 0) == 0
```

$$k5 - 2\,k2\,t == 0$$

From this relation, it follows that the group constants *k5* and *k2* have to vanish. We collect these results in a list of rules:

```
rule = {k5 → 0, k2 → 0}
```

$$\{k5 \to 0,\ k2 \to 0\}$$

The invariance of the first boundary condition *b1*, on the other hand, yields the relation

```
equt1 = k1 - k4 t/2 == I 2 π (ListXi[[2]] /. x → 0)
```

$$k1 - \frac{k4\ t}{2} == 2\ I\ \pi\ (k3 + t\ (2\ k6 + k4\ t))$$

which has the general solutions

```
res = Reduce[CoefficientList[equt1 /. 1_ == r_ → 1 - r, t]
  == {0, 0, 0}, {k1, k3, k4, k6}]
```

```
k1 == 2 I k3 π && k4 == 0 && k6 == 0
```

We transform the equations to rules and add it to the list of rules for the group constants:

```
AppendTo[rule, ToRules[res]]; rule = Flatten[rule]
```

```
{k5 → 0, k2 → 0, k1 → 2 I k3 π, k4 → 0, k6 → 0}
```

The general representation of the infinitesimals for the diffusion equation thus reduces to

```
Infinitesimals = {ListXi, ListPhi} /. rule /.
  {free[_][___] → 0, u[___] → u}
```

```
{{0, k3}, {2 I k3 π u}}
```

The reduction of the original equation follows by applying the function LieReduction[] in the subgroup of the diffusion equation:

```
LTF[Flatten[LieReduction[diffusion, {u}, {x, t},
 Infinitesimals[[1]], Infinitesimals[[2]]]] /. zeta1 → ζ1
```

$$x - \zeta_1 == 0$$

$$E^{-2\ I\ \pi\ t}\ u - F_1 == 0$$

$$I\ E^{2\ I\ \pi\ t}\ (2\ \pi\ F_1 + I\ (F_1)_{\zeta_1,\zeta_1}) == 0$$

The similarity variable is *zeta1* = *x* and the solution has the similarity form

```
SimilaritySolution = u → Function[{x, t},
   Exp[I 2 π t] F1[x]]
```

```
u → Function[{x, t}, Exp[I 2 π t] F1[x]]
```

Substituting the similarity solution into the diffusion equation gives us

```
equat2 = Expand[ diffusion /. SimilaritySolution / Exp[I 2 π t] ][[1]] == 0;

equat2 // LTF
```

$$2\,I\,\pi\,F_1 - (F_1)_{x,x} == 0$$

This second-order equation also contained in the result of LieReduction[] is solved by applying the function DSolve[]:

```
res2 = Simplify[DSolve[equat2, F1, x]]
```

$$\left\{\left\{F1 \to \left(E^{(-1-I)\sqrt{\pi}\ \#1}\ C[1] + E^{(1+I)\sqrt{\pi}\ \#1}\ C[2]\&\right)\right\}\right\}$$

Examining the behavior for $x \to \infty$, we find that the second constant of integration *C[2]* has to vanish to satisfy the second boundary condition *b2*

```
h1 = Factor[Expand[Simplify[
     TrigToExp[ComplexExpand[F1[x] /. res2]]]] /.
    {C[2] → 0}]
```

$$\left\{E^{(-1-I)\sqrt{\pi}\ x}\ C[1]\right\}$$

Comparing the result with the first boundary condition, we observe that the real part of *C[1]* is given by T_2:

```
h2 = (h1 /. C[1] → T2)[[1]]
```

$$E^{(-1-I)\sqrt{\pi}\ x}\ T_2$$

The complete solution is thus given by

```
SimSol = SimilaritySolution /. F1[x] → h2
```

$$u \to \text{Function}\left[\{x, t\}, \text{Exp}[I\,2\,\pi\,t]\ \left(E^{(-1-I)\sqrt{\pi}\ x}\ T_2\right)\right]$$

After a rearrangement of terms in the exponent,

```
sol = u[x, t] /. SimSol /.
  c_. Exp[a_. Complex[d_, e_] + b_. Complex[f_, g_]] →
   c Exp[(a d + b f) + (a e + b g) I]
```

$$E^{-\sqrt{\pi}\ x + I\ (2\,\pi\,t - \sqrt{\pi}\ x)}\ T_2$$

```
Re[sol]
```

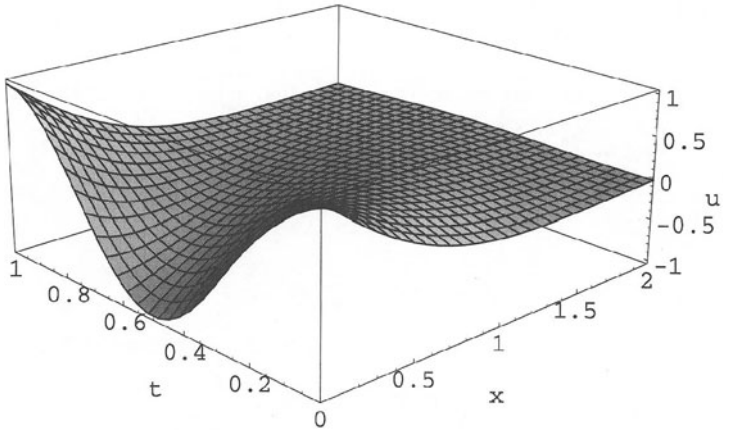
$$\text{Re}\left[E^{-\sqrt{\pi}\ x + I\ (2\,\pi\,t - \sqrt{\pi}\ x)}\ T_2\right]$$

we can extract the real part of the solution by

```
solution = Re[ExpToTrig[sol]] // Simplify
Re[ (Cosh[ (1 + I)
        ( (-1 - I) π t + x (Cosh[ Log[π] / 2 ] + Sinh[ Log[π] / 2 ])) ] - Sinh[
      (1 + I) ( (-1 - I) π t + x (Cosh[ Log[π] / 2 ] + Sinh[ Log[π] / 2 ])) ]) /
  T2 ]
```

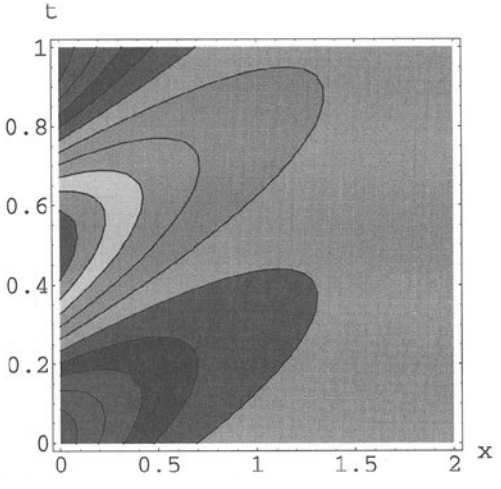
The derived solution describes the temperature variation near the surface of the earth. The solution indicates that the spatial and temporal coordinates are coupled. If now $t = T$ is the New Year of a person on the surface, then an earthworm will celebrate New Year at time $t = T + \sqrt{1/2\pi}\ x_l$, where x_l is the depth of the earthworm. The solution in the time and spatial coordinate with unique T_2 looks like

```
Plot3D[solution /. T2 → 1, {x, 0, 2}, {t, 0, 1},
 PlotRange → All, PlotPoints → 30,
 AxesLabel → {"x", "t", "u"},
 ViewPoint → {-2.468, -2.587, 1.256}]
```



A contour plot of the temperature variation illustrates that the worm in a depth of about 1.5 meters cannot feel any changes in the temperature. Worms which are between the surface and the 1.5-meter limit are able to realize the temperature changes during the year. However, the contour plot shows also that a worm located at a certain depth will measure a certain value of the temperature some time later than an observer on the surface. This delay is larger for worms living in a deeper region.

```
ContourPlot[solution /. T2 → 1, {x, 0, 2}, {t, 0, 1},
 PlotRange → All,
 PlotPoints → 30, AxesLabel → {"x", "t"},
 ColorFunction → Hue, Axes → True]
```



This example demonstrates that the package *MathLie* is not only helpful in finding the symmetries of a PDE but also can facilitate the solution of boundary value problems.

5.6.3 Single Flux Line in Superconductors

In another example, let us examine an equation published by Tang et al. [1994]. These authors discuss the dynamics and noise spectra of a driven single flux line in superconductors. The authors examine the low-temperature dynamics of a single flux line in a bulk type-II superconductor, driven by the Lorentz force acting near the sample surface, both near and above the depinning threshold. The equation of motion they derive without considering the random fluctuations is given by

$$\texttt{tang} = \left\{\partial_t \texttt{u[x, t]} - \frac{\texttt{k}\,\partial_{\{x,2\}}\texttt{u[x, t]}}{1 + (\partial_x \texttt{u[x, t]})^2}\right\};\ \texttt{tang // LTF}$$

$$u_t - \frac{k\,u_{x,x}}{1 + u_x^2} == 0$$

where u is the shape function of the flux line and k is a constant. As Tang et al. note the equation of motion in its two-dimensional representation is a valid approximation when the driving force is very large ($j \gg j_c$) and the string moves with a high velocity. The authors discuss only the steady-state solution of the non-linear problem. We will demonstrate that the symmetries of the equation are the cornerstone for the solution of the non-steady equation. We will add this example to our database of equations. The input parameters needed to create the data file containing the information for Lie[] is given by the independent variables

```
independentVariables = {x, t}
{x, t}
```

and the dependent variables

```
dependentVariables = {u}
{u}
```

The parameters of the equations are collected in the list

```
parameters = {k}
{k}
```

The parameter k combines the critical field H_c, the Ginzburg-Landau parameter and the damping coefficient of the Bardeen-Stephen model. The physical interpretation of these parameters is discussed, for example, in the book by Tinkham [1975]. We choose the title of the equation to be

```
title = {"Single Flux Line in Superconductors"}
{Single Flux Line in Superconductors}
```

The source of the equation, contained in a nested list, is given by

```
source = {{"C. Tang, S. Feng and L. Golubovic"},
  {"Dynamics and Noise Spectra of a
     Driven SingleFlux  Line in  Superconductors"},
  {"Phys. Rev. Lett. 72, 1264-1267, (1994)"},
  {"equation 7"}}

{{C. Tang, S. Feng and L. Golubovic},
 {Dynamics and Noise Spectra of a
    Driven SingleFlux  Line in  Superconductors},
 {Phys. Rev. Lett. 72, 1264-1267, (1994)},
 {equation 7}}
```

All the information available on the equation is now stored in the file *ctang.dgl*. The function LieEquation[] is designed to do it for us. Be sure that you do not use *.deq* as an extension of your file names. The extension *.deq* is reserved for a file containing the information on the determining equation for the infinitesimals. This file is automatically created by the functions Lie[], LieSolve[], and LieStructureForm[].

```
LieEquations["ctang.dgl", tang,
 dependentVariables, independentVariables,
 parameters, title, source]
```

After the completion of this line, we added another example to our database of differential equations. The next step is the determination of the symmetries.

5.6.3.1 Symmetries

Using the information collected in the file *ctang.dgl* in the calculation for the infinitesimals, we start the symmetry analysis with

```
LieSolve["ctang.dgl"] // LTF
```

$$\xi_2 == k1 + 2\,k5\,t$$
$$\phi_1 == k2 + k5\,u + k3\,x$$
$$\xi_1 == k4 - k3\,u + k5\,x$$

The application of LieSolve[] results in a list of infinitesimals representing a finite five-dimensional point group. Curiously, the result is that the infinitesimal ξ_1 depends on the variable u. Such a dependence is quite unusual, although it is possible. All the examples discussed so far do not show this kind of symmetry (cf. the Burgers and heat equation). The symmetry of the subgroup related to this special kind of transformation allows a rotation in the (x, u)-plane. This rotation keeps the equation unchanged.

The function LieSolve[] produces, in addition to the infinitesimals, additional information not displayed on the screen. These results, like the determining equations of the equation, the original equations themselves, the substitutions, etc. are collected during the calculation in a separate global variable called *FinalResult*. The reason why we introduced a global variable *FinalResult* is the necessity of having all information on the calculation available. A printout of the information contained in *FinalResult* for our calculation of Tang's problem can be obtained by calling

```
FinalResult // Flatten // LieTraditionalForm // TableForm
```

$$u_t - \frac{k\,u_{x,x}}{1 + u_x^2}$$
$$u_t \to \frac{k\,u_{x,x}}{1 + u_x^2}$$
$$(\xi_2)_u$$
$$(\xi_2)_x$$
$$-3\,(\xi_1)_t + k\,(\xi_1)_{u,u} + 2\,k\,(\xi_1)_{x,x} - 4\,k\,(\phi_1)_{x,u}$$
$$-2\,(\xi_1)_t + k\,(\xi_1)_{u,u} + k\,(\xi_1)_{x,x} - 2\,k\,(\phi_1)_{x,u}$$

```
-6 (ξ1)t + k (ξ1)u,u + 5 k (ξ1)x,x - 10 k (ϕ1)x,u
-3 (ξ1)t + 2 k (ξ1)u,u + k (ξ1)x,x - 2 k (ϕ1)x,u
-(ξ1)t + k (ξ1)x,x - 2 k (ϕ1)x,u
-6 (ξ1)t + 5 k (ξ1)u,u + k (ξ1)x,x - 2 k (ϕ1)x,u
6 (ϕ1)t + 10 k (ξ1)x,u - 5 k (ϕ1)u,u - k (ϕ1)x,x
(ϕ1)t - k (ϕ1)x,x
-3 (ϕ1)t - 4 k (ξ1)x,u + 2 k (ϕ1)u,u + k (ϕ1)x,x
6 (ϕ1)t + 2 k (ξ1)x,u - k (ϕ1)u,u - 5 k (ϕ1)x,x
-2 (ϕ1)t - 2 k (ξ1)x,u + k (ϕ1)u,u + k (ϕ1)x,x
-3 (ϕ1)t - 2 k (ξ1)x,u + k (ϕ1)u,u + 2 k (ϕ1)x,x
(ϕ1)t + 2 k (ξ1)x,u - k (ϕ1)u,u
6 (ξ1)x - 5 (ξ2)t + 4 (ϕ1)u
8 (ξ1)x - 5 (ξ2)t + 2 (ϕ1)u
4 (ξ1)x - 5 (ξ2)t + 6 (ϕ1)u
-2 (ξ1)x + (ξ2)t
2 (ξ1)x - 5 (ξ2)t + 8 (ϕ1)u
(ξ1)u + (ϕ1)x
(ξ1)u + (ϕ1)x
(ξ1)u + (ϕ1)x
-(ξ2)t + 2 (ϕ1)u
-(ξ1)t + k (ξ1)u,u
LieStructure
Metric
SymmGroup
```

The last four elements of the list *FinalResult* are empty or contain symbolic names which carry no information at this stage of the calculation. Another global variable of *MathLie* is called *Result2*, containing the information on the infinitesimals:

```
Result2
```

```
{{xi[2] → Function[{x, t, u}, k1 + 2 k5 t],
  phi[1] → Function[{x, t, u}, k2 + k5 u + k3 x],
  xi[1] → Function[{x, t, u}, k4 - k3 u + k5 x]},
 {}}
```

The global variable *Result2* contains the infinitesimals and the remaining equations of the determining equations not solved by the function PDESolve[]. The function PDESolve[] is the solver of the package *MathLie*. For the current equation, all determining equations are solved. Thus, the last element of *Result2* is empty. The results for the infinitesimals are given by substitution rules usable in other calculations. For example, an application of this kind of calculation is the reduction of the equation.

We select the subgroup of interest by setting a subgroup of the group constants to a numeric value and others to a symbolic value. The infinitesimals are thus reduced by

```
infi1 = {{xi[1][x, t, u[x, t]], xi[2][x, t, u[x, t]]},
    {phi[1][x, t, u[x, t]]}} /. Result2[[1]] /.
  {k1 → v, k2 → 1, k3 → 0, k4 → 1, k5 → 0, u[___] → u}
```

```
{{1, v}, {1}}
```

This special extraction of a subgroup is related to a moving wave solution. Tang's equation is thus invariant with respect to translations:

```
rtang = LieReduction[tang, {u}, {x, t}, infi1[[1]], infi1[[2]]];
LTF[Flatten[rtang]] /. {zeta1 → ζ1}
```

$$t - v\,x - \zeta_1 == 0$$
$$u - x - F_1 == 0$$
$$2\,F1_{\zeta_1} - 2\,v\,F1^2_{\zeta_1} + v^2\,F1^3_{\zeta_1} - k\,v^2\,F1_{\zeta_1,\zeta_1} == 0$$

The reduced equation is solved by

```
mwtang = DSolve[rtang[[3, 1]] == 0, F1, zeta1]
```

$$\left\{\text{Solve}\left[\frac{1}{2\,k}\left(-k\,v^2\,\text{ArcTan}[1 - v\,F1'[\#1]] + 2\,k\,\text{Log}\left[F1'[\#1]^{\frac{v^2}{2}}\,(2 - 2\,v\,F1'[\#1] + v^2\,F1'[\#1]^2)^{-\frac{v^2}{4}}\right] - 2\,\#1\right) == C[1],\ \{F1'[\#1]\}\right]\right\}$$

The result is an implicit representation of the solution entangling the similarity function *F1* in trigonometric and logarithmic relations.

Another reduction of the equation follows by the choice k3 = *c*:

```
infi1 = {{xi[1][x, t, u[x, t]], xi[2][x, t, u[x, t]]},
    {phi[1][x, t, u[x, t]]}} /.
  Result2[[1]] /. {k1 → 0, k2 → 0, k3 → c, k4 → 0, k5 → 0,
   u[___] → u}
```

```
{{-c u, 0}, {c x}}
```

The related reduction is

```
r1tang = LieReduction[tang, {u}, {x, t}, infi1[[1]], infi1[[2]];
LTF[Flatten[r1tang]] /. zeta1 → ζ1
```

$$t - \zeta_1 == 0$$

$$-\frac{u^2}{2} - \frac{x^2}{2} - F_1 == 0$$

$$-k + F1_{\zeta_1} == 0$$

Solving the reduced equation, we get the explicit representation for the similarity solution by

```
s1tang = DSolve[r1tang[[3, 1]] == 0, F1, zeta1]
```

```
{{F1 → (C[1] + k #1&)}}
```

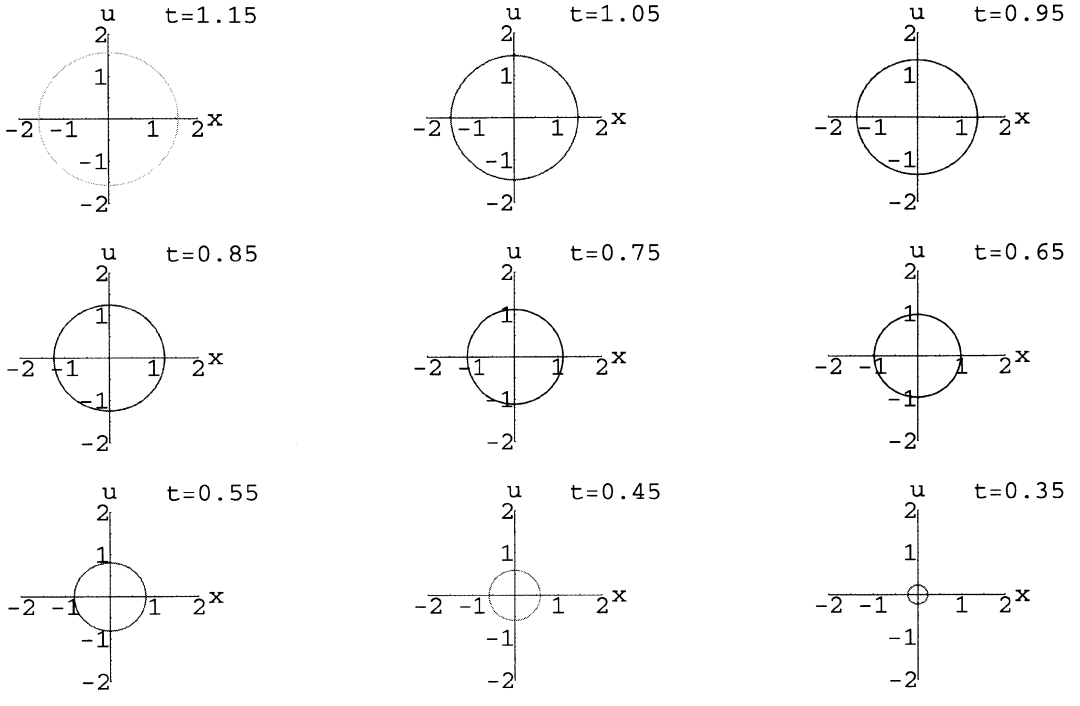
Inserting this solution into the similarity representation, we end up with an implicit representation of the solution for the field u:

```
sol1 = Flatten[r1tang[[2]] /. s1tang]
```

$$\left\{-\frac{u^2}{2} - \frac{x^2}{2} == k\,t + C[1]\right\}$$

This result illustrates that the circles in the (x, u)-plane are dependent on the time t. The radius of the circle depends on the actual time and on the signs of the constants $C[1]$ and k. If both constants are negative the radius of the circle increases to infinity if time t increases. If $C[1]$ is negative and $k > 0$, we get an upper limit for $t = |C[1]|/k$ for which the radius is real. However, if we set $C[1] > 0$, there exist no real radii in the (x, u)-plane. We demonstrate this behavior for $k = -3/2$ and $C[1] = 1/2$ for which a lower limit in t exists. The following animation shows the increase of the radius $-k\,t - C[1]$ by increasing the time t above the threshold of $t \geq 1/4$:

```
<< Graphics`ImplicitPlot`

Do[ImplicitPlot[
   u^2/2 + x^2/2 == -k t - C[1] /. {t → ti, k → -1.5, C[1] → 1/2},
   {x, -2, 2}, PlotRange → {{-2, 2}, {-2, 2}},
   PlotPoints → 35, PlotStyle → Hue[ti], PlotLabel →
    "                t=" <> ToString[ti], AxesLabel → {"x", "u"}],
  {ti, 1.5, 0.251, -.05}]
```



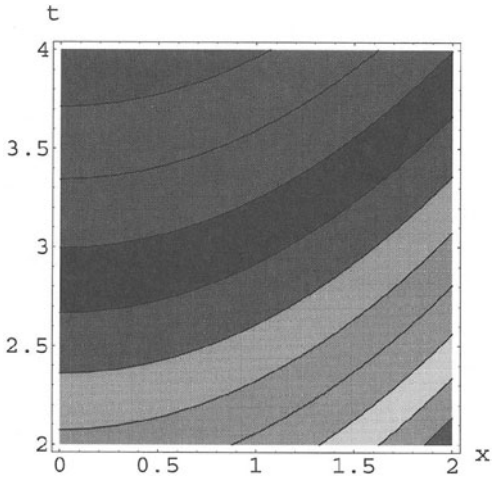
The solution of this quadratic equation in x and u yields the final representation of the solution of Tang's equation to be

```
sol11 = Solve[sol1[[1]], u]
```

$$\{\{u \to -\sqrt{-2\,k\,t - x^2 - 2\,C[1]}\},\ \{u \to \sqrt{-2\,k\,t - x^2 - 2\,C[1]}\}\}$$

containing the parameter k of the original equation and one constant of integration $C[1]$. The graphical representation of the second solution for parameters $k = -2$ and $C[1] = 1/2$ gives us the impression that the solution has a smooth behavior:

```
ContourPlot[u /. sol11[[2]] /. {k → -2, C[1] → 1/2},
 {x, 0, 2}, {t, 2, 4}, ColorFunction → Hue, Axes → True,
 AxesLabel → {"x", "t"}]
```



5.6.4 The Korteweg-de Vries Equation and Its Generalizations

The aim of this section is the determination of the point symmetries for different generalizations of the Korteweg-de Vries equation (gKdV). We first examine the original equation of Korteweg and de Vries (KdV). The generalizations of the KdV are based on additional terms, larger sets of independent coordinates, and changes in the non-linearity. The models examined are all taken from literature and describe different physical problems. At the end of this section, we collect the results in a table containing the equation and the symmetries

The Korteweg-de Vries equation is one of the prominent equations in non-linear physics which describes a highly regular behavior. The KdV equation originally derived by Korteweg and de Vries [1895] is used to describe shallow water waves in a narrow channel. The equation was originally designed to describe the experimental observations of John Scott Russell [1844] that a heap of water travels along the Edinburgh to Glasgow channel without change of form and with a constant velocity.

Today, the KdV equation is encountered in different physical systems such as in plasmas, in elastic strings, in lattice vibrations of a crystal at low temperatures, and in the description of a rotating liquid in a pipe. All these applications start from a more or less general physical model and end up in the KdV equation by considering a specific limit of the physical problem. In this sense, the KdV equation is universal. The universality is contained in the behavior that the dispersion of linear waves is counterbalanced by the non-linearity. The interaction of dispersion and non-linearity stabilizes the solution of the equation and results in the outstanding behavior of regularity. This kind of solution is called a soliton by Zabusky and Kruskal [1965]. A soliton is a special kind of localized wave. More generally, a soliton is a solution of a

non-linear equation or system which represents a wave of permanent shape, is localized, decaying or becoming constant at infinity, and may interact strongly with other solitons so that after the interaction, it retains its form (cf. Drazin [1983]).

The KdV equation is a non-linear PDE which reads in its standard form

$$u_t + 6\,u\,u_x + u_{x,x,x} = 0. \tag{5.48}$$

The field $u = u(x, t)$ describes the deviation from the mean water depth in the shallow water channel application. Our intention here is to show that the KdV equation not only allows solitons but also other kinds of solutions. To derive these types of solutions, we first need to know the infinitesimal transformation of the KdV equation. In *Mathematica*, the KdV equation is given by

```
KdV = ∂t u[x, t] + 6 u[x, t] ∂x u[x, t] + ∂x,x,x u[x, t] == 0;
KdV // LTF
```

$$u_t + 6\,u\,u_x + u_{x,x,x} == 0$$

The infinitesimals of the KdV follow by

```
InfinitesimalsKdV = Infinitesimals[KdV, u, {x, t}];
InfinitesimalsKdV // LTF
```

$$\phi_1 == \frac{k2}{6} - 2\,k4\,u$$
$$\xi_1 == k3 + k2\,t + k4\,x$$
$$\xi_2 == k1 + 3\,k4\,t$$

The result is a four-dimensional finite group containing translation and scaling transformations.

The second model we examine is the cylindrical KdV equation. This kind of equation was discussed by Calogero and Degasperis [1978] in connection with the spectral transform method. The cylindrical form of the KdV equation reads

$$u_t + 6\,u\,u_x + u_{x,x,x} + \frac{1}{2t}\,u = 0. \tag{5.49}$$

The Infinitesimals for this equation follows from

```
infCylibdicylKdV =
 Infinitesimals[∂t u[x, t] + 6 u[x, t] ∂x u[x, t] +
    ∂x,x,x u[x, t] + 1/(2 t) u[x, t] == 0, u, {x, t}];
infCylibdicylKdV // LTF
```

$$\phi_1 == \frac{4\,k3 - 16\,k1\,\sqrt{t}\,u - 24\,k2\,t\,u + k2\,x}{24\,\sqrt{t}}$$

$$\xi_1 == k4 + 2\,k3\,\sqrt{t} + \frac{1}{6}\,(2\,k1 + 3\,k2\,\sqrt{t}\,)\,x$$

$$\xi_2 == (k1 + k2\,\sqrt{t}\,)\,t$$

We find a four-dimensional finite symmetry group containing translations, scaling, and two special symmetry transformations.

Another KdV equation closely related to the cylindrical KdV equation is the spherical KdV equation given by

$$u_t + 6\,u\,u_x + u_{x,x,x} + \frac{1}{t}\,u = 0. \tag{5.50}$$

The infinitesimals of this equation are

```
infsphericalKdV =
 Infinitesimals[∂t u[x, t] + 6 u[x, t] ∂x u[x, t] +
     ∂x,x,x u[x, t] + 1/t u[x, t] == 0, u, {x, t}];
infsphericalKdV // LTF
```

$$\phi_1 == \frac{k3}{6\,t} - 2\,k2\,u$$

$$\xi_1 == k1 + k4 + k2\,x + k3\,\mathrm{Log}[t]$$

$$\xi_2 == 3\,k2\,t$$

The symmetry group is a three-dimensional finite group. Compared with the cylindrical KdV in which the symmetries changed drastically, there is no longer a rational exponent of t present but logarithmic dependencies in t.

Another kind of generalization of the KdV was given by Ko and Kuehl [1978] in their discussion of ion acoustic solitons in a non-uniform plasma. The two authors assumed that the coefficients of the non-linear and dispersive term depend on time. The KdV for such a slowly varying medium is given by

$$u_t + \alpha(\tau)\,u\,u_x + \beta(\tau)\,u_{x,x,x} = 0 \text{ and } \alpha,\ \beta > 0, \tag{5.51}$$

where the coefficients α and β are arbitrary positive functions of a slow time variable $\tau = \epsilon t$. The infinitesimals for equation (5.51) are given by

```
infVaryingKdV = Infinitesimals[
  ∂t u[x, t] + α[ϵ t] u[x, t] ∂x u[x, t] + β[ϵ t] ∂x,x,x u[x, t] == 0,
  u, {x, t}, {ϵ}];
infVaryingKdV //. TraditionalLieForm /.
   {Rule → Equal, HoldPattern[Function[x_, y_]] → y} // Sort //
 TableForm
```

ξ_1 == k2 + k1 $\int_0^t \alpha$[DSolve`t $\epsilon$] dDSolve`t

ξ_2 == 0

ϕ_1 == k1

The infinitesimals of equation (5.51) show that a discrete group of dimension two exists. Surprisingly, this symmetry group depends on an integral over the coefficient of the non-linearity $\int_0^t \alpha(\epsilon t')\, dt'$ and not on β. A similar equation extended by linear derivatives was discussed by Chan and Li [1994] in connection with the non-standard dynamics of solitons (oscillating or standing). The equation discussed by Chan and Li in connection with the inverse scattering theory is given by

$$u_t + (6u^2 u_x + u_{x,x,x})\, k_0(t) - (x u_x + u)\, h(t) + u_x\, k_1(t) = 0, \tag{5.52}$$

where k_0, k_1, and h are continuous functions of t. Equation (5.52) reduces to the modified KdV equation when $k_0 = 1$ and $k_1 = h = 0$. The symmetries of equation (5.52) are

```
infNonPropagatingKdV = Infinitesimals[
  ∂t u[x, t] + k0[t] (6 u[x, t]^2 ∂x u[x, t] + ∂x,x,x u[x, t]) -
    h[t] (x ∂x u[x, t] + u[x, t]) + k1[t] ∂x u[x, t] == 0, u, {x, t}];
infNonPropagatingKdV // LTF
```

$$3\, h\, k_0\, \mathcal{F}_2 + k_1\, (\mathcal{F}_1\, (k_0)_t - 2\, k_0\, (\mathcal{F}_1)_t) + 3\, k_0\, (-\mathcal{F}_1\, (k_1)_t + (\mathcal{F}_2)_t) == 0$$

$$-\mathcal{F}_1\, (k_0)_t^2 + k_0\, ((k_0)_t\, (\mathcal{F}_1)_t + \mathcal{F}_1\, (k_0)_{t,t}) + k_0^2\, (3\, h_t\, \mathcal{F}_1 + 3\, h\, (\mathcal{F}_1)_t + (\mathcal{F}_1)_{t,t}) == 0$$

$$\xi_1 == \mathcal{F}_2 + \frac{x\, (\mathcal{F}_1\, (k_0)_t + k_0\, (\mathcal{F}_1)_t)}{3\, k_0}$$

$$\xi_2 == \mathcal{F}_1$$

$$\phi_1 == -\frac{u\, (\mathcal{F}_1\, (k_0)_t + k_0\, (\mathcal{F}_1)_t)}{3\, k_0}$$

The infinitesimals of Chan's equation define an infinite dimensional symmetry group. The symmetries are defined by the two arbitrary functions *free[1]* and *free[2]* and the coefficients k_0, k_1, and h. One of the arbitrary functions, *free[1]*, occurs in the infinitesimals, whereas the other is contained in the remaining determining equations. Equation (5.52) demonstrates that *MathLie* is capable of handling equations with general analytic coefficients.

A related model to Chan's was discussed by Fung and Au [1982] to bridge the solutions and vacuum states of the KdV equation and the equation

$$u_t - 6u^2 u_x + u_{x,x,x} + 6\lambda u_x = 0 \tag{5.53}$$

with λ a real parameter. Equation (5.53), despite of the sign in the non-linearity, follows from Chan's model by setting $h = 0$, $k_0 = 1$, and $k_1 = 6\lambda$. This λ-dependent model allows the infinitesimals

```
infλKdV = Infinitesimals[∂t u[x, t] - 6 u[x, t]^2 ∂x u[x, t] +
    ∂x,x,x u[x, t] + 6 λ ∂x u[x, t] == 0, u, {x, t}, {λ}];
infλKdV // LTF
```

ϕ_1 == -k3 u
ξ_1 == k2 + k3 (x + 12 t λ)
ξ_2 == k1 + 3 k3 t

We realize that the restrictions on the Chan equation (5.52) lead to a three-dimensional symmetry group allowing translations and scalings.

In connection with traveling wave solutions, Yang [1994] examined the generalized KdV equation

$$u_t + \beta u^{\alpha} u_x + u_{x,x,x} = 0, \tag{5.54}$$

where α and β are real numbers. This equation reduces to the original KdV equation and the modified KdV equation for $\beta = 6$ and $\alpha = 1, 2$, respectively. Other combinations of α and β are known and discussed in connection with ion acoustic waves in cold-ion and multi-component plasma (cf. Yang [1994]). The infinitesimals of this gKdV equation are

```
infGKdV = Infinitesimals[
  ∂t u[x, t] + β u[x, t]^α ∂x u[x, t] + ∂x,x,x u[x, t] == 0,
  u, {x, t}, {α, β}];
infGKdV // LTF
```

$$\phi_1 == -\frac{2\,k3\,u}{3\,\alpha}$$
$$\xi_1 == k1 + \frac{k3\,x}{3}$$
ξ_2 == k2 + k3 t

a set of transformations depending on three group parameters and on the exponent α. We also observe that the symmetry group does not depend on the non-linearity factor β.

Discussing weakly non-linear, long-wavelength waves propagating on the surface of an incompressible, inviscid, irrotational fluid, the following perturbed KdV equation arises:

$$u_t + 6uu_x + u_{x,x,x} + \epsilon(-\alpha u^2 u_x + \beta u u_{x,x} + \gamma u_x u_{x,x} + \delta u_{x,x,x,x,x}) = 0. \tag{5.55}$$

Here, $\alpha, \beta, \gamma, \delta$, and ϵ are constant parameters. ϵ is the amplitude-to-depth ratio and is assumed to be much smaller than unity; $|\epsilon| \ll 1$. This kind of KdV equation was examined by Alexeyev [1994] in connection with Bäcklund transformations and by Porsezian and Lakshmanan [1993] in connection with higher-order integrable models. The infinitesimals of this higher-order PDE follow from

```
infPKdV =
 Infinitesimals[∂t u[x, t] + 6 u[x, t] ∂x u[x, t] + ∂x,x,x u[x, t] +
    ϵ (-α u[x, t]^2 ∂x u[x, t] + β u[x, t] ∂x,x u[x, t] +
       γ ∂x u[x, t] ∂x,x u[x, t] + δ ∂{x,5} u[x, t]) == 0,
  u, {x, t}, {α, β, γ, δ, ϵ}];
infPKdV // LTF
```

ϕ_1 == 0
ξ_1 == k1
ξ_2 == k2

This equation allows only a two-dimensional finite symmetry group representing translations in the independent variables.

A model extending the KdV equation by an additional dispersive term of second order is discussed by Parkes [1994]. This so-called Korteweg-deVries-Burgers (KdVB) equation reads

$$u_t + uu_x + \mu u_{x,x,x} - \nu u_{x,x} = 0, \tag{5.56}$$

where μ and ν are real constants. The infinitesimals of (5.56) are

```
infKdVB = Infinitesimals[∂t u[x, t] + u[x, t] ∂x u[x, t] +
    μ ∂x,x,x u[x, t] - ν ∂x,x u[x, t] == 0, u, {x, t}, {μ, ν}];
infKdVB // LTF
```

ϕ_1 == k3
ξ_1 == k2 + k3 t
ξ_2 == k1

The resulting symmetries of the KdVB equation form a three-dimensional group independent of the model parameters. In connection with two-dimensional spatial solitons, Parkes [1994] and Ma [1993] discussed a two-dimensional KdVB equation of the form

$$(u_t + u\,u_x + \mu u_{x,x,x} - \nu u_{x,x})_x + \sigma u_{y,y} = 0. \tag{5.57}$$

The infinitesimals of this (2 + 1)-dimensional equation follow by

```
inf2KdVB = Infinitesimals[
  ∂x (∂t u[x, y, t] + u[x, y, t] ∂x u[x, y, t] + μ ∂x,x,x u[x, y, t]) +
    σ ∂y,y u[x, y, t] == 0,
  u, {x, y, t}, {μ, ν, σ}];
inf2KdVB // LTF
```

$$\phi_1 == -\frac{4\,u\,\sigma\,(\mathcal{F}_1)_t - 6\,\sigma\,(\mathcal{F}_3)_t - 2\,x\,\sigma\,(\mathcal{F}_1)_{t,t} + 3\,y\,(\mathcal{F}_2)_{t,t} + y^2\,(\mathcal{F}_1)_{t,t,t}}{6\,\sigma}$$

$$\xi_1 == \frac{6\,\sigma\,\mathcal{F}_3 + 2\,x\,\sigma\,(\mathcal{F}_1)_t - y\,(3\,(\mathcal{F}_2)_t + y\,(\mathcal{F}_1)_{t,t})}{6\,\sigma}$$

$$\xi_2 == \mathcal{F}_2 + \frac{2}{3}\,y\,(\mathcal{F}_1)_t$$

$$\xi_3 == \mathcal{F}_1$$

By adding an independent variable to the field u, the symmetry group of the one-dimensional (1D)-KdVB equation makes a transition to an infinite dimensional symmetry group. The two arbitrary functions $free[1] = \mathcal{F}_1$ and $free[2] = \mathcal{F}_2$ are not restricted by any equation. We note that the symmetry group depends on the parameter σ.

A last example demonstrates the application to a complex field. In this case, the KdV equation finds its application in the asymptotic investigation of electrostatic waves of a magnetized plasma. This problem of elastostatic waves is discussed by Mohammad and Can [1995]. The complex-valued KdV equation is given by

$$w_t + \alpha((|w|)^2 w)_x + \beta w_{x,x,x} = 0, \tag{5.58}$$

where w is the complex field amplitude $w = u + iv$, and α and β are real parameters. The representation in real field variables u and v is given by

$$u_t + \alpha(u^3)_x + \beta u_{x,x,x} + ((\alpha(uv^2))_x = 0, \tag{5.59}$$

$$v_t + \alpha(v^3)_x + \beta v_{x,x,x} + ((\alpha(vu^2))_x = 0. \tag{5.60}$$

The infinitesimals of this system of equations are

```
infCKdV = Infinitesimals[{∂t u[x, t] + α ∂x (u[x, t]^3) +
      β ∂x,x,x u[x, t] + α ∂x (u[x, t] v[x, t]^2) == 0,
   ∂t v[x, t] + α ∂x (v[x, t]^3) +
      β ∂x,x,x v[x, t] + α ∂x (v[x, t] u[x, t]^2) == 0},
  {u, v}, {x, t}, {α, β}];
infCKdV // LTF
```

$$\phi_1 == -\frac{k4\,u}{3} + k1\,v$$

$$\phi_2 == \frac{1}{3}\,(-3\,k1\,u - k4\,v)$$

$$\xi_1 == k2 + \frac{k4\,x}{3}$$

$$\xi_2 == k3 + k4\,t$$

The symmetry group is a four-dimensional discrete group containing translations, a rotation, and a scaling transformation.

All the calculations carried out above are collected in the following table containing the equation, the symmetries, and the dimension of the symmetry group.

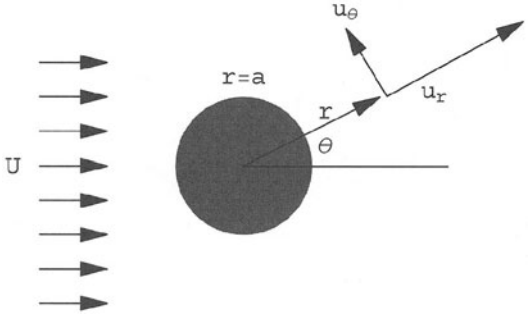
KdV type equation	Infinitesimals	Dim
$u_t + u\,u_x + u_{x,x,x} == 0$	$\phi_1 == \frac{k2}{6} - 2\,k4\,u$ $\xi_1 == k3 + k2\,t + k4\,x$ $\xi_2 == k1 + 3\,k4\,t$	4
$\frac{u}{2t} + u_t + 6\,u\,u_x + u_{x,x,x} == 0$	$\phi_1 == \frac{4\,k3 - 16\,k1\,\sqrt{t}\,u - 24\,k2\,t\,u + k2\,x}{24\,\sqrt{t}}$ $\xi_1 == k4 + 2\,k3\,\sqrt{t} + \frac{1}{6}\,(2\,k1 + 3\,k2\,\sqrt{t})\,x$ $\xi_2 == (k1 + k2\,\sqrt{t})\,t$	4
$\frac{u}{t} + u_t + 6\,u\,u_x + u_{x,x,x} == 0$	$\phi_1 == \frac{k3}{6\,t} - 2\,k2\,u$ $\xi_1 == k1 + k4 + k2\,x + k3\,\mathrm{Log}[t]$ $\xi_2 == 3\,k2\,t$	3
$u_t + u\,\alpha\,u_x + \beta\,u_{x,x,x} == 0$	$\xi_1 == k2 + k1\int_0^t \alpha[\epsilon\,t']\,dt'$ $\xi_2 == 0$ $\phi_1 == k1$	2
$u_t + k_1\,u_x - h\,(u + x\,u_x) + k_0\,(6\,u^2\,u_x + u_{x,x,x}) == 0$	$3\,h\,k_0\,\mathcal{F}_2 + k_1\,(\mathcal{F}_1\,(k_0)_t - 2\,k_0\,(\mathcal{F}_1)_t) + 3\,k_0\,(-\mathcal{F}_1\,(k_1)_t + (\mathcal{F}_2)_t) == 0$ $-\mathcal{F}_1\,(k_0)_t^2 + k_0\,((k_0)_t\,(\mathcal{F}_1)_t + \mathcal{F}_1\,(k_0)_{t,t}) + k_0^2\,(3\,h_t\,\mathcal{F}_1 + 3\,h\,(\mathcal{F}_1)_t + (\mathcal{F}_1)_{t,t}) == 0$ $\xi_1 == \mathcal{F}_2 + \frac{x\,(\mathcal{F}_1\,(k_0)_t + k_0\,(\mathcal{F}_1)_t)}{3\,k_0}$ $\xi_2 == \mathcal{F}_1$ $\phi_1 == -\frac{u\,(\mathcal{F}_1\,(k_0)_t + k_0\,(\mathcal{F}_1)_t)}{3\,k_0}$	
$u_t - 6\,u^2\,u_x + 6\,\lambda\,u_x + u_{x,x,x} == 0$	$\phi_1 == -k3\,u$ $\xi_1 == k2 + k3\,(x + 12\,t\,\lambda)$ $\xi_2 == k1 + 3\,k3\,t$	3
$u_t + u^\alpha\,\beta\,u_x + u_{x,x,x} == 0$	$\phi_1 == -\frac{2\,k3\,u}{3\,\alpha}$ $\xi_1 == k1 + \frac{k3\,x}{3}$ $\xi_2 == k2 + k3\,t$	3
$u_t + 6\,u\,u_x + u_{x,x,x} + \epsilon\,(-u^2\,\alpha\,u_x + u\,\beta\,u_{x,x} + \gamma\,u_x\,u_{x,x} + \delta\,u_{x,x,x,x,x}) == 0$	$\phi_1 == 0$ $\xi_1 == k1$ $\xi_2 == k2$	2
$u_t + u\,u_x - \nu\,u_{x,x} + \mu\,u_{x,x,x} == 0$	$\phi_1 == k3$ $\xi_1 == k2 + k3\,t$ $\xi_2 == k1$	3
$u_x^2 + u_{x,t} + u\,u_{x,x} + \sigma\,u_{y,y} + \mu\,u_{x,x,x,x} == 0$	$\phi_1 == -\frac{4\,u\,\sigma\,(\mathcal{F}_1)_t - 6\,\sigma\,(\mathcal{F}_3)_t - 2\,x\,\sigma\,(\mathcal{F}_1)_{t,t} + 3\,y\,(\mathcal{F}_2)_{t,t} + y^2\,(\mathcal{F}_1)_{t,t,t}}{6\,\sigma}$ $\xi_1 == \frac{6\,\sigma\,\mathcal{F}_3 + 2\,x\,\sigma\,(\mathcal{F}_1)_t - y\,(3\,(\mathcal{F}_2)_t + y\,(\mathcal{F}_1)_{t,t})}{6\,\sigma}$ $\xi_2 == \mathcal{F}_2 + \frac{2}{3}\,y\,(\mathcal{F}_1)_t$ $\xi_3 == \mathcal{F}_1$	
$u_t + 3\,u^2\,\alpha\,u_x + \alpha\,(v^2\,u_x + 2\,u\,v\,v_x) + \beta\,u_{x,x,x} == 0$ $v_t + 3\,v^2\,\alpha\,v_x + \alpha\,(2\,u\,v\,u_x + u^2\,v_x) + \beta\,v_{x,x,x} == 0$	$\phi_1 == -\frac{k4\,u}{3} + k1\,v$ $\phi_2 == \frac{1}{3}\,(-3\,k1\,u - k4\,v)$ $\xi_1 == k2 + \frac{k4\,x}{3}$ $\xi_2 == k3 + k4\,t$	4

Table 5.1

We note that the symmetries of the examined KdV-type equations can be quite different in their form. The order of the symmetry groups range from two to infinity and the symmetries are far from being always polynomial. The collection of these symmetry groups also illustrate that the access to symmetries is very simple and that a large number of models can be examined within a short period. This is the real power of computer algebra in connection with symmetry analysis that the information is available in split seconds and that variations of the equation can be checked very quickly.

5.6.5 Stokes' Solution of the Creeping Flow

Let us consider the creeping motion of a fluid stream of speed U around a solid sphere of radius a. The physical situation is shown in the following figure.



Coordinate system

It is convenient to use spherical coordinates (r, θ). We choose the origin of θ in such a way that $\theta = 0$ defines the direction of U. The velocity components u_r and u_θ are related to the Stokes stream function Ψ in spherical coordinates by the relations

```
StreamFunction = {ur → Function[{r, θ}, ∂θ Ψ[r, θ] / (r² Sin[θ])],
  uth → Function[{r, θ}, -∂r Ψ[r, θ] / (r Sin[θ])]}
```

```
{ur → Function[{r, θ}, ∂θ Ψ[r, θ] / (r² Sin[θ])],
 uth → Function[{r, θ}, -∂r Ψ[r, θ] / (r Sin[θ])]}
```

In the case of creep flows, the motion has a Reynolds number much less than unity. A consequence of this fact is that we can neglect acceleration terms in the momentum

equation. The momentum equation for the stream function is derived by applying the Laplacian in spherical coordinates:

```
Operator[expr_] := ∂{r,2} expr + (∂{θ,2} expr)/r^2 - (Cot[θ] ∂θ expr)/r^2
```

Applying this operator twice to the stream function Ψ, we find the governing equation

```
Momentum = Operator[Operator[Ψ[r, θ]]]; Momentum // LTF
```

$$-\frac{6\,\mathrm{Cot}[\theta]\,\Psi_{\theta}}{r^4}+\frac{4\,\mathrm{Cot}[\theta]\,\Psi_{r,\theta}}{r^3}+\frac{6\,\Psi_{\theta,\theta}}{r^4}-\frac{\mathrm{Cot}[\theta]\,\Psi_{r,r,\theta}}{r^2}-\frac{4\,\Psi_{r,\theta,\theta}}{r^3}-$$
$$\frac{\mathrm{Cot}[\theta]\left(\frac{\mathrm{Csc}[\theta]^2\,\Psi_{\theta}}{r^2}-\frac{\mathrm{Cot}[\theta]\,\Psi_{\theta,\theta}}{r^2}+\Psi_{r,r,\theta}+\frac{\Psi_{\theta,\theta,\theta}}{r^2}\right)}{r^2}+$$
$$\Psi_{r,r,r,r}+\frac{\Psi_{r,r,\theta,\theta}}{r^2}+\frac{1}{r^2}\left(-\frac{2\,\mathrm{Cot}[\theta]\,\mathrm{Csc}[\theta]^2\,\Psi_{\theta}}{r^2}+\right.$$
$$\left.\frac{2\,\mathrm{Csc}[\theta]^2\,\Psi_{\theta,\theta}}{r^2}-\frac{\mathrm{Cot}[\theta]\,\Psi_{\theta,\theta,\theta}}{r^2}+\Psi_{r,r,\theta,\theta}+\frac{\Psi_{\theta,\theta,\theta,\theta}}{r^2}\right)==$$
$$0$$

The resulting equation is a fourth-order linear partial differential equation in spherical coordinates. The equation, although linear, contains a lot of analytic coefficients. This linear equation is an example to test the reliability of *MathLie*. Extending our database by creating another file called *stokes.dgl* containing the information on this equation helps us to derive the symmetries of the equation:

```
LieEquations["stokes.dgl", {Momentum}, {Ψ}, {r, θ}, {},
 {"Creeping flow for an immersed sphere"},
 {{"G.G. Stokes"},
  {"Trans. Camb. Phil. Soc. 9, 8-106, (1851)"}},
 SubstitutionRules → {∂{r,4} Ψ[r, θ]}]
```

The Lie point symmetries of the momentum equation are derived by

```
LieSolve["stokes.dgl"] // LTF
```

$$-3\,\mathrm{Cot}[\theta]\,(2+\mathrm{Csc}[\theta]^2)\,(\mathcal{F}_1)_{\theta}+$$
$$4\,r\,\mathrm{Cot}[\theta]\,(\mathcal{F}_1)_{r,\theta}+(6+\mathrm{Cot}[\theta]^2+2\,\mathrm{Csc}[\theta]^2)\,(\mathcal{F}_1)_{\theta,\theta}-$$
$$2\,r^2\,\mathrm{Cot}[\theta]\,(\mathcal{F}_1)_{r,r,\theta}-4\,r\,(\mathcal{F}_1)_{r,\theta,\theta}-2\,\mathrm{Cot}[\theta]\,(\mathcal{F}_1)_{\theta,\theta,\theta}+$$
$$r^4\,(\mathcal{F}_1)_{r,r,r,r}+2\,r^2\,(\mathcal{F}_1)_{r,r,\theta,\theta}+(\mathcal{F}_1)_{\theta,\theta,\theta,\theta}==$$
$$0$$
$$\xi_2 == 0$$
$$\xi_1 == k2\,r$$
$$\phi_1 == k1\,\Psi+\mathcal{F}_1$$

The analysis shows that Stokes' model of creeping flow owns an infinite symmetry as expected for linear equations. The continuous part of the symmetry group is determined by the arbitrary function $\mathcal{F}_1$ = *free*[1][r, θ]. The function $\mathcal{F}_1$ has to satisfy the original linear fourth-order PDE. The two constants *k1* and *k2* are the determining elements of the finite group representing a scaling symmetry of the momentum equation. Knowing that the momentum equation allows only a scaling symmetry, we can use this information to reduce the PDE to an ODE. The application of the function LieReduction[] delivers for the scaling group *k1=1* and *k2=*α an ordinary differential equation of fourth order. Unfortunately, this ODE is not solvable by DSolve[].

```
rmoment = LieReduction[{Momentum}, {Ψ}, {r, θ}, {r, 0}, {α Ψ}];
LTF[Flatten[rmoment]] /. zeta1 → ζ1
```

$$\theta - \zeta_1 == 0$$

$$r^{-\alpha}\,\Psi - F_1 == 0$$

$$\begin{aligned} r^{\alpha}\,(&-6\,\alpha\,F_1 + 11\,\alpha^2\,F_1 - 6\,\alpha^3\,F_1 + \\ &\alpha^4\,F_1 - 6\,\mathrm{Cot}[\zeta_1]\,(F_1)_{\zeta_1} + 6\,\alpha\,\mathrm{Cot}[\zeta_1]\,(F_1)_{\zeta_1} - \\ &2\,\alpha^2\,\mathrm{Cot}[\zeta_1]\,(F_1)_{\zeta_1} - 3\,\mathrm{Cot}[\zeta_1]\,\mathrm{Csc}[\zeta_1]^2\,(F_1)_{\zeta_1} + 6\,(F_1)_{\zeta_1,\zeta_1} - \\ &6\,\alpha\,(F_1)_{\zeta_1,\zeta_1} + 2\,\alpha^2\,(F_1)_{\zeta_1,\zeta_1} + \mathrm{Cot}[\zeta_1]^2\,(F_1)_{\zeta_1,\zeta_1} + \\ &2\,\mathrm{Csc}[\zeta_1]^2\,(F_1)_{\zeta_1,\zeta_1} - 2\,\mathrm{Cot}[\zeta_1]\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} + (F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1}) == \\ 0 \end{aligned}$$

A closer look at the above result reveals that the stream function Ψ can be represented by a product of a radial component and a function containing the angular part:

```
stream = Ψ → Function[{r, θ}, r^α g[θ]]
```

$$\Psi \to \mathrm{Function}[\{r, \theta\}, r^{\alpha}\,g[\theta]]$$

We replaced F1 in the similarity solution by g. Inserting this result into the original equation, we get

```
mom1 = Expand[Momentum /. stream / r^(-4+α)]; mom1 // LTF
```

$$\begin{aligned} &-6\,g\,\alpha + 11\,g\,\alpha^2 - 6\,g\,\alpha^3 + g\,\alpha^4 - 6\,\mathrm{Cot}[\theta]\,g_{\theta} + 6\,\alpha\,\mathrm{Cot}[\theta]\,g_{\theta} - \\ &2\,\alpha^2\,\mathrm{Cot}[\theta]\,g_{\theta} - 3\,\mathrm{Cot}[\theta]\,\mathrm{Csc}[\theta]^2\,g_{\theta} + 6\,g_{\theta,\theta} - 6\,\alpha\,g_{\theta,\theta} + 2\,\alpha^2\,g_{\theta,\theta} + \\ &\mathrm{Cot}[\theta]^2\,g_{\theta,\theta} + 2\,\mathrm{Csc}[\theta]^2\,g_{\theta,\theta} - 2\,\mathrm{Cot}[\theta]\,g_{\theta,\theta,\theta} + g_{\theta,\theta,\theta,\theta} == \\ &0 \end{aligned}$$

which is an equation in r and θ. The remaining equation for g is a linear ODE of fourth order. This type of equation is equivalent with the equation gained by LieReduction[]. Since we already noted that the equation is not solved by DSolve[], we try an ansatz for the angular part $g(\theta)$ by

```
sub = g → Function[θ, Sin[θ]^2]
```

```
g → Function[θ, Sin[θ]^2]
```

which reduces the determining equation for g to a polynomial in α of fourth order:

```
pol = Simplify[mom1 /. sub]
```

```
(-8 + 6 α + 7 α^2 - 6 α^3 + α^4) Sin[θ]^2
```

```
pol1 = pol / Sin[θ]^2
```

```
-8 + 6 α + 7 α^2 - 6 α^3 + α^4
```

The solutions for the exponent α are found by solving this fourth-order polynomial

```
exponents = Solve[pol1 == 0, α]
```

```
{{α → -1}, {α → 1}, {α → 2}, {α → 4}}
```

This list is the basis for a combination of the radial components

```
radialpart = r^α /. exponents
```

```
{1/r, r, r^2, r^4}
```

Since the four solutions for α are independent of each other, a linear combination of the radial parts provides

```
f = Plus @@ (Table[c[i], {i, 1, 4}] radialpart)
```

```
c[1]/r + r c[2] + r^2 c[3] + r^4 c[4]
```

The final representation for the stream function is thus

```
StreamF = f Sin[θ]^2
```

```
(c[1]/r + r c[2] + r^2 c[3] + r^4 c[4]) Sin[θ]^2
```

Up to now, we did not consider any boundary conditions for the problem. As one boundary condition, we have to assume that the derivatives of the stream function with respect to r and θ vanish at the surface of the sphere. This results in

```
surf1 = (∂_r StreamF /. r → a) == 0
```

```
(-c[1]/a^2 + c[2] + 2 a c[3] + 4 a^3 c[4]) Sin[θ]^2 == 0
```

```
surf2 = (∂θ StreamF /. r → a) == 0
```

$$2\left(\frac{c[1]}{a} + a\,c[2] + a^2\,c[3] + a^4\,c[4]\right)\text{Cos}[\theta]\,\text{Sin}[\theta] == 0$$

These two non-slip conditions for the surface are sufficient to determine two of the four integration constants *c[i]*:

```
iconstants = Solve[{surf1, surf2}, {c[1], c[2], c[3], c[4]}]
```

```
Solve::svars :
 Equations may not give solutions for all "solve" variables.
```

$$\left\{\left\{c[1] \to \frac{1}{2}a^3\,c[3] + \frac{3}{2}a^5\,c[4],\ c[2] \to -\frac{3}{2}a\,c[3] - \frac{5}{2}a^3\,c[4]\right\}\right\}$$

The representation of the stream function reduces thus to

```
StreamF = Simplify[StreamF /. iconstants][[1]]
```

$$\frac{1}{2r}\,((a-r)^2\,(3a^3\,c[4] + 6a^2\,r\,c[4] + 2r\,(c[3] + r^2\,c[4]) + a\,(c[3] + 4r^2\,c[4]))\,\text{Sin}[\theta]^2)$$

containing two arbitrary constants *c*[3] and *c*[4]. If we consider the stream function for large values of *r*, we observe that the dominant changes result from the r^4 and r^2 terms in the solution:

```
Series[StreamF, {r,  , 2}]
```

$$\frac{c[4]\,\text{Sin}[\theta]^2}{\left(\frac{1}{r}\right)^4} + \frac{1}{O\left[\frac{1}{r}\right]^2}$$

If we assume that the radial and angular components of the velocity are finite for large *r*, we have to set *c*[4] = *0* and *c*[3] = *U*/2.

```
StreamF = StreamF /. {c[4] → 0, c[3] → U/2}
```

$$\frac{(a-r)^2\,(a+2r)\,U\,\text{Sin}[\theta]^2}{4r}$$

The stream function Ψ is thus defined by

```
Ψ[r_, θ_, a_, U_] := 1/4 Sin[θ]^2 a^2 U (a/r - 3 r/a + 2 r^2/a^2)
```

This representation in spherical coordinates can be used to display the stream function in a Cartesian coordinate system. To this end, we have to transform the coordinates by

```
coordTrafo = {r → Sqrt[x^2 + y^2], θ → ArcTan[x, y]}
```

```
{r → Sqrt[x^2 + y^2], θ → ArcTan[x, y]}
```

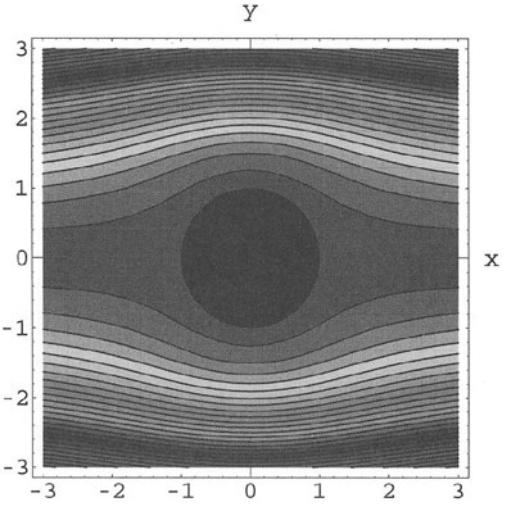
To suppress the singularities at $x=0$ and $y=0$, we introduce the condition

```
ps = If[x == 0 && y == 0, 0, Ψ[r, θ, 1, 1] /. coordTrafo]
```

```
If[x == 0 && y == 0, 0, Ψ[r, θ, 1, 1] /. coordTrafo]
```

The representation of the stream function in x and y is thus given by

```
Show[{ContourPlot[ps, {x, -3, 3}, {y, -3, 3},
    PlotPoints → 25,
    ColorFunction → Hue, AxesLabel → {"x", "y"}, Axes → True,
    Contours → 25, DisplayFunction → Identity],
  Graphics[Disk[{0, 0}, 1]]}, AspectRatio → Automatic,
 DisplayFunction → $DisplayFunction]
```



The black disk in the center of the picture represents the sphere. The different shadings outline the different strengths of the stream function around the disk.

The velocity components follow immediately by using the definitions

$$\texttt{ur[r\_, θ\_, a\_, U\_] := } \frac{\partial_\theta \Psi[r, \theta, a, U]}{r^2 \sin[\theta]}$$

$$\texttt{uth[r\_, θ\_, a\_, U\_] := } -\frac{\partial_r \Psi[r, \theta, a, U]}{r \sin[\theta]}$$

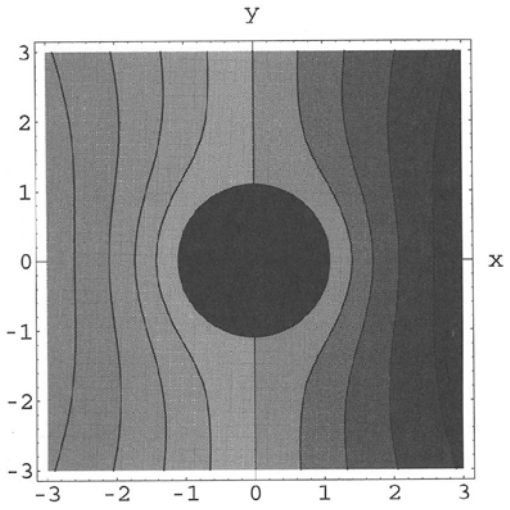
A representation of the radial component of the velocity in Cartesian coordinates follow by

```
u1 = ur[r, θ, 1, 1] /. coordTrafo
```

$$\frac{\left(\frac{1}{\sqrt{x^2+y^2}} - 3\sqrt{x^2+y^2} + 2\,(x^2+y^2)\right)\,\text{Cos}[\text{ArcTan}[x, y]]}{2\,(x^2+y^2)}$$

The graphical representation in a contour plot shows the orthogonal orientation of the velocity field u_r to the stream function:

```
Show[
 {ContourPlot[If[x == 0 && y == 0, 0, u1], {x, -3, 3}, {y, -3, 3},
   PlotPoints → 25, ColorFunction → Hue, AxesLabel → {"x", "y"},
   Axes → True, Contours → 25, DisplayFunction → Identity],
  Graphics[Disk[{0, 0}, 1.1]]}, AspectRatio → Automatic,
 DisplayFunction → $DisplayFunction]
```



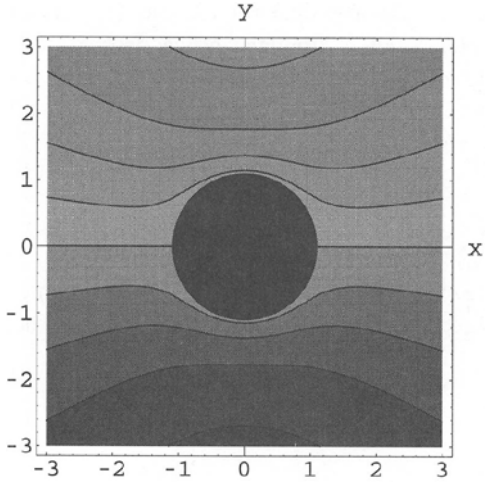
The related representation of the angular part of the velocity field is given by

```
u2 = uth[r, θ, 1, 1] /. coordTrafo
```

$$-\frac{\left(-3 - \frac{1}{x^2+y^2} + 4\sqrt{x^2+y^2}\right)\,\text{Sin}[\text{ArcTan}[x, y]]}{4\sqrt{x^2+y^2}}$$

The graphical illustration of the angular component u_θ is

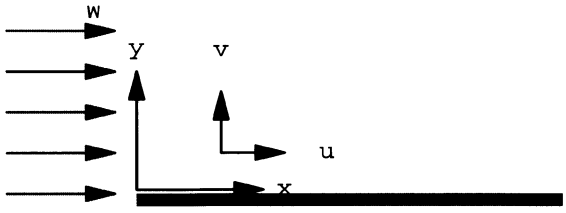
```
Show[
 {ContourPlot[If[x == 0 && y == 0, 0, u2], {x, -3, 3}, {y, -3, 3},
   PlotPoints → 35, ColorFunction → Hue, AxesLabel → {"x", "y"},
   Axes → True, Contours → 25, DisplayFunction → Identity],
  Graphics[Disk[{0, 0}, 1.1]]}, AspectRatio → Automatic,
 DisplayFunction → $DisplayFunction]
```



In this section, we presented the derivation of Stokes' classical solution of the creeping flow around a sphere. Contrary to Stokes, we did not guess the solution by an ingenious ansatz. The solution in our calculation followed directly from the symmetry analysis of the problem. This example illustrates the strength of this method by a straightforward calculation.

5.6.6 Two-Dimensional Boundary Layer Flows: Group Classification

In this section, we examine the problem of an incompressible boundary layer flow over a flat plate. We will not solve the physical problem but discuss the symmetries of several related models. The models differ from each other in the behavior of the vertical velocity component above the boundary. We demonstrate that the models are the result of group classification of the general equation. We show that the symmetry of the different models is closely connected with the symmetry of the general model. In addition, we illustrate that *MathLie* is capable of extracting these models from a symmetry calculation. The physical arrangement of the flow is given in the following figure. The flow above a plate is governed by the mainstream velocity W. The coordinates are denoted by x and y and the velocity components in the two directions are u and v, respectively.



The general two-dimensional, steady, laminar incompressible boundary layer equations follow from the Navier-Stokes equations if we neglect buoyancy. The resulting equations are the continuity equation and the momentum equation for steady flow (Lamb [1945]):

$$u_x + v_y = 0, \tag{5.61}$$

$$u\,u_x + v\,u_y - W(x)\,W_x - \eta\,u_{x,x} = 0. \tag{5.62}$$

Here, $W = W(x)$ is a known function describing the mainstream flow velocity in the x direction and η is the kinematic viscosity. The boundary conditions to be satisfied are

$$u(x, y = 0) = v(x, y = 0) = 0 \tag{5.63}$$

and

$$u(x, \infty) = W(x). \tag{5.64}$$

If we denote by

```
U = u[x, y]; V = v[x, y];
```

the velocity fields in x and y directions, we can write the left-hand side of the two equations in *Mathematica* by

```
bound = {∂x U + ∂y V, U ∂x U + V ∂y U - W[x] ∂x W[x] - η ∂{y,2} U};
bound // LTF
```

$$u_x + v_y == 0$$

$$u\,u_x + v\,u_y - W\,W_x - \eta\,u_{y,y} == 0$$

The equations *bound* constitute a system of equations in two dependent variables u and v. This is a system of partial differential equations of parabolic type. The equations are today known as Prandtel's boundary layer equations. If we now define a stream function by

```
stream = {u → Function[{x, y}, ∂y Ψ[x, y]],
  v → Function[{x, y}, -∂x Ψ[x, y]]}
```

```
{u → Function[{x, y}, ∂y Ψ[x, y]],
 v → Function[{x, y}, -∂x Ψ[x, y]]}
```

we can reduce the two equations of motion to a single equation given by

```
bound1 = bound /. stream; bound1 // LTF
```

```
True
-W Wx + Ψy Ψx,y - Ψx Ψy,y - η Ψy,y,y == 0
```

representing a non-linear partial differential equation for the stream function Ψ of third order. In the following, we will analyze this equation to derive explicit solutions for different mainstream velocities W. Utilizing the functions of *MathLie*, we are able to determine the symmetries by

```
symbound = Infinitesimals[{bound1[[2]]}, {Ψ}, {x, y}, {η}];
symbound // LTF
```

```
ϕ1 == k1
ξ1 == 0
ξ2 == F1
```

The result shows that the equation for the stream function with arbitrary mainstream velocity allows only a translation with respect to the stream function. In the following, we will examine the influence of the mainstream function on the symmetries.

5.6.6.1 The Blasius Solution

If we assume that the mainstream velocity is a constant $W(x) = w$ as in the problem discussed by Blasius [1908], the symmetry analysis results in the representation of infinitesimals:

```
mainstream = W → Function[x, w];

symbound =
 Infinitesimals[{bound1[[2]]} /. mainstream, {Ψ}, {x, y}, {η, w}];
symbound // LTF
```

```
ϕ1 == k1 + k2 Ψ
ξ1 == k3 + k4 x
ξ2 == (-k2 + k4) y + F1
```

The similarity reduction for this model follows by using the function LieReduction[]. Let us first examine the reduction of the four-dimensional group with respect to translations. This subgroup is selected by the group constants *k3*, *k1*, and the arbitrary function *free[1]* which we set equal to a constant. The related infinitesimals are thus given by

```
infi1 = {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]},
    {phi[1][x, y, Ψ]}} /. symbound /.
  {k1 → 1, k2 → 0, k3 → 1, k4 → 0, free[1][x] → c}
```

```
{{1, c}, {1}}
```

The reduction of the equation follows for this subgroup to

```
red1 = LieReduction[
  {bound1[[2]]} /. mainstream, {Ψ}, {x, y}, infi1[[1]], infi1[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$-c\,x + y - \zeta_1 == 0$$
$$-x + \Psi - F_1 == 0$$
$$-F1_{\zeta_1,\zeta_1} - \eta\, F1_{\zeta_1,\zeta_1,\zeta_1} == 0$$

The related similarity solution is obtained if we integrate the third-order ordinary differential equation

```
sred1 = DSolve[red1[[3, 1]] == 0, F1, zeta1]
```

$$\left\{\left\{F1 \to \left(C[1] + E^{-\frac{\#1}{\eta}}\, C[3] + C[2]\ \#1\&\right)\right\}\right\}$$

and insert the result into the representation of the solution

```
ssred1 = Solve[Flatten[red1[[2]] /. sred1], Ψ]
```

$$\left\{\left\{\Psi \to x + C[1] + (-c\,x + y)\, C[2] + E^{-\frac{-c\,x+y}{\eta}}\, C[3]\right\}\right\}$$

The stream function Ψ is determined by three integration constants $C[i]$, $i = 1, 2, 3$. The solution was derived under the condition that the problem allows the invariance of translation in the independent and dependent variables. The corresponding components of the velocity fields follow by

```
rule = Ψ → Function[{x, y}, w] /. (ssred1 /. Ψ → w);

velocities = {u[x, y], v[x, y]} /. stream /. rule
```

$$\left\{C[2] - \frac{E^{-\frac{-c\,x+y}{\eta}}\,C[3]}{\eta},\ -1 + c\,C[2] - \frac{c\,E^{-\frac{-c\,x+y}{\eta}}\,C[3]}{\eta}\right\}$$

where c is the group parameter and $C[2]$ and $C[3]$ are constants of integration.

Another subgroup of the Blasius model for the steady two-dimensional flow for a flat plate is the scaling invariance. This kind of invariance is selected from the total group if we set the group parameters *k2* and *k4* equal to a constant value. We get the infinitesimals from

```
infi1 = {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]},
    {phi[1][x, y, Ψ]}} /. symbound /.
  {k1 → 0, k2 → c, k3 → 0, k4 → 1, free[1][x] → 0}
```

```
{{x, (1 - c) y}, {c Ψ}}
```

The second reduction of the equation follows by

```
red2 =
 Simplify[PowerExpand[LieReduction[{bound1[[2]]} /. mainstream,
    {Ψ}, {x, y}, infi1[[1]], infi1[[2]]]]];
LTF[Flatten[red2]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$x^{-1+c}\,y - \zeta_1 == 0$$

$$x^{-c}\,\Psi - F_1 == 0$$

$$y^{-\frac{4c}{-1+c}}\,\zeta_1^{\frac{4c}{-1+c}}\,((-1+2c)\,(F_1)^2_{\zeta_1} - c\,F_1\,(F_1)_{\zeta_1,\zeta_1} - \eta\,(F_1)_{\zeta_1,\zeta_1,\zeta_1}) == 0$$

If we try to solve this type of equation by DSolve[], we get the result

```
DSolve[red2[[3]], F1, zeta1]
```

```
DSolve[{y^(-4 c/(-1 + c)) zeta1^(4 c/(-1 + c)) ((-1 + 2 c) F1'[zeta1]^2 -
      c F1[zeta1] F1''[zeta1] - η F1^(3)[zeta1]) == 0},
 F1,
 zeta1]
```

However, we can use Lie's methods to examine the symmetries of this ordinary differential equation. The symmetries of this equation are given by

```
red2eq = Thread[red2[[3, 1]] / y^(-4c/(-1+c)) zeta1^(-4c/(-1+c)), Equal];

iblasius = Infinitesimals[red2eq, F1, zeta1, {c, η}];
iblasius // LTF
```

ϕ_1 == -F1 k2
ξ_1 == k1 + k2 zeta1

demonstrating that the Blasius equation allows a two-dimensional symmetry group containing a translation with respect to the independent variable *zeta1* and a scaling in the independent and dependent variables. Since the Blasius equation is a third-order ODE but the symmetry group is of dimension two, we know from Chapter 4 that at least a reduction of the order is possible. At this point, the solution procedure ends since the number of symmetries is smaller than the degree of the ODE.

5.6.6.2 Falkner-Skan Solution

For the same geometrical situation, Falkner and Skan [1931] proposed that the mainstream velocity $W(x)$ is a power law function of the horizontal coordinate. We define this relation as

```
mainstream = W → Function[x, k x^m]
```

W → Function[x, k x^m]

where k and m are real constants. The symmetries of the stream function Ψ are determined for this case by

```
symbound = Infinitesimals[
  {bound1[[2]]} /. mainstream, {Ψ}, {x, y}, {η, k, m}];
symbound // LTF
```

ϕ_1 == k1 + k2 Ψ

$$\xi_1 == \frac{2\,k2\,x}{1+m}$$

$$\xi_2 == -\frac{k2\,(-1+m)\,y}{1+m} + \mathcal{F}_1$$

For the Falkner-Skan model, we find a two-dimensional discrete symmetry group and, in addition, an infinite dimensional group represented by *free[1]* = $\mathcal{F}_1$. Compared with the case when W is arbitrary, the group is enlarged by an additional degree of freedom. With respect to the Blasius group, the dimension is reduced by two components. The main symmetries consist of a translation and a scaling. Let us first discuss the translation symmetry. The related infinitesimals follow by setting *k1* and *free[1]* to constants:

```
infi1 = {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]},
    {phi[1][x, y, Ψ]}} /. symbound /.
  {k1 → 1, k2 → 0, free[1][x] → c}
```

```
{{0, c}, {1}}
```

The reduction of the stream function equation follows by

```
red1 =
 Simplify[PowerExpand[LieReduction[{bound1[[2]]} /. mainstream,
    {Ψ}, {x, y}, infi1[[1]], infi1[[2]]]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$x - \zeta_1 == 0$$

$$-\frac{y}{c} + \Psi - F_1 == 0$$

$$-k^2\, m\, \zeta_1^{2\,m} == 0$$

The result is somehow surprising since it does not contain any reduction of the original equation. If we examine the equation for the stream function, we observe that all terms contain derivatives with respect to the coordinate y which single out the similarity solution. Only the additive term containing the mainstream velocity W remains in the reduction. The result shows us that under the symmetry of translations, only a solution depending linearly on the vertical coordinate y exists.

The other type of symmetry contained in the Falkner-Skan case represents a scaling. The reduction for the scaling symmetry is given by

```
infi1 = {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]},
    {phi[1][x, y, Ψ]}} /. symbound /.
  {k1 → 0, k2 → c, free[1][x] → 0}
```

$$\left\{\left\{\frac{2\,c\,x}{1+m}, -\frac{c\,(-1+m)\,y}{1+m}\right\}, \{c\,\Psi\}\right\}$$

where we set the group parameters $k2$ equal to a constant c. The reduction of the equation for the stream function follows by

```
red2 =
 Simplify[PowerExpand[LieReduction[{bound1[[2]]} /. mainstream,
    {Ψ}, {x, y}, infi1[[1]], infi1[[2]]]]];
LTF[Flatten[red2]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$x^{\frac{1}{2}(-1+m)}\, y - \xi_1 == 0$$

$$x^{\frac{1}{2}(-1-m)}\, \Psi - F_1 == 0$$

$$-y^{-\frac{4m}{-1+m}}\, \xi_1^{\frac{4m}{-1+m}}\, (-2\, m\, (F_1)_{\xi_1}^2 + (1+m)\, F_1\, (F_1)_{\xi_1,\xi_1} + 2\, (k^2\, m + \eta\, (F_1)_{\xi_1,\xi_1,\xi_1})) == 0$$

Again, DSolve[] is unable to find a solution. The type of the resulting equation is the same as the equation in the Blasius model. The same arguments apply here. The number of symmetries of the equation is not sufficient for an integration. This is one reason why the Falkner-Skan equation is an unsolved problem. We will not examine the numerical solution of this equation which actually was carried out by Falkner and Skan [1931] in their paper. However our interest is concerned with other possibilities to model the mainstream velocity. There is another case which enlarges the number of symmetries of the steady two-dimensional flow.

5.6.6.3 Exponential Mainstream Velocity

Another way to choose the mainstream velocity is to assume an exponential increase in the horizontal direction. The function of this type is given by

```
mainstream = W → Function[x, k Exp[a x]]
```

```
W → Function[x, k Exp[a x]]
```

The infinitesimals for this sort of the mainstream velocity are

```
symbound = Infinitesimals[
  {bound1[[2]]} /. mainstream, {Ψ}, {x, y}, {v, k, a}];
symbound // LTF
```

$$\phi_1 == k1 - k2\, \Psi$$

$$\xi_1 == -\frac{2\, k2}{a}$$

$$\xi_2 == k2\, y + \mathcal{F}_1$$

The result contains a two-dimensional finite symmetry group representing a translation and some sort of scaling. The undetermined function $free[1][x] = \mathcal{F}_1$ extends the finite group to an infinite one. Again, the symmetry group is extended if we compare it with the general case in which the mainstream velocity is an arbitrary function. The reductions for the scaling symmetry follow by

```
infi1 = {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]},
    {phi[1][x, y, Ψ]}} /. symbound /.
  {k1 → 0, k2 → c, free[1][x] → 0}
```

$$\left\{\left\{-\frac{2\, c}{a},\ c\, y\right\},\ \{-c\, \Psi\}\right\}$$

```
red2 =
 Simplify[PowerExpand[LieReduction[{bound1[[2]]} /. mainstream,
    {Ψ}, {x, y}, infi1[[1]], infi1[[2]]]]];
LTF[Flatten[red2]] /. zeta1 → ζ1

Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$E^{\frac{a x}{2}} y - \zeta_1 == 0$$

$$E^{-\frac{a x}{2}} \Psi - F_1 == 0$$

$$-\zeta_1^4 \left(2 a k^2 - 2 a (F_1)_{\zeta_1}^2 + a F_1 (F_1)_{\zeta_1,\zeta_1} + 2 \eta (F_1)_{\zeta_1,\zeta_1,\zeta_1}\right) == 0$$

The reduced equation is again of the Falkner-Skan type. Thus, both types of the mainstream velocity result into the same type of equation.

5.6.6.4 Group Classification

The general topic behind the calculations carried out above is the problem of group classification of a partial differential equation. The question is formulated as follows. Assume we have a system of equations containing a certain arbitrariness, expressed in the dependence of the equations on certain parameters or functions. These equations admit a certain group G. If we now change the arbitrariness to a specific form, we may observe that the group G is enlarged. This behavior of enlargement of a group was the result of our previous discussion. The question now is: Can we find the specific forms for the mainstream velocity W discussed in the previous sections by using the functions of *MathLie*? The problem of group classification is closely connected with the common factors occurring in the determining equations. These common factors are eliminated by the functions of *MathLie*. The information removed from the determining equations is not lost but collected in a global variable called *EliminatedFactors*. This list collects all factors removed by the functions Lie[], LieSolve[], Infinitesimals[], DeterminingEquations[], and PDESolve[]. If we need to solve the classification problem, we have to examine the list *EliminatedFactors*.

The following considerations will illustrate the special cases for the mainstream velocity discussed above. All models discussed so far follow from a group classification and can be calculated from the eliminated common factors. We start the determination of the general classification problem by calling the function Infinitesimals[]:

```
symbound = Infinitesimals[{bound1[[2]]}, {Ψ}, {x, y}, {v}];
symbound // LTF
```

$$\phi_1 == k1$$
$$\xi_1 == 0$$
$$\xi_2 == \mathcal{F}_1$$

The factors which were cancelled in the derivation of the determining equations can be inspected just by reading the variable *EliminatedFactors:*

```
elFactor = EliminatedFactors; elFactor // LieTraditionalForm
```

$$\left\{-1,\ 1,\ \frac{4}{3},\ -3\,W\,W_x,\ \frac{W_x}{3\,W}+\frac{W_{x,x}}{3\,W_x},\ \frac{W_x^3}{W\,(W_x^2+W\,W_{x,x})}-\frac{W_x\,W_{x,x}}{W_x^2+W\,W_{x,x}}+\right.$$
$$\frac{W\,W_{x,x}^2}{W_x\,(W_x^2+W\,W_{x,x})}-\frac{W\,W_{x,x,x}}{W_x^2+W\,W_{x,x}},\ -\frac{3\,W_x\,W_{x,x}}{4\,(W_x^2+W\,W_{x,x})}-\frac{W\,W_{x,x,x}}{4\,(W_x^2+W\,W_{x,x})},$$
$$-\frac{W_x^3}{W\,(W_x^2+W\,W_{x,x})}+\frac{W_x\,W_{x,x}}{4\,(W_x^2+W\,W_{x,x})}-\frac{W\,W_{x,x}^2}{W_x\,(W_x^2+W\,W_{x,x})}+\frac{3\,W\,W_{x,x,x}}{4\,(W_x^2+W\,W_{x,x})},$$
$$\frac{9\,W_x^5\,W_{x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}-$$
$$\frac{27\,W\,W_x^3\,W_{x,x}^2}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}+$$
$$\frac{18\,W^2\,W_x\,W_{x,x}^3}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}+$$
$$\frac{15\,W\,W_x^4\,W_{x,x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}-$$
$$\frac{18\,W^2\,W_x^2\,W_{x,x}\,W_{x,x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}+$$
$$\frac{3\,W^3\,W_{x,x}^2\,W_{x,x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}-$$
$$\frac{6\,W^3\,W_x\,W_{x,x,x}^2}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}+$$
$$\frac{3\,W^2\,W_x^3\,W_{x,x,x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}+$$
$$\left.\frac{3\,W^3\,W_x\,W_{x,x}\,W_{x,x,x,x}}{(W_x^2+W\,W_{x,x})\,(-4\,W_x^4+W\,W_x^2\,W_{x,x}-4\,W^2\,W_{x,x}^2+3\,W^2\,W_x\,W_{x,x,x})}\right\}$$

In addition to three numerical factors, the list contains six relations which determine the mainstream velocity W by a differential equation. In the following calculations, we will show that all models for W discussed so far are contained in these equations. Let us start with the first equation which is extracted from position four of the list *elFactor*. Applying DSolve[] to this equation, we get

```
eq1 = elFactor[[4]] == 0; eq1 // LTF
```

$$-3\,W\,W_x == 0$$

```
DSolve[eq1, W, x]
```

$$\{\{W \to (C[1]\&)\}\}$$

The result shows that any constant is sufficient to satisfy this equation. The constant case for the mainstream velocity is the model discussed by Blasius [1908]. The fifth equation from the list of common factors gives us

```
eq2 = elFactor[[5]] == 0; eq2 // LTF
```

$$\frac{W_x}{3\,W} + \frac{W_{x,x}}{3\,W_x} == 0$$

```
DSolve[eq2, W, x]
```

$$\left\{\left\{W \to \left(-\sqrt{E^{3\,C[2]}\,(2\,\#1 - 2\,C[1])}\,\&\right)\right\},\ \left\{W \to \left(\sqrt{E^{3\,C[2]}\,(2\,\#1 - 2\,C[1])}\,\&\right)\right\}\right\}$$

This solution is a special case of the Falkner-Skan model with $m = 1/2$. Thus, the eliminated prefactors contain at least the special case of $W(x) = k\,x^{1/2}$. The sixth equation of our list contains a very complicated non-linear third-order ordinary differential equation which we use in the form

```
eq3 = Numerator[Together[elFactor[[6]]]]; eq3 // LTF
```

$$W_x^4 - W\,W_x^2\,W_{x,x} + W^2\,W_{x,x}^2 - W^2\,W_x\,W_{x,x,x} == 0$$

If we try to solve this equation by using DSolve[], we end up with

```
DSolve[eq3 == 0, W, x]
```

```
DSolve[
 W'[x]^4 - W[x] W'[x]^2 W''[x] + W[x]^2 W''[x]^2 - W[x]^2 W'[x] W^(3)[x] == 0,
 W, x]
```

However, the equation is solved by an exponential function

```
eq3 /. W → Function[x, k Exp[a x]]
```

$$0$$

This result shows that the exponential model discussed earlier is also consistent with the determining equations for the mainstream velocity. The seventh equation of our

list *elFactor* is connected with the equation *eq2*. We can show this by just integrating the equation with respect to x:

```
eq4 = Numerator[Together[elFactor[[7]]]]; eq4 // LTF
```

$$-3\,W_x\,W_{x,x} - W\,W_{x,x,x} == 0$$

```
∫ eq4 dx
```

$$-W'[x]^2 - W[x]\,W''[x]$$

Thus, no more information is gained by considering this equation. The eighth equation is a third-order ODE which cannot be treated by DSolve[]. The solution of this third-order ODE is again a special case of the Falkner-Skan type with $m = -1/2$:

```
eq5 = Numerator[Together[elFactor[[8]]]]; eq5 // LTF
```

$$-4\,W_x^4 + W\,W_x^2\,W_{x,x} - 4\,W^2\,W_{x,x}^2 + 3\,W^2\,W_x\,W_{x,x,x} == 0$$

```
Simplify[eq5 /. W → Function[x, k x^m]]
```

$$-2\,k^4\,m^2\,(-1 + m + 2\,m^2)\,x^{4\,(-1+m)}$$

The last relation is a fourth-order ODE not solvable by DSolve[]. A particular solution of this equation is, however, given by the Falkner-Skan relation for the mainstream velocity. We can check this by

```
eq6 = Numerator[Together[elFactor[[9]]]]; eq6 // LTF
```

$$\begin{aligned}-3\,(&3\,W_x^5\,W_{x,x} - 9\,W\,W_x^3\,W_{x,x}^2 + 6\,W^2\,W_x\,W_{x,x}^3 + \\ &5\,W\,W_x^4\,W_{x,x,x} - 6\,W^2\,W_x^2\,W_{x,x}\,W_{x,x,x} + W^3\,W_{x,x}^2\,W_{x,x,x} - \\ &2\,W^3\,W_x\,W_{x,x,x}^2 + W^2\,W_x^3\,W_{x,x,x,x} + W^3\,W_x\,W_{x,x}\,W_{x,x,x,x}) == \\ 0\end{aligned}$$

```
Simplify[eq6 /. W → Function[x, k x^m]]
```

$$0$$

Thus, we demonstrated that the group classification problem for the steady two-dimensional flow is solved by three types of mainstream velocities: (i) $W(x) = \text{const.}$, (ii) $W(x) = k\,x^m$, and (iii) $W(x) = k\,e^{ax}$. Special cases also contained in the classification of type (ii) are the cases with $m = 1/2$ and $m = -1/2$. The maximal group order of four occurs for the type (i), all other types possess a lower group order.

5.6.7 The Plane Jet

The plane jet is an example for Prandtl's boundary layer equations. We consider the steady two-dimensional motion of an incompressible viscous fluid due to a jet issuing from a long narrow orifice. We use Cartesian coordinates in the plane of motion. The origin of our coordinate system is located in the orifice and the x-axis lies in the plane of symmetry of the jet. The velocities in the x and y directions are denoted by u and v, respectively. If we assume that the Prandtl boundary layer equations give a sufficiently good approximation and that the pressure is a constant, then the equations of the stationary problem read

$$u_x + v_y = 0, \tag{5.65}$$

$$u\,u_x + v\,u_y = \eta\,u_{y,y}\,. \tag{5.66}$$

In *Mathematica* notation, we find

```
jet = {u[x, y] ∂x u[x, y] + v[x, y] ∂y u[x, y] == η ∂{y,2} u[x, y],
  ∂x u[x, y] + ∂y v[x, y] == 0};
jet // LTF
```

```
u uₓ + v u_y − η u_{y,y} == 0
uₓ + v_y == 0
```

where η is the kinematic viscosity. The equations of motion are accompanied with the boundary conditions

```
bound = {∂y u[x, y] == 0, v[x, y] == 0} /. y → 0
```

```
{u^(0,1)[x, 0] == 0, v[x, 0] == 0}
```

and the asymptotic behavior

```
asympt = {u[x,y]->0 /; y->Infinity}
```

An additional relation for the total x-component of the fluid momentum must be satisfied

```
M = 2 ρ ∫_ u[x, y]^2 dy == constant;
```

This sort of model was first discussed by Schlichting in 1933. We will use this model to discuss the analytic solution by means of a symmetry analysis. First, we transform Prandtl's boundary layer equations to a single equation by introducing the stream function representation:

```
stream = {u → Function[{x, y}, ∂y Ψ[x, y]],
  v → Function[{x, y}, -∂x Ψ[x, y]]}
```

```
{u → Function[{x, y}, ∂y Ψ[x, y]],
 v → Function[{x, y}, -∂x Ψ[x, y]]}
```

Applying this transformation to the jet equations, we find

```
sjet = jet /. stream; sjet // LTF
```

$$\Psi_y\,\Psi_{x,y} - \Psi_x\,\Psi_{y,y} - \eta\,\Psi_{y,y,y} == 0$$
$$\text{True} == 0$$

The original two equations reduce to a single equation for the stream function Ψ which is a non-linear partial differential equation of third order:

```
jet = sjet[[1]]
```

$$-\Psi^{(0,2)}[x, y]\,\Psi^{(1,0)}[x, y] + \Psi^{(0,1)}[x, y]\,\Psi^{(1,1)}[x, y] == \eta\,\Psi^{(0,3)}[x, y]$$

We determine the symmetries of this equation by applying the function Infinitesimals[]

```
symm = Infinitesimals[jet, Ψ, {x, y}, {η}]; symm // LTF
```

$$\phi_1 == k1 + k2\,\Psi$$
$$\xi_1 == k3 + k4\,x$$
$$\xi_2 == (-k2 + k4)\,y + \mathcal{F}_1$$

The result is a four-parameter group allowing translations and scalings as symmetries. Let us first discuss the translation symmetries and afterward use the scaling symmetry in our calculations. The related reduction of the original PDE follows by selecting the subgroups with *k3=1*, *free[1][x]=c*, and *k1=1*.

```
infi1 =
 {{xi[1][x, y, Ψ], xi[2][x, y, Ψ]}, {phi[1][x, y, Ψ]}} /. symm /.
  {k1 → 1, k2 → 0, k3 → 1, k4 → 0, free[1][x] → c}
```

```
{{1, c}, {1}}
```

The reduction follows by

```
red1 = LieReduction[jet, {Ψ}, {x, y}, infi1[[1]], infi1[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$-c\,x + y - \zeta_1 == 0$$
$$-x + \Psi - F_1 == 0$$
$$-F1_{\zeta_1,\zeta_1} - \eta\,F1_{\zeta_1,\zeta_1,\zeta_1} == 0$$

The solution of the reduced equation follows by

```
ssol = DSolve[red1[[3]], F1, zeta1]
```

$$\left\{\left\{\text{F1} \to \left(\text{E}^{-\frac{\#1}{\eta}}\ \eta^2\ \text{C[1]} + \text{C[2]} + \text{C[3]}\ \#1\&\right)\right\}\right\}$$

where #1 represents the variable *zeta*1. The similarity solution in the original variables x and y is

```
sol = Flatten[Solve[Flatten[red1[[2]] /. ssol], Ψ]]
```

$$\left\{\Psi \to \text{x} + \text{E}^{-\frac{-\text{c x}+\text{y}}{\eta}}\ \eta^2\ \text{C[1]} + \text{C[2]} + (-\text{c x} + \text{y})\ \text{C[3]}\right\}$$

This solution contains three constants of integration *C[i]*, i = 1, 2, 3. The constants must be chosen in such a way that the boundary conditions are satisfied. The first boundary condition requires

```
Simplify[∂{y,2} (Ψ /. sol) /. y → 0] == 0
```

$$\text{E}^{\frac{\text{c x}}{\eta}}\ \text{C[1]} == 0$$

which can only be satisfied by

```
rule = {C[1] → 0}
```

$$\{\text{C[1]} \to 0\}$$

The second boundary condition requires

```
Simplify[∂{x,1} (Ψ /. sol) /. y → 0] == 0 /. rule
```

$$1 - \text{c C[3]} == 0$$

which allows a special choice $C[3]=1/c$

```
AppendTo[rule, C[3] → 1/c]
```

$$\left\{\text{C[1]} \to 0,\ \text{C[3]} \to \frac{1}{\text{c}}\right\}$$

Using the asymptotic behavior of u results in

```
Simplify[∂{y,1} (Ψ /. sol) /. rule]
```

$$\frac{1}{\text{c}}$$

The result shows that the requirement that $u \to 0$ for $y \to \infty$ can only be satisfied if we set $c \to \infty$. In conclusion, the symmetry of translation is not compatible with the boundary conditions.

The second type of symmetry under which the third-order PDE is invariant is given by a scaling. The related subgroup is extracted by setting *k2=c* and *k4=1*:

```
infi1 =
 {{xi[1][x, y, Φ], xi[2][x, y, Φ]}, {phi[1][x, y, Φ]}} /. symm /.
  {k1 → 0, k2 → c, k3 → 0, k4 → 1, free[1][x] → 0}
```

```
{{x, (1 - c) y}, {c Φ}}
```

The similarity reduction for the scaling group follows by

```
red2 = LieReduction[jet, {Φ}, {x, y}, infi1[[1]], infi1[[2]]];
LTF[Flatten[red2]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$x^{-1+c}\, y - \zeta_1 == 0$$

$$x^{-c}\, \Phi - F_1 == 0$$

$$y^{-\frac{4c}{-1+c}}\, \zeta_1^{\frac{4c}{-1+c}}\, \left(-(F_1)_{\zeta_1}^2 + 2c\,(F_1)_{\zeta_1}^2 - c\,F_1\,(F_1)_{\zeta_1,\zeta_1} - \eta\,(F_1)_{\zeta_1,\zeta_1,\zeta_1}\right) == 0$$

The left-hand side of the equation is without any common coefficients

```
eq2 = red2[[3, 1, 1]] / (y^(-4c/(-1+c)) zeta1^(4c/(-1+c))); LTF[eq2] /. zeta1 → ζ1
```

$$-(F_1)_{\zeta_1}^2 + 2c\,(F_1)_{\zeta_1}^2 - c\,F_1\,(F_1)_{\zeta_1,\zeta_1} - \eta\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} == 0$$

This third-order ODE is not solved by DSolve[]. However, a first integration with respect to *zeta1* shows us that the equation is integrable if we choose a special value for *c*. If we set $c \to 1/3$, we get a second-order ODE which is equal to a first constant of integration:

```
(eq2 /. c → 1/3)
```

$$-\frac{1}{3}\,\text{F1}'[\text{zeta1}]^2 - \frac{1}{3}\,\text{F1}[\text{zeta1}]\,\text{F1}''[\text{zeta1}] - \eta\,\text{F1}^{(3)}[\text{zeta1}]$$

```
int1 = ∫ (eq2 /. c → 1/3) dzeta1 == K1
```

$$-\frac{1}{3}\,\text{F1}[\text{zeta1}]\,\text{F1}'[\text{zeta1}] - \eta\,\text{F1}''[\text{zeta1}] == \text{K1}$$

Taking the boundary conditions into account, we realize that $K1 = 0$. This follows since $F1'(0) = 0$ and $F1''(0) = 0$. Thus, the result simplifies and we can integrate the relation a second time:

```
int2 = ∫ int1[[1]] dzeta1 == -K2
```

$$-\frac{1}{6}\,\text{F1}[\text{zeta1}]^2 - \eta\,\text{F1}'[\text{zeta1}] == -\text{K2}$$

Now using DSolve[] to integrate the first-order ordinary differential equation, we get the solution

```
sols = Flatten[DSolve[int2, F1, zeta1]]
```

$$\left\{\text{F1} \to \left(\sqrt{6}\,\sqrt{\text{K2}}\,\text{Tanh}\left[\frac{1}{6}\left(\frac{\sqrt{6}\,\sqrt{\text{K2}}\,\#1}{\eta} - 6\sqrt{6}\,\sqrt{\text{K2}}\,\text{C}[1]\right)\right]\&\right)\right\}$$

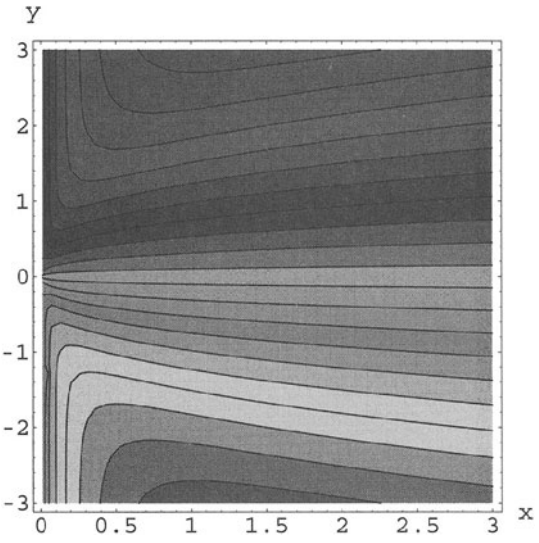
The corresponding similarity representation follows by

```
ssol = Solve[red2[[2]] /. sols /. c → 1/3, Ψ]
```

$$\left\{\left\{\Psi \to \sqrt{6}\,\sqrt{\text{K2}}\,x^{1/3}\,\text{Tanh}\left[\frac{1}{6}\left(\frac{\sqrt{6}\,\sqrt{\text{K2}}\,y}{x^{2/3}\,\eta} - 6\sqrt{6}\,\sqrt{\text{K2}}\,\text{C}[1]\right)\right]\right\}\right\}$$

This solution contains three parameters η, $K2$, and $C[1]$. η is the kinematic viscosity which we set to unity in the following. The integration constant $K2$ changes the amplitude as well as the argument of the Tanh[] function. $C[1]$ determines the location of the orifice which we set to the origin. The following contour plot of the stream function gives us a representation of the stream lines in the (x, y)-plane:

```
sol = (Ψ /. ssol /. {K2 → 1, η → 1, C[1] → 0})[[1]];

ContourPlot[sol, {x, .01, 3}, {y, -3, 3}, PlotPoints → 50,
 Contours → 20, ColorFunction → Hue, AxesLabel → {"x", "y"},
 Axes → True]
```



The representation of the stream lines shows that the jet fans out and that the velocity decreases for larger values of x. This behavior is obvious in a graphical representation of the velocity for the two components

```
u = ∂y sol1
```

$$\frac{\text{Sech}\left[\frac{y}{\sqrt{6}\ x^{2/3}}\right]^2}{x^{1/3}}$$

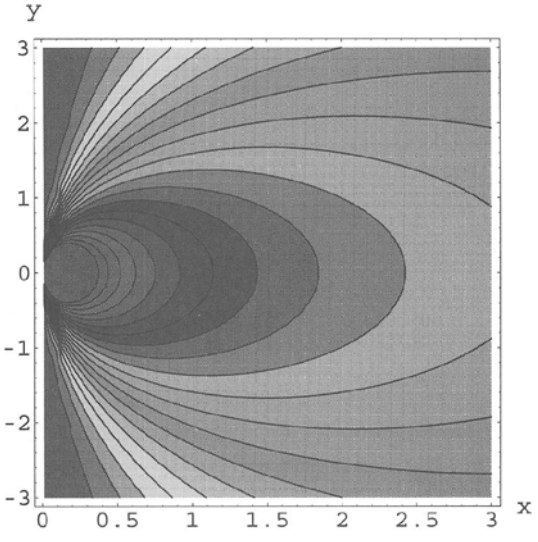
and

```
v = -∂x sol1
```

$$\frac{2\,y\,\text{Sech}\left[\frac{y}{\sqrt{6}\ x^{2/3}}\right]^2}{3\,x^{4/3}} - \frac{\sqrt{\frac{2}{3}}\,\text{Tanh}\left[\frac{y}{\sqrt{6}\ x^{2/3}}\right]}{x^{2/3}}$$

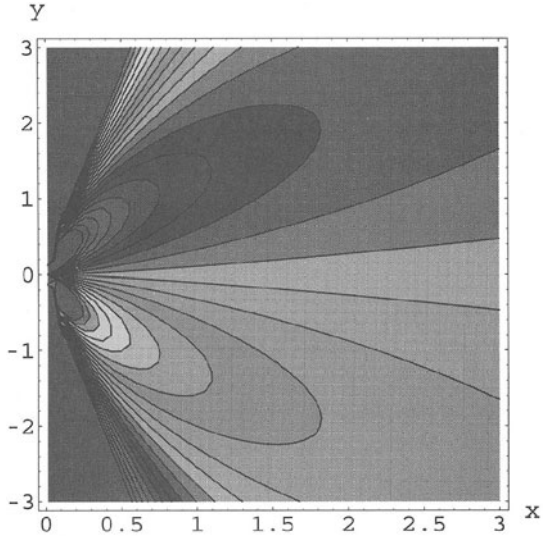
which both follow as derivatives from the stream function. We represent the two solutions by a contour plot. The velocity in the x direction is

```
ContourPlot[u, {x, .01, 3}, {y, -3, 3}, PlotPoints → 50,
 Contours → 20, ColorFunction → Hue, AxesLabel → {"x", "y"},
 Axes → True]
```

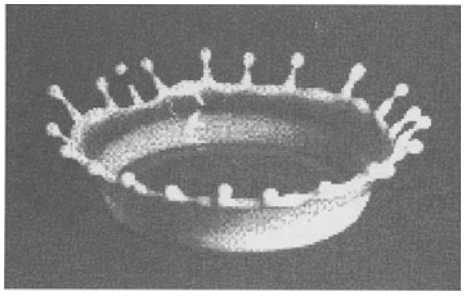


The v-component of the velocity looks like

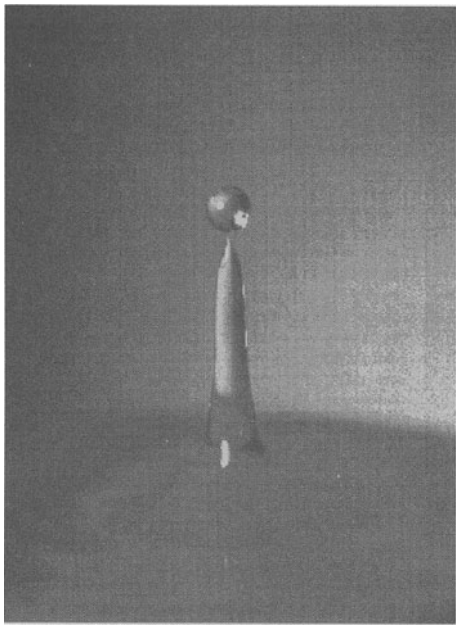
```
ContourPlot[v, {x, .01, 3}, {y, -3, 3}, PlotPoints → 50,
 Contours → 20, ColorFunction → Hue, AxesLabel → {"x", "y"},
 Axes → True]
```



5.6.8 Drop Formation



In this section we discuss the formation of drops by an axisymmetric model. A typical example of drop formation is shown in the pictures below.



The figure of a single drop was taken from Eggers [1997]. Such a picture is observed if a drop hits a fluid surface. Drop formation is a common phenomenon in our daily world. Everybody knows the kinetics of rain drops if they hit the surface of a pond. An animation available in the notebook version of this book for a falling milk drop demonstrates the dynamic behind drop formation. Another animation showing the formation of a water drop illustrates that not only the drop itself is created but also secondary droplets. The pictures are taken from Perigrin et al. The laws behind this interaction has been the topic of research for the last 300 years. The interest in drop

formation is still growing because of its potential industrial application in technologies like ink jet printers, chemical mixing processes, fuel injection in engines, etc. The formation of a drop is mainly governed by two non-linear equations derived from the three-dimensional Navier-Stokes equation.

We assume that the Navier-Stokes equation is a reasonable model to describe the drop formation. The Navier-Stokes equation for an axisymmetric column of fluid with kinematic viscosity η, density ρ, and surface tension γ are given in cylindrical coordinates by

$$u_t + u\,u_r + v\,v_r = -\frac{p_r}{\rho} + \eta\left(u_{r,r} + u_{z,z} + \frac{u_r}{r} - \frac{u}{r^2}\right), \tag{5.67}$$

$$v_t + u\,v_r + v\,v_z = -\frac{p_z}{\rho} + \eta\left(v_{r,r} + v_{z,z} + \frac{v_r}{r}\right) - g, \tag{5.68}$$

where u is the velocity in the radial direction, v is the velocity along the axis, and p is the pressure. The equations are given by Landau and Lifshitz [1987] in the volume on fluid mechanics. The continuity equation in cylindrical coordinates reads

$$u_r + v_z + \frac{u}{r} = 0. \tag{5.69}$$

In addition to these three equations, we have to satisfy boundary conditions controlling the free surface of the fluid:

$$\vec{n}\,\sigma\,\vec{n} = -\gamma\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{5.70}$$

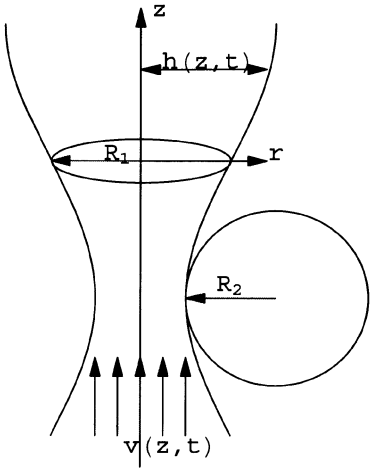
and

$$\vec{v}\,\sigma\,\vec{t} = 0. \tag{5.71}$$

Here, σ is the stress tensor, $\vec{n}$ the outward normal, and R_1 and R_2 are the principal radii of curvature (cf. figure below). The equation of motion for the height of the fluid neck $h = h(z, t)$ is

$$h_t + v\,h_z = u\,|_{r=h}\,. \tag{5.72}$$

In all the equations, subscripts refer to a partial differentiation with respect to the independent variables t, r, and z. The formula for the mean curvature $(1/R_1 + 1/R_2)/2$ of a body of revolution is known from differential geometry (cf. Gray [1993]). The geometrical relations are shown in the following figure:



The mean curvature which is responsible for the capillary pressure is given by the relation (Garcia and Castellanos [1994])

$$p = \frac{\gamma}{(1 + \epsilon h_z^2)^{1/2}} \left(\frac{1}{1 + \epsilon h} - \frac{\epsilon h_{z,z}}{1 + \epsilon h_z^2} \right) \tag{5.73}$$

where ϵ is a small parameter. The model equations (5.67)–(5.69) simplify considerably by expanding the velocity fields and the pressure into a power series in the radial direction. This expansion additionally allows the reduction from a two-dimesnional to a one-dimensional model. Keeping only the lowest-order radial dependence, we end up with the equations

$$v_t + v v_z = -\frac{p_z}{\rho} + 3 \frac{(\eta(h^2 v_z))_z}{h^2} - g, \tag{5.74}$$

$$h_t + v h_z = -\frac{1}{2} v_z h \tag{5.75}$$

where the pressure is given by (5.73) with $\epsilon = 1$. The detailed calculation of the approximation can be found in the paper by Eggers [1993]. Our interest here is the derivation of a solution for the velocity field v and the height of the fluid neck h in terms of similarity solutions. We first examine the model equations by Eggers with the complete pressure expression. Equations (5.74) and (5.75) are the starting point for our symmetry analysis. We first define the variables and the pressure by

```
V = v[z, t]; H = h[z, t];

P = γ ( 1 / (H (1 + (∂z H)^2)^1/2) - ∂z,z H / (1 + (∂z H)^2)^3/2 );

P // LieTraditionalForm
```

$$\gamma\left(\frac{1}{h\sqrt{1+h_z^2}}-\frac{h_{z,z}}{(1+h_z^2)^{3/2}}\right)$$

The equations of motion for the liquid jet are given by

```
eggers = {∂t V + V ∂z V + ∂z P/ρ - 3 η ∂z (H^2 ∂z V)/H^2 + ψ,
    ∂t H + V ∂z H + H ∂z V/2} // Together // Numerator;
eggers // LTF
```

$$\begin{aligned}
&-\gamma h_z - 2\gamma h_z^3 - \gamma h_z^5 + h^2\rho\psi\sqrt{1+h_z^2} + 2h^2\rho\psi h_z^2\sqrt{1+h_z^2} + \\
&\quad h^2\rho\psi h_z^4\sqrt{1+h_z^2} + h^2\rho\sqrt{1+h_z^2}\,v_t + 2h^2\rho h_z^2\sqrt{1+h_z^2}\,v_t + \\
&\quad h^2\rho h_z^4\sqrt{1+h_z^2}\,v_t + h^2 v\rho\sqrt{1+h_z^2}\,v_z - 6h\eta\rho h_z\sqrt{1+h_z^2}\,v_z + \\
&\quad 2h^2 v\rho h_z^2\sqrt{1+h_z^2}\,v_z - 12h\eta\rho h_z^3\sqrt{1+h_z^2}\,v_z + \\
&\quad h^2 v\rho h_z^4\sqrt{1+h_z^2}\,v_z - 6h\eta\rho h_z^5\sqrt{1+h_z^2}\,v_z - h\gamma h_z h_{z,z} - h\gamma h_z^3 h_{z,z} + \\
&\quad 3h^2\gamma h_z h_{z,z}^2 - 3h^2\eta\rho\sqrt{1+h_z^2}\,v_{z,z} - 6h^2\eta\rho h_z^2\sqrt{1+h_z^2}\,v_{z,z} - \\
&\quad 3h^2\eta\rho h_z^4\sqrt{1+h_z^2}\,v_{z,z} - h^2\gamma h_{z,z,z} - h^2\gamma h_z^2 h_{z,z,z} == \\
&0 \\
&2h_t + 2v h_z + h v_z == 0
\end{aligned}$$

where ρ, γ, η, and ψ denote the constant density, the surface tension, the kinematic viscosity, and the acceleration due to gravity, respectively. The infinitesimals of the equation follows by applying Infinitesimals[] to the equations:

```
infiEggers = Infinitesimals[eggers, {v, h}, {z, t},
  {ρ, γ, η, ψ}, SubstitutionRules → {∂{z,3} H, ∂t H}];
infiEggers // LTF
```

$$\begin{aligned}
&\phi_1 == k3 \\
&\phi_2 == 0 \\
&\xi_1 == k2 + k3\,t \\
&\xi_2 == k1
\end{aligned}$$

The result of the calculation is a finite three-dimensional point group. The symmetries of the group are translations in x and t, and a special subgroup denoted by *k3* allowing a translation in v and a transformation of x connected with t. These symmetries do not allow a scaling solution of the original equations. The scaling solutions are today used to discuss the pinch of the drop. However, we can find such solutions by changing the representation of the pressure.

The following calculations are based on the capillary pressure given by Garcia and Castellanos [1994]

```
P1 = γ ( 1 / ((1 + ϵ H) (1 + ϵ (∂z H)^2)^(1/2)) - ϵ ∂z,z H / (1 + ϵ (∂z H)^2)^(3/2) );
P1 // LieTraditionalForm
```

$$\gamma\left(\frac{1}{(1+h\,\epsilon)\sqrt{1+\epsilon\,h_z^2}}-\frac{\epsilon\,h_{z,z}}{(1+\epsilon\,h_z^2)^{3/2}}\right)$$

Following the considerations by Garcia and Castellanos, we assume first that the parameter ϵ is a small quantity. This property recommends a Taylor expansion of the pressure around $\epsilon = 0$. So we get an approximation of the capillary pressure for small ϵ's up to first order. To identify any correspondence of the following calculations with those carried out above, we set $\epsilon = 1$ in the expansion:

```
p1 = (Series[P1, {ϵ, 0, 1}] // Normal) /. ϵ → 1
```

$$\gamma+\gamma\left(-h[z,\,t]-\frac{1}{2}\,h^{(1,0)}[z,\,t]^2-h^{(2,0)}[z,\,t]\right)$$

The pressure in this approximation is given mainly by a constant altered by geometrical terms linear in h, quadratic in the gradient of the height of the fluid neck h, and linear in the second derivative of h. Inserting this result into the equations of motion (5.74) and (5.75), we find the reduced set

```
model1 = {∂t V + V ∂z V + ∂z p1/ρ - 3 η ∂z (H^2 ∂z V)/H^2 + ψ,
    ∂t H + V ∂z H + H ∂z V/2} // Together // Numerator;
model1 // LTF
```

$$h\,\rho\,\psi-h\,\gamma\,h_z+h\,\rho\,v_t+h\,v\,\rho\,v_z-6\,\eta\,\rho\,h_z\,v_z-h\,\gamma\,h_z\,h_{z,z}-3\,h\,\eta\,\rho\,v_{z,z}-h\,\gamma\,h_{z,z,z}==0$$

$$2\,h_t+2\,v\,h_z+h\,v_z==0$$

which are much simpler than the original model. Surprisingly, the infinitesimals of this reduced model are the same as in the original model:

```
inf1 = Infinitesimals[model1, {v, h}, {z, t},
  {ρ, γ, η, ψ}, SubstitutionRules → {∂{z,3} H, ∂t H}];
inf1 // LTF
```

$$\phi_1==k3$$

$$\phi_2==0$$

$$\xi_1==k2+k3\,t$$

$$\xi_2==k1$$

Thus, we observe that the change of the pressure for small values of ϵ does not affect the symmetries of the original model. This is also true for higher-order approximations in ϵ of p. The calculation is left as an exercise for the reader.

Another expression for the capillary pressure follows if we assume that the parameter ϵ is large. This limit can be achieved by replacing ϵ with $1/\lambda$ in formula (5.73). In *Mathematica*, we substitute $1/\lambda$ for ϵ and expand the resulting pressure formula around $\lambda = 0$:

```
P1 = γ ( 1 / ((1 + 1/λ H) (1 + 1/λ (∂z H)^2)^(1/2)) - 1/λ ∂z,z H / (1 + 1/λ (∂z H)^2)^(3/2) );
P1 // LieTraditionalForm
```

$$\gamma\left(\frac{1}{\left(1+\frac{h}{\lambda}\right)\sqrt{1+\frac{h_z^2}{\lambda}}}-\frac{h_{z,z}}{\lambda\left(1+\frac{h_z^2}{\lambda}\right)^{3/2}}\right)$$

The Taylor expansion of the pressure in the limit $\epsilon \to \infty$ follows by

```
p3 = (Series[P1, {λ, 0, 1}] // Normal) /. λ → 1;
p3 // LieTraditionalForm
```

$$-\frac{\gamma\, h_{z,z}}{(h_z^2)^{3/2}}$$

In this approximation, the pressure is given by a pure geometric term determined only by derivatives. This kind of approximation is similar to the Saffman-Taylor [1958] approximation and is also known as lubrication approximation. The equations of motion (5.74) and (5.75) in connection with the lubrication approximation are thus given by

```
model2 =
  {∂t V + V ∂z V + ∂z p3/ρ - 3 η ∂z (H^2 ∂z V)/H^2 + ψ, ∂t H + V ∂z H + H ∂z V/2} //
    Together // Numerator // PowerExpand;
model2 // LTF
```

$$h\rho\psi h_z^4 + h\rho h_z^4 v_t + hv\rho h_z^4 v_z - 6\eta\rho h_z^5 v_z + 3h\gamma h_{z,z}^2 - 3h\eta\rho h_z^4 v_{z,z} - h\gamma h_z h_{z,z,z} == 0$$

$$2h_t + 2vh_z + hv_z == 0$$

The corresponding infinitesimals for this kind of model follows by

```
inf3 = Infinitesimals[model2, {v, h}, {z, t},
  {ρ, γ, η, ψ}, SubstitutionRules → {∂{z,3} H, ∂t H}];
inf3 // LTF
```

$$\phi_1 == k4 + \frac{2\,k5\,(v + 3\,t\,\psi)}{3\,\psi}$$

$$\phi_2 == -\frac{h\,k5}{\psi}$$

$$\xi_1 == k2 + k3 + k4\,t + k5\,t^2 - \frac{2\,k5\,z}{3\,\psi}$$

$$\xi_2 == k1 - \frac{4\,k5\,t}{3\,\psi}$$

We clearly observe that the structure of the symmetries changed. The number of group parameters increased. The additional symmetries now allow a scaling solution of the equations. We also observe that the constant of gravity ψ determines the symmetry properties of the scaling transformation. The related group parameter of the scaling group is *k5*.

Another method of approximating the capillary pressure taken from Eggers is the assumption that the spatial derivatives in p are small compared with the horizontal elongation h itself. This assumption can be realized in *Mathematica* by the following replacement:

```
p4 = P /. Derivative[___][h][___] → 0
```

$$\frac{\gamma}{h[z,\,t]}$$

The two equations of motion simplify to

```
simplifiedEggers = {∂t V + V ∂z V + ∂z p4/ρ - 3 η ∂z (H^2 ∂z V)/H^2 + ψ,
    ∂t H + V ∂z H + H ∂z V/2} // Together // Numerator;
simplifiedEggers // LTF
```

$$h^2\,\rho\,\psi - \gamma\,h_z + h^2\,\rho\,v_t + h^2\,v\,\rho\,v_z - 6\,h\,\eta\,\rho\,h_z\,v_z - 3\,h^2\,\eta\,\rho\,v_{z,z} == 0$$

$$2\,h_t + 2\,v\,h_z + h\,v_z == 0$$

The symmetries of this set of equations follow by

```
inf4 = Infinitesimals[simplifiedEggers, {v, h},
  {z, t}, {ρ, γ, η, ψ}, SubstitutionRules → {∂t V, ∂t H}];
inf4 // LTF
```

$$\phi_1 == k4 + \frac{2\,k5\,(v + 3\,t\,\psi)}{3\,\psi}$$

$$\phi_2 == -\frac{4\,h\,k5}{3\,\psi}$$

$$\xi_1 == k2 + k3 + k4\,t + k5\,t^2 - \frac{2\,k5\,z}{3\,\psi}$$

$$\xi_2 == k1 - \frac{4\,k5\,t}{3\,\psi}$$

The result is again a five-dimensional finite group containing translations *k1*, *k2*, *k3*, and *k4*, and a scaling symmetry given by *k5*. The scaling symmetry depends on the gravity acceleration ψ. This set of symmetries is isomorphic to the infinitesimals *inf3*. The difference between the two models however is the pressure in the equations. Let us first examine the solution of the equations connected with the symmetry of translation. A subgroup of translations is given by the infinitesimals

```
infi1 = {{xi[1][z, t, v, h], xi[2][z, t, v, h]},
    {phi[1][z, t, v, h], phi[2][z, t, v, h]}} /. inf4 /.
  {k1 → 1, k2 → c, k3 → 0, k4 → 0, k5 → 0}
```

```
{{c, 1}, {0, 0}}
```

The corresponding similarity reduction of the original equations follows with

```
red1 = LieReduction[simplifiedEggers ,
  {v, h}, {z, t}, infi1[[1]], infi1[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$-\frac{-c\,t+z}{c}-\zeta_1==0$$

$$v-F_1==0$$

$$h-F_2==0$$

$$c^2\,\rho\,\psi\,F_2^2+c^2\,\rho\,F_2^2\,(F_1)_{\zeta_1}-c\,\rho\,F_1\,F_2^2\,(F_1)_{\zeta_1}+c\,\gamma\,(F_2)_{\zeta_1}-6\,\eta\,\rho\,F_2\,(F_1)_{\zeta_1}\,(F_2)_{\zeta_1}-3\,\eta\,\rho\,F_2^2\,(F_1)_{\zeta_1,\zeta_1}==0$$

$$-F_2\,(F_1)_{\zeta_1}+2\,c\,(F_2)_{\zeta_1}-2\,F_1\,(F_2)_{\zeta_1}==0$$

The similarity solution of the equations including gravity is given by a moving wave solution. Examining the second equation of the reduction, we realize that a special solution of this equation is a constant. Inserting this kind of solution into the reduced equations, we get

```
rode = red1[[3]] /. F1 → Function[zeta1, c]
```

$$\{c^2\,\rho\,\psi\,F2[zeta1]^2+c\,\gamma\,F2'[zeta1]==0,\ True\}$$

The two equations simplify to a single first-order ODE which allows the solution

```
s1 = DSolve[rode[[1]], F2, zeta1] // Flatten
```

$$\left\{F2\to\left(\frac{\gamma}{c\,\#1\,\rho\,\psi-C[1]}\&\right)\right\}$$

A special set of solutions is thus

```
AppendTo[s1, F1 → Function[zeta1, c]]; s1
```

$$\left\{F2 \to \left(\frac{\gamma}{c\,\#1\,\rho\,\psi - C[1]}\&\right),\ F1 \to \text{Function}[\text{zeta1, c}]\right\}$$

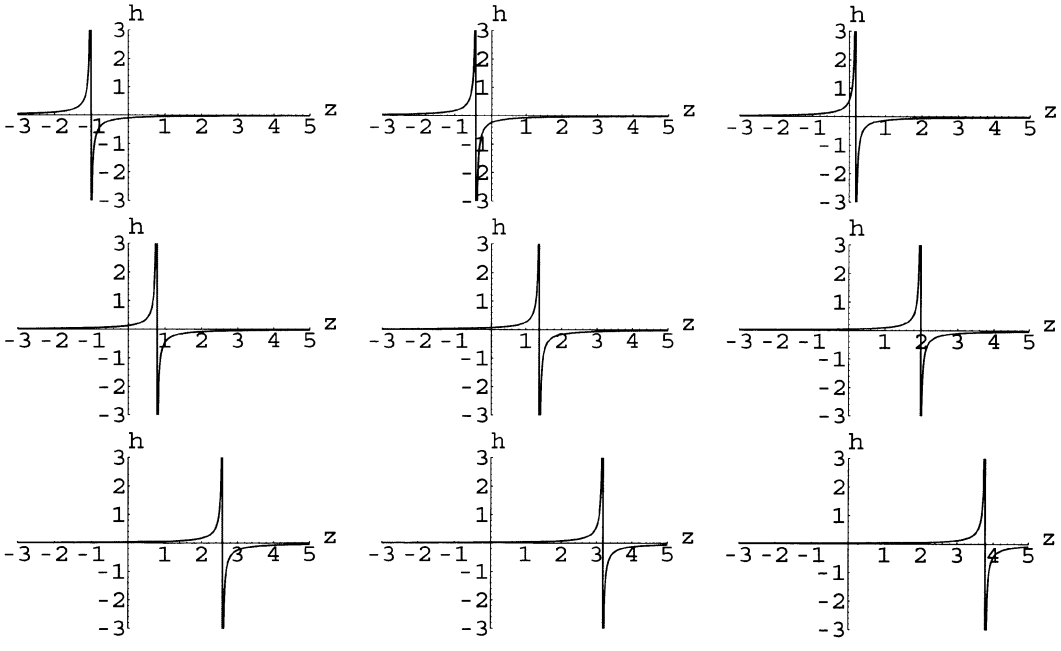
It is obvious from the formula above that the fluid neck becomes infinite if $\zeta = C[1]/(c\rho\psi)$. In the original variables, the solution reads

```
sol1 = red1[[2]] /. s1 // Simplify
```

$$\left\{v == c,\ h + \frac{\gamma}{-c\,t\,\rho\,\psi + z\,\rho\,\psi + C[1]} == 0\right\}$$

where $C[1]$ is a constant of integration. If we insert for the parameters γ, c, ρ, ψ, and $C[1]$ numeric values, we are able to plot the solution. The following animation shows the movement of the singularity of h along the z-axis while the time is changed from 0 to 2.5 in steps of 0.1.

```
Do[Plot[-γ/(-c t ρ ψ + z ρ ψ + C[1]) /.
    {γ → 1, c → 2, ρ → 1, ψ → 9.81, C[1] → 10, t → ti},
   {z, -3, 5}, PlotStyle → RGBColor[0, 0, 1],
   PlotRange → {{-3, 5}, {-3, 3}}, AxesLabel → {"z", "h"}],
  {ti, 0, 2.5, .1}]
```



This sequence of pictures shows us that under a translation invariance of equations (5.74) and (5.75), a singularity of the height h of the fluid neck moves from the left to the right. It is essential that the singularity in h exist from the beginning to the end. The structure of the solution is not changed during the evolution in time. If we look at the analytic expression of the solution, we realize that gravity seems to be an essential component to create the singularity. To study the influence of the gravity constant ψ on the solution, let us examine the simplified equations of motion without gravity. The reduced equations of motion are

```
modelWithoutGravity = simplifiedEggers /. ψ → 0;
modelWithoutGravity // LTF
```

$$-\gamma\, h_z + h^2\, \rho\, v_t + h^2\, v\, \rho\, v_z - 6\, h\, \eta\, \rho\, h_z\, v_z - 3\, h^2\, \eta\, \rho\, v_{z,z} == 0$$
$$2\, h_t + 2\, v\, h_z + h\, v_z == 0$$

The infinitesimals of this simplified model follow from

```
inf5 = Infinitesimals[modelWithoutGravity, {v, h},
  {z, t}, {ρ, γ, η}, SubstitutionRules → {∂t V, ∂t H}];
inf5 // LTF
```

$$\phi_1 == k2 - k4\, v$$
$$\phi_2 == 2\, h\, k4$$
$$\xi_1 == k3 + k2\, t + k4\, z$$
$$\xi_2 == k1 + 2\, k4\, t$$

The four-dimensional symmetry transformation contains a scaling symmetry, translations in z and t, and a special symmetry related to $k2$. Let us first examine the moving wave solution corresponding to the translational invariance. The related infinitesimals of the symmetry group are

```
infi2 = {{xi[1][z, t, v, h], xi[2][z, t, v, h]},
    {phi[1][z, t, v, h], phi[2][z, t, v, h]}} /. inf5 /.
  {k1 → 1, k2 → 0, k3 → c, k4 → 0}
```

```
{{c, 1}, {0, 0}}
```

The reduction of the equations of motion without gravity follows from

```
red2 = LieReduction[modelWithoutGravity,
  {v, h}, {z, t}, infi2[[1]], infi2[[2]]];
(red2 // Flatten // LTF) /. zeta1 → ζ1
```

$$-\frac{-c\, t + z}{c} - \zeta_1 == 0$$
$$v - F_1 == 0$$

$$h - F_2 == 0$$

$$c^2 \rho F_2^2 (F_1)_{\zeta_1} - c \rho F_1 F_2^2 (F_1)_{\zeta_1} + c \gamma (F_2)_{\zeta_1} - 6 \eta \rho F_2 (F_1)_{\zeta_1} (F_2)_{\zeta_1} - 3 \eta \rho F_2^2 (F_1)_{\zeta_1, \zeta_1} == 0$$

$$-F_2 (F_1)_{\zeta_1} + 2 c (F_2)_{\zeta_1} - 2 F_1 (F_2)_{\zeta_1} == 0$$

We again realize that the second equation is solved with $F1 = c$, meaning that the velocity field v is a constant. Inserting this solution into similarity reduction, we find

```
red3 = red2[[3]] /. F1 → Function[zeta1, c];
(red3 // LieTraditionalForm) /. zeta1 → ζ1
```

$$\{c \gamma F2_{\zeta_1} == 0, \text{True}\}$$

The remaining first-order ODE is solved by

```
s1 = DSolve[red3[[1]], F2, zeta1]
```

$$\{\{F2 \to (C[1]\&)\}\}$$

Thus, the height of the fluid neck remains a constant during the evolution. The mathematical result in physical terms means that a column of fluid of height h moving with a constant velocity a without any influence of gravity remains in the same initial state forever. Thus, the shape of the fluid column does not change. In conclusion, the singularity of h in the previous model is, in fact, generated by gravity. The examination of the other symmetry reductions are left as exercises for the reader.

5.6.9 The Rayleigh Particle

The Rayleigh particle is similar to the Brownian particle. The difference between the two species is that the Brownian particle has a constant velocity, whereas the Rayleigh particle is characterized by the macroscopic law for the velocity V in the form of the damping law

$$\dot{V} = -\gamma V . \tag{5.76}$$

We assume that V is not too large. The Fokker-Planck (FP) equation related to the Rayleigh particle is discussed by Van Kampen [1981]. The equation of motion for the probability density P reads

```
Clear[P, V];
rayleigh = ∂t P[V, t] - γ (∂V (V P[V, t]) + k T ∂{V,2} P[V, t] / M) == 0;
rayleigh // LTF
```

$$P_t - \gamma \left(P + V P_V + \frac{k T P_{V,V}}{M}\right) == 0$$

This linear equation for the probability density P depending on the velocity V and time t describes a damped particle of mass M at a temperature T. The constants γ and k are the damping parameter and the Boltzmann constant, respectively. The independent variables in the equation can be scaled in such a way that the equation is free of any parameters [$\tau = \gamma t$ and $v = V\,(m/(kT))^{1/2}$]. The scaled equation is given by

```
rayleigh = ∂τ P[v, τ] - (∂v (v P[v, τ]) + ∂{v,2} P[v, τ]) == 0;
rayleigh // LTF
```

$$-P - v\,P_v + P_\tau - P_{v,v} == 0$$

This type of equation was examined by Cicogna and Vitali [1990]. We reproduce their results to demonstrate that the structure of the infinitesimals can be quite different from a polynomial and that *MathLie* is capable of finding these symmetries automatically. The infinitesimals read

```
infirayl = Infinitesimals[rayleigh, {P}, {v, τ}]; infirayl // LTF
```

$$\mathcal{F}_1 + v\,(\mathcal{F}_1)_v - (\mathcal{F}_1)_\tau + (\mathcal{F}_1)_{v,v} == 0$$

$$\xi_1 == E^{-\tau}\,k1 + E^{\tau}\,k2 + \frac{1}{2}\,E^{-2\tau}\,(k3 + E^{4\tau}\,k5)\,v$$

$$\xi_2 == -\frac{1}{2}\,E^{-2\tau}\,k3 + \frac{1}{2}\,E^{2\tau}\,k5 + k6$$

$$\phi_1 == \frac{1}{2}\,P\,(-E^{-2\tau}\,k3 + 2\,k4 - 2\,E^{\tau}\,k2\,v - E^{2\tau}\,k5\,v^2) + \mathcal{F}_1$$

The symmetry group is given by a six-dimensional discrete group and an infinite dimensional group determined by the function *free[1]* satisfying the original Fokker-Planck equation. The structure of the infinitesimals illustrates that *MathLie* is capable of deriving non-polynomial results.

The infinitesimals can be used to derive solutions of the FP equation. A simple solution is derived for the subgroup $k1 = 1$ and $k2 = 1$; the other constants are set to zero. The infinitesimals for this subgroup reduce to

```
inf1 = {{xi[1][v, τ, P], xi[2][v, τ, P]},
    {phi[1][v, τ, P]}} /. infirayl[[1]] /.
  {k1 → 1, k2 → 1, k3 → 0, k4 → 0, k5 → 0, k6 → 0, free[1][___] → 0}
```

$$\{\{E^{-\tau} + E^{\tau},\ 0\},\ \{-E^{\tau}\,P\,v\}\}$$

The corresponding similarity reduction of the FP equation follows from

```
red1 = LieReduction[rayleigh, {P}, {v, τ}, inf1[[1]], inf1[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$\tau - \zeta_1 == 0$$

$$E^{\frac{E^{2\tau} v^2}{2(1+E^{2\tau})}} P - F_1 == 0$$

$$E^{\frac{E^{2\zeta_1} v^2}{-2-2E^{2\zeta_1}}} \left(-F_1 + (F_1)_{\zeta_1} + E^{2\zeta_1} (F_1)_{\zeta_1}\right) == 0$$

We realize that the similarity variable is given by the temporal variable τ. The resulting first-order ODE containing analytic coefficients is solved by DSolve[]:

```
solr1 = DSolve[-F1[zeta1] + F1'[zeta1] + E^(2 zeta1) F1'[zeta1] == 0,
  F1, zeta1]
```

$$\left\{\left\{F1 \to \left(E^{-\frac{1}{2}\text{Log}[1+E^{2\#1}]+\#1} C[1]\&\right)\right\}\right\}$$

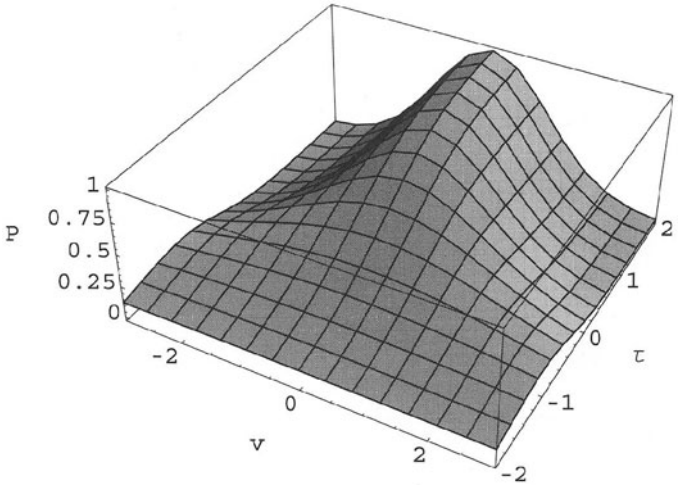
The solution of the FP equation in original coordinates results by inverting the similarity representation derived above:

```
sol1 = Solve[red1[[2, 1]] /. solr1, P] // Simplify
```

$$\left\{\left\{P \to \frac{E^{\frac{E^{2\tau} v^2}{-2-2E^{2\tau}}+\tau} C[1]}{\sqrt{1+E^{2\tau}}}\right\}\right\}$$

A picture for a fixed value of the constant of integration $C[1]$ is shown below. The figure demonstrates that the probability density P has a single maximum in the velocity domain. The amplitude of the probability density increases with increasing time.

```
Plot3D[P /. sol1[[1]] /. C[1] → 1, {v, -3, 3}, {τ, -2, 2},
 AxesLabel → {"v", "τ", "P"}]
```



A second solution with similar properties develops from the subgroup $k1 = k2 = k4 = 1$. The corresponding infinitesimals are

```
inf1 = {{xi[1][v, τ, P], xi[2][v, τ, P]},
    {phi[1][v, τ, P]}} /. infirayl[[1]] /.
  {k1 → 1, k2 → 1, k3 → 0, k4 → 1, k5 → 0, k6 → 0, free[1][___] → 0}
```

$$\left\{\{E^{-\tau} + E^{\tau},\ 0\},\ \left\{\frac{1}{2} P\,(2 - 2\,E^{\tau} v)\right\}\right\}$$

The similarity reduction of the FP equation is gained with

```
red2 = LieReduction[rayleigh, {P}, {v, τ}, inf1[[1]], inf1[[2]];
LTF[Flatten[red2]] /. zeta1 → ζ1
```

$$\tau - \zeta_1 == 0$$

$$E^{-\frac{E^{\tau} v}{1+E^{2\tau}} + \frac{E^{2\tau} v^2}{2\,(1+E^{2\tau})}}\ P - F_1 == 0$$

$$-F_1 - 2\,E^{2\zeta_1}\ F_1 + (F_1)_{\zeta_1} + 2\,E^{2\zeta_1}\ (F_1)_{\zeta_1} + E^{4\zeta_1}\ (F_1)_{\zeta_1} == 0$$

Again the similarity variable is τ. The similarity representation of the solution combines an unknown function F1 with an exponential depending on the velocity v and time τ. The related first-order ODE which F1 has to satisfy determines the similarity solution:

```
solr2 = DSolve[red2[[3, 1]], F1, zeta1]
```

$$\left\{\left\{F1 \to \left(E^{-\frac{1+E^{2\#1}}{2\,(1+2\,E^{2\#1}+E^{4\#1})} - \frac{(1+E^{2\#1})^2\,\mathrm{Log}[1+E^{2\#1}]}{2\,(1+2\,E^{2\#1}+E^{4\#1})} + \frac{(1+E^{2\#1})^2\,\#1}{1+2\,E^{2\#1}+E^{4\#1}}}\ C[1]\,\&\right)\right\}\right\}$$

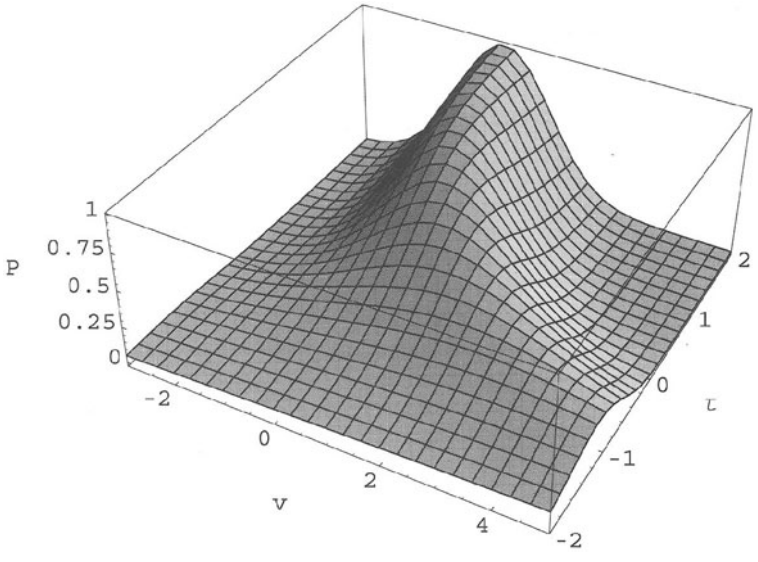
This solution reads, in original variables,

```
sol2 = Solve[red2[[2, 1]] /. solr2, P] // Simplify
```

$$\left\{\left\{P \to \frac{E^{\frac{-1+2\,E^{\tau} v - E^{2\tau} v^2 + 2\,(1+E^{2\tau})\,\tau}{2\,(1+E^{2\tau})}}\ C[1]}{\sqrt{1+E^{2\tau}}}\right\}\right\}$$

The plot of the solution shows that, contrary to the previous group, a small difference exists. The difference occurs for larger positive velocities and negative times τ. In this region, the probability density has a non-vanishing value.

```
Plot3D[P /. sol2[[1]] /. C[1] → 1, {v, -3, 5}, {τ, -2, 2},
 AxesLabel → {"v", "τ", "P"}, PlotPoints → 25]
```



A quite different solution of the FP equation follows if we set $k5 = k6 = 1$. The rest of the group parameters are set to zero. The similarity variable for this subgroup is a combination of the velocity v and the time τ. The infinitesimals are

```
inf1 = {{xi[1][v, τ, P], xi[2][v, τ, P]},
    {phi[1][v, τ, P]}} /. infirayl[[1]] /.
  {k1 → 0, k2 → 0, k3 → 0, k4 → 0, k5 → 1, k6 → 1, free[1][___] → 0}
```

$$\left\{\left\{\frac{1}{2}\,E^{2\tau}\,v,\ 1+\frac{E^{2\tau}}{2}\right\},\ \left\{-\frac{1}{2}\,E^{2\tau}\,P\,v^2\right\}\right\}$$

The similarity reduction of the original equations displays that the similarity solution for P is given by a function F1 which now has to satisfy a second-order equation:

```
red3 = LieReduction[rayleigh, {P}, {v, τ}, inf1[[1]], inf1[[2]]];
LTF[Flatten[red3]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$\frac{\sqrt{2+E^{2\tau}}}{v}-\zeta_1 == 0$$

$$E^{\frac{v^2}{2}}\,P-F_1 == 0$$

$$-\zeta_1\,(2\,F1_{\zeta_1}+2\,F1_{\zeta_1}\,\zeta_1^2+\zeta_1^3\,F1_{\zeta_1,\zeta_1}) == 0$$

The second-order ODE is identified by DSolve[] as

```
solr3 = DSolve[red3[[3, 1]], F1, zeta1]
```

$$\left\{\left\{\text{F1} \to \left(\text{C[2]} - \frac{1}{2}\sqrt{\pi}\,\text{C[1]}\,\text{Erfi}\left[\frac{1}{\#1}\right]\&\right)\right\}\right\}$$

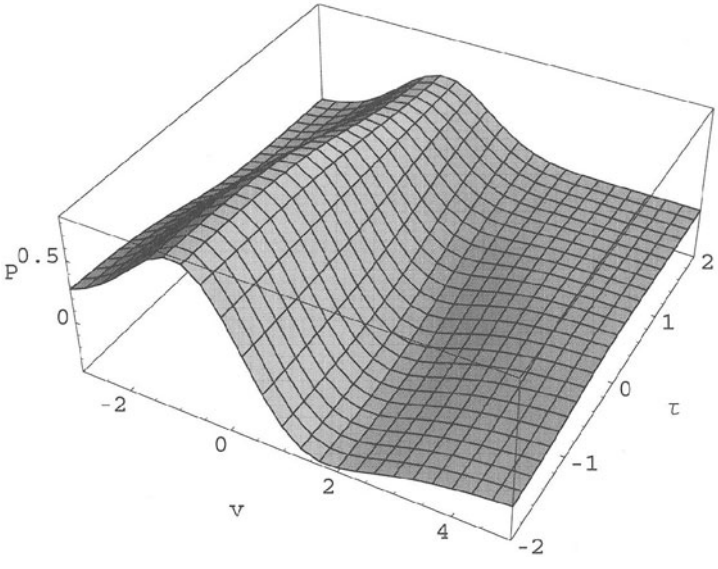
The result contains the special function Erfi[] which gives the imaginary error function erfi(z). In original variables, the result reads

```
sol3 = Solve[red3[[2, 1]] /. solr3, P] // Simplify
```

$$\left\{\left\{\text{P} \to \frac{1}{2}\,\text{E}^{-\frac{v^2}{2}}\left(2\,\text{C[2]} - \sqrt{\pi}\,\text{C[1]}\,\text{Erfi}\left[\frac{\text{v}}{\sqrt{2 + \text{E}^{2\tau}}}\right]\right)\right\}\right\}$$

The plot of this function demonstrates that the probability density is localized along a fixed axis parallel to the time axis. However, there are small variations in the velocity direction.

```
Plot3D[P /. sol3[[1]] /. {C[1] → 1, C[2] → 1 / 2}, {v, -3, 5},
 {τ, -2, 2}, AxesLabel → {"v", "τ", "P"}, PlotPoints → 25]
```



Another non-trivial similarity reduction for the FP equation follows from the subgroup *k3* = 1. The infinitesimals for this subgroup read

```
inf1 = {{xi[1][v, τ, P], xi[2][v, τ, P]},
    {phi[1][v, τ, P]}} /. infirayl[[1]] /.
  {k1 → 0, k2 → 0, k3 → 1, k4 → 0, k5 → 0, k6 → 0, free[1][___] → 0}
```

$$\left\{\left\{\frac{1}{2}\,E^{-2\tau}\,v,\ -\frac{1}{2}\,E^{-2\tau}\right\},\ \left\{-\frac{1}{2}\,E^{-2\tau}\,P\right\}\right\}$$

The reduced FP equation represents a second-order ODE. The similarity variable *zeta1* combines the time τ and the logarithm of the velocity v:

```
red4 = LieReduction[rayleigh, {P}, {v, τ}, inf1[[1]], inf1[[2]]];
LTF[Flatten[red4]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$\tau + \text{Log}[v] - \zeta_1 == 0$$
$$P\,v - F_1 == 0$$
$$-E^{3\tau}\,(2\,F_1 - 3\,(F_1)_{\zeta_1} + (F_1)_{\zeta_1,\zeta_1}) == 0$$

The solution of the second-order ODE is a combination of two exponentials

```
solr4 = DSolve[red4[[3, 1]], F1, zeta1]
```

$$\{\{F1 \to (E^{\#1}\,C[1] + E^{2\,\#1}\,C[2]\&)\}\}$$

In the original coordinates, the solution is linear in v and increases exponentially in time:

```
sol4 = Solve[red4[[2, 1]] /. solr4, P] // Simplify
```

$$\{\{P \to E^{\tau}\,(C[1] + E^{\tau}\,v\,C[2])\}\}$$

The solutions derived from different subgroups of the FP equation demonstrate that it is straightforward to uncover solutions with *MathLie* for a (1 + 1)-dimensional PDE.

5.6.10 Molecular Beam Epitaxy

Before discussing a detailed analysis of the growth equations of molecular beam epitaxy (MBE), let us describe the main relevant microscopic processes taking place on a crystal interface. The morphology of the interface is determined by the interplay among deposition, desorption, and surface diffusion. The term deposition means the sticking of an atom on the surface if it arrives from vapor. Crystals grow by atomic deposition. Desorption is the reverse effect, competing with deposition. Under desorption, we understand that atoms deposited on the surface leave the interface. When an atom is deposited on a surface, it forms bonds that must be broken before

desorption can occur. If only some of the bonds are broken, the atom gains the ability to move on the surface. From a microscopic point of view, surface diffusion is an activated process. The discrete positions of the atom are determined by the crystal lattice. For an atom on the surface to diffuse to the next lattice position, it must overcome the lattice potential existing between two neighboring positions. This excess energy required for diffusion is the microscopic origin of the lattice potential.

The physical mechanism that governs MBE is the surface diffusion of the deposited particles. For the moment, we assume that desorption is negligible. So we consider the scenario of atoms deposited on a surface, whereupon they diffuse. The goal is to find an equation for the form

$$\partial_t\, a(x, t) \;=\; F(a,\, x,\, t) \tag{5.77}$$

that describes the variation in the interface amplitude $a(x, t)$. Diffusion processes are directly connected with a macroscopic current. The local changes in the surface height are the result of the currents along the surface. Since we neglect desorption, the total number of atoms remains unchanged during the diffusion process and the current must obey a continuity equation:

$$\partial_t a(x, t) \;=\; -\,\mathrm{div}\big(\vec{j}(x, t)\big)\,. \tag{5.78}$$

On the other hand, the surface current is driven by the difference in the local chemical potential $\vec{j} \sim \mathrm{grad}(\mu(x, t))$. As we already mentioned, the diffusion is an activated process. The motion of an atom depends on the number of bonds that must be broken. The more neighbors an atom has, the lower the mobility. A measure of mobility is the local radius of curvature r. A simple assumption is that the chemical potential is proportional to $-1/r$, which, in turn, is proportional to $\mathrm{div}(\mathrm{grad}(a(x, t)))$. Hence, $\mu \sim -\Delta a(x, t)$, where Δ denotes the Laplacian. Combining the arguments given, we end up with the equation

$$\partial_t a(x, t) \;+\; \kappa\, \nabla^4\, a(x, t) \;=\; 0. \tag{5.79}$$

In MBE experiments, we have, on the other side, a flux $\phi = \phi(x, t)$ of atoms bombarding the surface. The flux is defined as the number of particles arriving on the unit surface in a unit time. At large length scales, the beam is homogeneous with an average intensity ϕ. Thus, the growth equation incorporating surface diffusion and deposition has the form

$$\partial_t a(x, t) \;+\; \kappa\, \nabla^4\, a(x, t) \;=\; \phi\,. \tag{5.80}$$

The growth equation incorporating statistical fluctuations in the flux was originally introduced independently by Wolf and Villain [1990] and Das Sarma and Tamborenea [1991]. Since we are only interested in the deterministic behavior of the model, we suppress the noise in the equations. If we now include the process of

desorption, we can write down a linear model for MBE. Let us assume with Villain that the desorption-dominated growth of the surface is governed by the difference between the average chemical potential in the vapor $\overline{\mu}$ and the local chemical potential on the surface $\mu(x, t)$. Thus, the temporal change in the amplitude a is given by $-\beta(\mu(x, t) - \overline{\mu})$. The complete deterministic model including deposition, desorption, and surface diffusion now reads

$$\partial_t a(x, t) + \kappa \nabla^4 a(x, t) = \phi + \alpha \nabla^2 a(x, t) + \beta \overline{\mu}. \tag{5.81}$$

Apart from the beam flux ϕ, we incorporated an additional flux $\beta\overline{\mu}$. Both terms can be abbreviated by $\Phi = \phi + \beta\overline{\mu}$:

$$\partial_t a(x, t) + \kappa \nabla^4 a(x, t) = \alpha \nabla^2 a(x, t) + \Phi. \tag{5.82}$$

This equation contains all physically relevant linear terms describing the growth of interfaces by MBE. The competition between the diffusion and desorption processes will determine the growth of the surface. The parameters κ, α, and Φ are assumed to be real and positive.

Another aspect to be considered in MBE is the influence of non-linearity. Non-linearity comes into play if we consider large domains in space. For such cases, the small gradients in a are not negligible. The corresponding terms of second- and third-order non-linearity are incorporated in the growth equation, which now reads

$$\begin{aligned} &\partial_t a(x, t) + \kappa \nabla^4 a(x, t) = \\ &\alpha \nabla^2 a(x, t) + \Phi + \gamma \nabla^2 (\nabla a(x, t))^2 + \lambda \nabla \cdot (\nabla a(x, t))^3, \end{aligned} \tag{5.83}$$

where we kept terms up to the fourth order in ∂_x. To gain some insight into the dynamic of this model, let us first examine the symmetries of the one-dimensional version. For this case, equation (5.83) reduces to a (1+1)-dimensional expression. The related equation for the amplitude

```
A = a[x, t];
```

reads in *Mathematica*

```
nMBE =
 ∂t A + κ ∂{x,4} A - α ∂{x,2} A - γ ∂{x,2} (∂x A)^2 - λ ∂x (∂x A)^3 - Φ == 0;
nMBE // LTF
```

```
-Φ + a_t - α a_{x,x} - 3 λ a_x^2 a_{x,x} - γ (2 a_{x,x}^2 + 2 a_x a_{x,x,x}) + κ a_{x,x,x,x} == 0
```

where κ, α, γ, Φ, and λ are real constants. The symmetries for this general non-linear model follow by

```
inMBE = Infinitesimals[nMBE, a, {x, t}, {κ, α, λ, γ, Φ}];
inMBE // LTF
```

$$\phi_1 == k1$$
$$\xi_1 == k2$$
$$\xi_2 == k3$$

representing a three-dimensional group containing only translations. The reduction of this (1+1)-dimensional model to an ODE follows by

```
red1 = LieReduction[nMBE, {a}, {x, t}, {1, c}, {1}];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$t - c\,x - \zeta_1 == 0$$
$$a - x - F_1 == 0$$
$$-\Phi + F1_{\zeta_1} - c^2\,\alpha\,F1_{\zeta_1,\zeta_1} - 3\,c^2\,\lambda\,F1_{\zeta_1,\zeta_1} + 6\,c^3\,\lambda\,F1_{\zeta_1}\,F1_{\zeta_1,\zeta_1} - 3\,c^4\,\lambda\,F1_{\zeta_1}^2\,F1_{\zeta_1,\zeta_1} - 2\,c^4\,\gamma\,F1_{\zeta_1,\zeta_1}^2 + 2\,c^3\,\gamma\,F1_{\zeta_1,\zeta_1,\zeta_1} - 2\,c^4\,\gamma\,F1_{\zeta_1}\,F1_{\zeta_1,\zeta_1,\zeta_1} + c^4\,\kappa\,F1_{\zeta_1,\zeta_1,\zeta_1,\zeta_1} == 0$$

describing a moving wave solution governed by a non-linear fourth-order ODE. This reduced equation can be integrated once:

```
Clear[H];
intMBE = ∫ (red1[[3, 1, 1]]) dzeta1 == C[1] /. {F1 → H, zeta1 → ζ};
intMBE // LTF
```

$$H - \zeta\,\Phi - C[1] - c^2\,(\alpha + 3\,\lambda)\,H_\zeta + 3\,c^3\,\lambda\,H_\zeta^2 - c^4\,\lambda\,H_\zeta^3 - 2\,c^3\,\gamma\,(-1 + c\,H_\zeta)\,H_{\zeta,\zeta} + c^4\,\kappa\,H_{\zeta,\zeta,\zeta} == 0$$

The symmetries of the resulting third-order ODE are

```
infinirMBE = Infinitesimals[intMBE, H, ζ,
  {α, κ, γ, λ, Φ, c, C[1]}, SubstitutionRules → {∂{ζ,3} H[ζ]}];
infinirMBE // LTF
```

$$\phi_1 == k1\,\Phi$$
$$\xi_1 == k1$$

The result shows that the reduced equation following from the general model is not solvable by Lie's method.

In the following, we will discuss two limiting models, including surface diffusion in connection with non-linearity and desorption with non-linearity. Both models include a term describing deposition.

5.6.10.1 Surface Diffusion with Nonlinearity

If we choose in the general model (5.83) $\alpha = 0$ and $\lambda = 0$, we get a model including only surface diffusion and quadratic non-linearity in the gradient of a. The equation of motion reduces to

```
m8MBE = nMBE /. {α → 0, λ → 0}; m8MBE // LTF
```

$$-\Phi + a_t - \gamma\,(2\,a_{x,x}^2 + 2\,a_x\,a_{x,x,x}) + \kappa\,a_{x,x,x,x} == 0$$

The symmetry group of this special model

```
m8infi = Infinitesimals[m8MBE, a, {x, t}, {κ, γ, Φ}];
m8infi // LTF
```

$$\phi_1 == k1 + k4\,t\,\Phi$$
$$\xi_1 == k2 + \frac{k4\,x}{4}$$
$$\xi_2 == k3 + k4\,t$$

contains translations and a scaling. Let us examine the scaling group

```
infi = {{xi[1][x, t, a], xi[2][x, t, a]},
    {phi[1][x, t, a]}} /. m8infi /.
  {k1 → 0, k2 → 0, k3 → 0, k4 → 1}
```

$$\left\{\left\{\frac{x}{4}, t\right\}, \{t\,\Phi\}\right\}$$

This kind of infinitesimals reduce the original equation to a fourth-order ODE:

```
red1 = LieReduction[m8MBE, {a}, {x, t}, infi[[1]], infi[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$\frac{t}{x^4} - \zeta_1 == 0$$
$$a - t\,\Phi - F_1 == 0$$
$$F1_{\zeta_1} + 840\,\kappa\,F1_{\zeta_1}\,\zeta_1 - 1760\,\gamma\,F1_{\zeta_1}^2\,\zeta_1^2 + 3120\,\kappa\,\zeta_1^2\,F1_{\zeta_1,\zeta_1} -$$
$$3200\,\gamma\,F1_{\zeta_1}\,\zeta_1^3\,F1_{\zeta_1,\zeta_1} - 512\,\gamma\,\zeta_1^4\,F1_{\zeta_1,\zeta_1}^2 + 1920\,\kappa\,\zeta_1^3\,F1_{\zeta_1,\zeta_1,\zeta_1} -$$
$$512\,\gamma\,F1_{\zeta_1}\,\zeta_1^4\,F1_{\zeta_1,\zeta_1,\zeta_1} + 256\,\kappa\,\zeta_1^4\,F1_{\zeta_1,\zeta_1,\zeta_1,\zeta_1} ==$$
$$0$$

Interestingly, the resulting ODE does not contain the flux term Φ. This term is completely separated in the similarity representation of the solution by $a = \Phi t + F_1(t/x^4)$. The unknown function F_1 has to satisfy the fourth-order ODE,

which cannot be solved symbolically. However, we can derive a numerical solution if we add four initial conditions to the ODE:

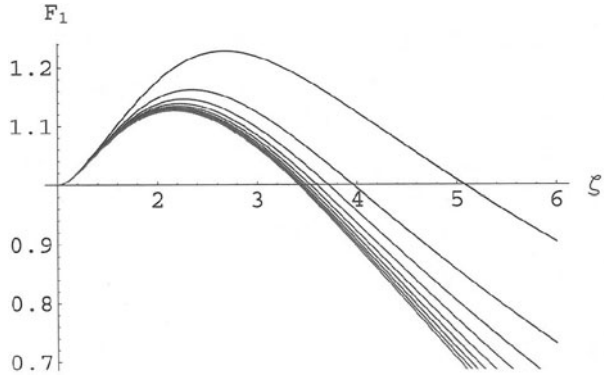
```
neq = Join[red1[[3]],
    {F1[1] == 1, F1'[1] == 0, F1''[1] == 3 / 2, F1⁽³⁾[1] == -10}] /.
  {γ → 1};
```

The numerical solution for a fixed value of γ and nine different values of κ follows by applying NDSolve[] to the equations

```
nsol =
 Table[NDSolve[neq /. κ → i, F1, {zeta1, 1, 15}], {i, .2, 1, .1}];
```

These solutions are plotted below.

```
color = Table[RGBColor[i, 0.1, 0.3], {i, .2, 1, .1}];

Plot[Evaluate[F1[ζ] /. nsol], {ζ, 1, 6}, PlotStyle → color,
 AxesLabel → {"ζ", "F₁"}]
```

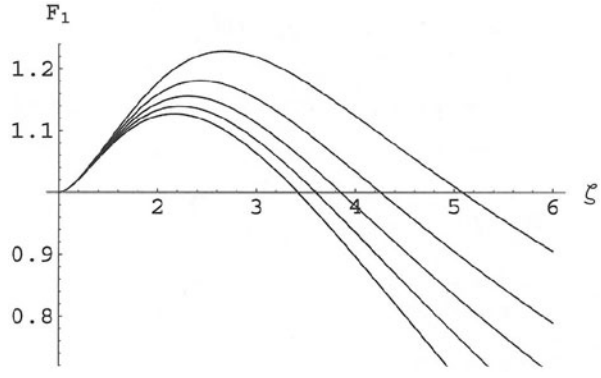


If we, on the other hand, fix the value of κ and vary γ, we can examine the influence of γ on the solution

```
neq = Join[red1[[3]],
    {F1[1] == 1, F1'[1] == 0, F1''[1] == 3 / 2, F1⁽³⁾[1] == -10}] /.
  {κ → 1};

nsol =
 Table[NDSolve[neq /. γ → i, F1, {zeta1, 1, 15}], {i, 1, 5, 1}];
```

The resulting solutions are shown in the following plot:

```
Plot[Evaluate[F1[ζ] /. nsol], {ζ, 1, 6}, PlotStyle → color,
 AxesLabel → {"ζ", "F₁"}]
```



We realize that the influence of the non-linear terms represented by γ increases if γ increases. The given solutions are by no means complete. We only get a caricature of the solution manifold by a numerical solution. However, for some practical applications, this solution may be sufficient to solve a specific problem.

5.6.10.2 Desorption with Non-linearity

The general model of MBE reduces to a model containing only effects of desorption and deposition if we set the constants κ and γ equal to zero:

```
m3MBE = nMBE /. {κ → 0, γ → 0}; m3MBE // LTF
```

$$-\Phi + a_t - \alpha\, a_{x,x} - 3\,\lambda\, a_x^2\, a_{x,x} == 0$$

The equation of motion is a non-linear second-order PDE with a constant flux Φ responsible for deposition. The equation of motion is connected with a non-linear diffusion equation including convective effects of the material in addition to the diffusive effects. The symmetries of this equation follow by

```
m3infi = Infinitesimals[m3MBE, a, {x, t}, {α, λ, Φ}];
m3infi // LTF
```

$$\phi_1 == k3 + \frac{k4\ (a + t\,\Phi)}{\Phi}$$

$$\xi_1 == k1 + \frac{k4\ x}{\Phi}$$

$$\xi_2 == k2 + \frac{2\ k4\ t}{\Phi}$$

describing a four-dimensional finite group with scaling and translation symmetries. We observe that the constant flux Φ is connected with the scaling symmetry. We will examine the solutions for the non-linear deposition model for this symmetry. The infinitesimals for the scaling subgroup read

```
infi = {{xi[1][x, t, a], xi[2][x, t, a]},
    {phi[1][x, t, a]}} /. m3infi /.
  {k1 → 0, k2 → 0, k3 → 0, k4 → 1}
```

$$\left\{\left\{\frac{x}{\Phi}, \frac{2\,t}{\Phi}\right\}, \left\{\frac{a + t\,\Phi}{\Phi}\right\}\right\}$$

The corresponding reduction follows from

```
red1 = LieReduction[m3MBE, {a}, {x, t}, infi[[1]], infi[[2]]];
LTF[Flatten[red1]] /. zeta1 → ζ1
```

$$\frac{t}{x^2} - \zeta_1 == 0$$

$$-\frac{-a + t\,\Phi}{x} - F_1 == 0$$

$$(F_1)_{\zeta_1} - 2\,\alpha\,\zeta_1\,(F_1)_{\zeta_1} - 6\,\lambda\,F_1^2\,\zeta_1\,(F_1)_{\zeta_1} + 24\,\lambda\,F_1\,\zeta_1^2\,(F_1)_{\zeta_1}^2 - 24\,\lambda\,\zeta_1^3\,(F_1)_{\zeta_1}^3 - 4\,\alpha\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1} - 12\,\lambda\,F_1^2\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1} + 48\,\lambda\,F_1\,\zeta_1^3\,(F_1)_{\zeta_1}\,(F_1)_{\zeta_1,\zeta_1} - 48\,\lambda\,\zeta_1^4\,(F_1)_{\zeta_1}^2\,(F_1)_{\zeta_1,\zeta_1} == 0$$

In the similarity reduction, the flux Φ is separated from the similarity function F_1. This function has to satisfy a non-linear second-order ODE which is not symbolically solvable by DSolve[]. The examination of the symmetry reveals that this equation does not allow any point symmetry and thus Lie's integration strategy terminates:

```
Infinitesimals[red1[[3]], F1, zeta1,
  {λ, α}, SubstitutionRules → {∂{zeta1,2} F1[zeta1]}] //
 LTF
```

$$\phi_1 == 0$$
$$\xi_1 == 0$$

The only way to study the equation is a numerical integration. Adding initial conditions to the second-order ODE and fixing the parameters,

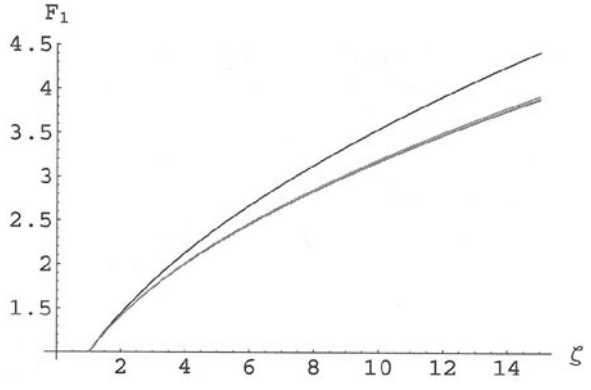
```
neq = Join[red1[[3]], {F1[1] == 1, F1'[1] == 1 / 2}] /. {λ → 1};
```

allows us the numerical integration. We first fixed λ describing the non-linear influence in the equation and changed α over a large range:

```
nsol = Table[NDSolve[neq /. α → i, F1, {zeta1, 1, 15}],
  {i, 1, 50, 10}];
```

The resulting solutions for different α and identical initial conditions are plotted below.

```
Plot[Evaluate[F1[ζ] /. nsol],
 {ζ, 1, 15}, PlotStyle → Table[Hue[i], {i, 0, .5, .1}],
 AxesLabel → {"ζ", "F1"}]
```



We observe that the solution shows an increasing behavior in ζ. By increasing the dispersive strength α in the ODE, the solutions decrease in their values and tend to a limiting curve. The other behavior studied under identical initial conditions is the variation of the non-linear strength λ at a fixed value of α. The equations plus initial conditions are created by

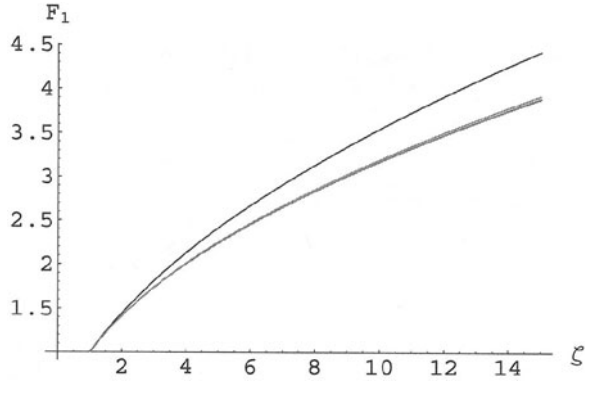
```
neq = Join[red1[[3]], {F1[1] == 1, F1'[1] == 1 / 2}] /. {α → 1};
```

The variation of λ in the integration process delivers the solutions

```
nsol = Table[NDSolve[neq /. λ → i, F1, {zeta1, 1, 15}],
  {i, 1, 50, 10}];
```

which are plotted over ζ by

```
Plot[Evaluate[F1[ζ] /. nsol],
 {ζ, 1, 15}, PlotStyle → Table[Hue[i], {i, 0, .5, .1}],
 AxesLabel → {"ζ", "F1"}]
```

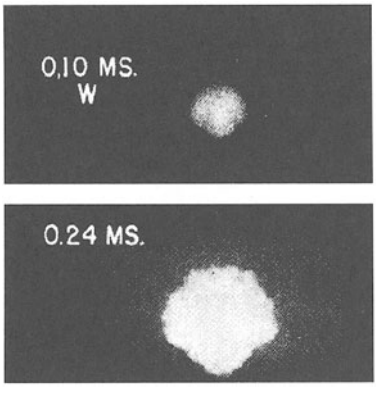


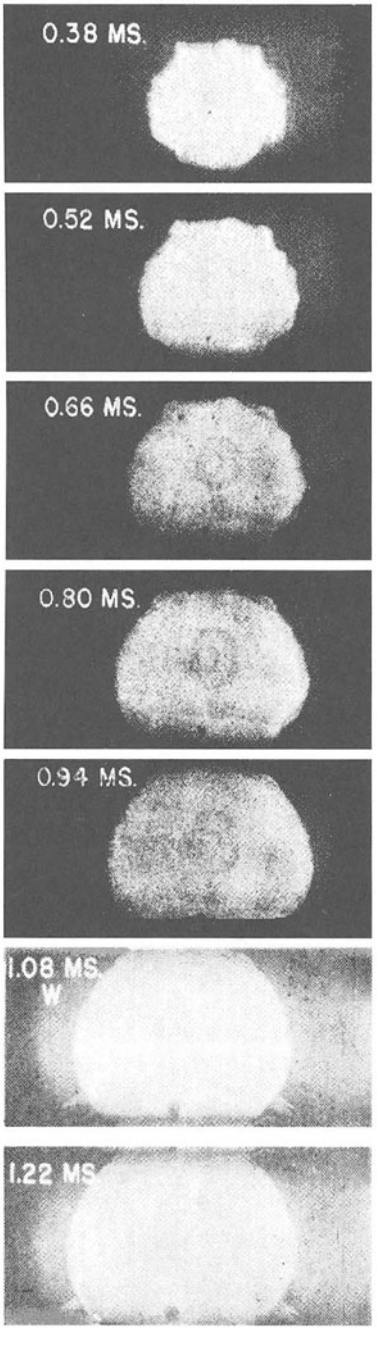
We observe again the same behavior as in the case of the α variation. However, the decrease in the solution is much smaller by changing the strength of the non-linearity.

5.6.11 The First Atomic Explosion

In March 1950, Sir Geoffrey Taylor [1950] published two papers which examined the first atomic explosion in 1945 in New Mexico. The author concludes that a similarity analysis of the experiment is in excellent agreement with the theory and can be used to calculate the energy release during the explosion. The information on the total release of energy was a well guarded secret of the U.S. government in these days. The paper by Taylor was therefore classified when the theoretical investigations were made. However, the publication 5 years later resolved this secret and made the results on energy release public contrary to the intention of the U.S. government. The results on the energy release caused much embarrassment in American government circles. The flaw of the government was that motion pictures recorded by Mack [1947] became unclassified while the energy release was considered top secret. These pictures contained not only the explosion but also a time record which allowed an estimation of the physical quantities.

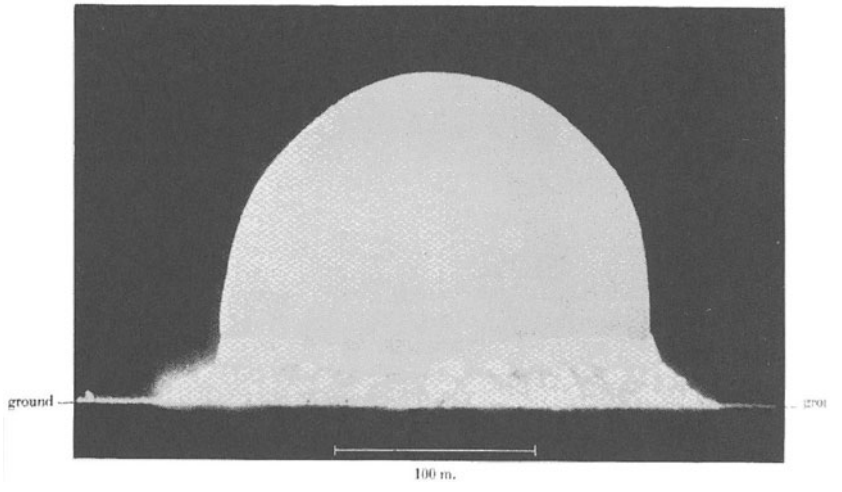
How such an estimation can be carried out is the subject of the present example. First, let us recall the sequence of pictures which were used by Sir Geoffrey Taylor to carry out the calculations. We took these pictures from the work of Taylor [1950]. They demonstrate the evolution of the blast in the first 2 ms. The animation capabilities of *Mathematica* empower us to follow the explosion at the desk.



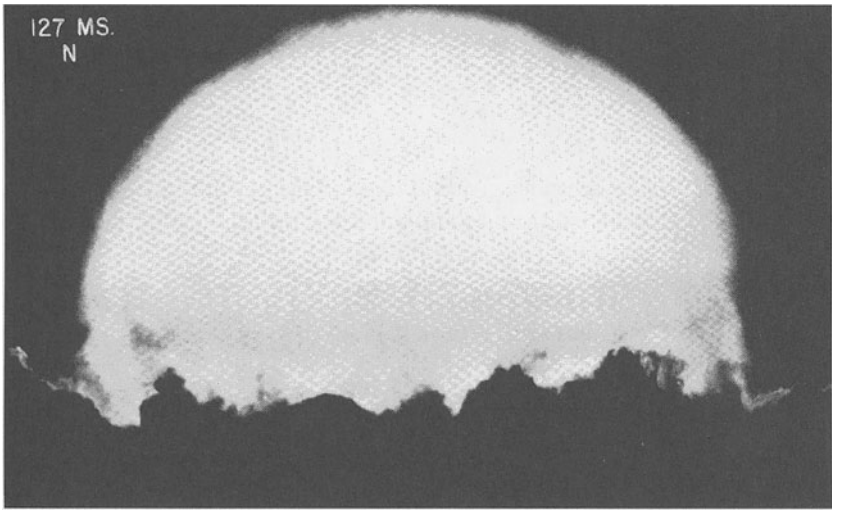

0.38 MS.
0.52 MS.
0.66 MS.
0.80 MS.
0.94 MS.
1.08 MS.
W
1.22 MS.

The main observation we make is that the thermal wave expands from a point to a spherical object. The motion of the gas was assumed spherically symmetric. This simplifying assumption received excellent confirmation in the first atomic test.

The picture 15 ms after the explosion looks like a very large mushroom. This photograph of the fire ball of the atomic explosion in New Mexico confirms the spherical symmetry of the gas motion in an excellent way.



The development of the gas after 127 ms is still nearly spherical. However, the region near the ground is more disturbed.



The pictures above were taken from G.I. Taylor and originally recorded by Mack in 1945. Sir Geoffrey Taylor used in his theoretical considerations three basic equations of motion describing the evolution of the pressure p, the density ρ, and the radial velocity u. The three equations are the equation of continuity, the Euler equation for the velocity, and the equation of state for a polytropic medium. The third equation expresses the fact that the entropy is constant along the path of a particle which generally is not the case, as Courant and Friederichs [1948] remark. According to the spherical symmetry of the problem Taylor used only the radial component of these quantities. The three equations of motion for the density ρ, the velocity field u, and the pressure p are

$$\rho_t + u\rho_r + \rho\left(u_r + \frac{2u}{r}\right) = 0, \tag{5.84}$$

$$u_t + u u_r + \frac{p_0 p_r}{\rho} = 0, \tag{5.85}$$

$$(p\rho^{-\gamma})_t + u(p\rho^{-\gamma})_r = 0. \tag{5.86}$$

In *Mathematica* we first define the three variables by

```
U = u[r, t]; Rh = ρ[r, t]; P = p[r, t];
```

The left-hand side of the three equations of motion are then collected in a list:

```
taylor = {∂t Rh + U ∂r Rh + Rh (∂r U + 2 U / r),
  ∂t U + U ∂r U + p0 ∂r P / Rh, ∂t (P Rh^-γ) + U ∂r (P Rh^-γ)};
taylor // LTF
```

$$\rho\left(\frac{2u}{r} + u_r\right) + u\rho_r + \rho_t == 0$$

$$\frac{p0\, p_r}{\rho} + u u_r + u_t == 0$$

$$\rho^{-\gamma} p_t + u\left(\rho^{-\gamma} p_r - p\gamma\rho^{-1-\gamma}\rho_r\right) - p\gamma\rho^{-1-\gamma}\rho_t == 0$$

The equations contain two parameters, γ and p_0, describing the ratio of the specific heats and the pressure of the undisturbed atmosphere. A symmetry analysis of these equations gives us the result

```
itaylor = Infinitesimals[taylor, {ρ, u, p}, {r, t}, {p0, γ},
  SubstitutionRules → {∂t ρ[r, t], ∂t u[r, t], ∂t p[r, t]}];
itaylor // LTF
```

$$\phi_1 == (2\,k2 - 2\,k3 - k4)\,\rho$$
$$\phi_2 == (-k2 + k3)\,u$$
$$\phi_3 == -k4\,p$$
$$\xi_1 == k3\,r$$
$$\xi_2 == k1 + k2\,t$$

The three equations permit a four-dimensional discrete symmetry group. The main symmetry is a scaling symmetry for all variables. In addition, there is a symmetry of translation in time. For the moment, we concentrate on the scaling symmetry with infinitesimals for the independent and dependent variables:

```
inf1 = {{xi[1][r, t, ρ, u, p],
     xi[2][r, t, ρ, u, p]}, {phi[1][r, t, ρ, u, p],
     phi[2][r, t, ρ, u, p], phi[3][r, t, ρ, u, p]}} /. itaylor /.
  {k1 → 0, k2 → 1, k3 → α, k4 → c}
```

$$\{\{r\,\alpha,\ t\},\ \{(2-c-2\,\alpha)\,\rho,\ u\,(-1+\alpha),\ -c\,p\}\}$$

The group parameters are chosen in such a way that we can find the parameters α and c in accordance with the measurements carried out by Taylor. The reduction of the equations of motion follows by

```
rtaylor = LieReduction[taylor,
   {ρ, u, p}, {r, t}, inf1[[1]], inf1[[2]]] // PowerExpand;
LTF[Flatten[rtaylor]] /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$r^{-1/\alpha}\,t - \zeta_1 == 0$$

$$r^{-\frac{2-c-2\,\alpha}{\alpha}}\,\rho - F_1 == 0$$

$$r^{-\frac{-1+\alpha}{\alpha}}\,u - F_2 == 0$$

$$p\,r^{c/\alpha} - F_3 == 0$$

$$\frac{t\,F_1\,F_2}{\zeta_1} - \frac{c\,t\,F_1\,F_2}{\zeta_1} + \frac{t\,\alpha\,F_1\,F_2}{\zeta_1} - t\,F_2\,(F_1)_{\zeta_1} + \frac{t\,\alpha\,(F_1)_{\zeta_1}}{\zeta_1} - t\,F_1\,(F_2)_{\zeta_1} == 0$$

$$t^{\alpha}\,\zeta_1^{-\alpha}\left(-\frac{t\,F_1\,F_2^2}{\zeta_1} + \frac{t\,\alpha\,F_1\,F_2^2}{\zeta_1} - \frac{c\,p0\,t\,F_3}{\zeta_1} - t\,F_1\,F_2\,(F_2)_{\zeta_1} + \frac{t\,\alpha\,F_1\,(F_2)_{\zeta_1}}{\zeta_1} - p0\,t\,(F_3)_{\zeta_1}\right) == 0$$

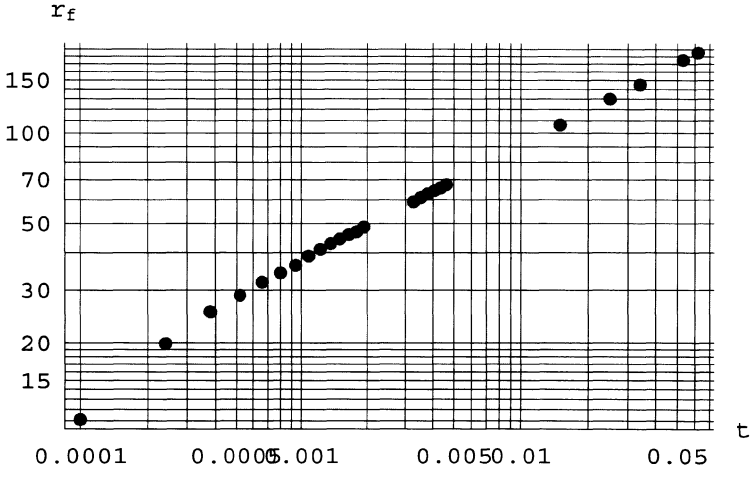
$$-\frac{c\,t\,F_1\,F_2\,F_3}{\zeta_1} - \frac{2\,t\,\gamma\,F_1\,F_2\,F_3}{\zeta_1} + \frac{c\,t\,\gamma\,F_1\,F_2\,F_3}{\zeta_1} + \frac{2\,t\,\alpha\,\gamma\,F_1\,F_2\,F_3}{\zeta_1} + t\,\gamma\,F_2\,F_3\,(F_1)_{\zeta_1} - \frac{t\,\alpha\,\gamma\,F_3\,(F_1)_{\zeta_1}}{\zeta_1} - t\,F_1\,F_2\,(F_3)_{\zeta_1} + \frac{t\,\alpha\,F_1\,(F_3)_{\zeta_1}}{\zeta_1} == 0$$

Thus, the original equations reduce to a coupled system of first-order ODEs. This set of equations contains the new variable $zeta1 = r^{-1/\alpha}\, t$, where ζ_1 is a constant. This constant is determined by the radial coordinate r and the time t. To determine the exponent α in our analysis, we use the measurements of Taylor for the radius r_f of the explosion front. Since $\zeta_1 = t\, r^{-1/\alpha}$ is a constant, we can determine the exponent α in a double logarithmic plot of the radius r_f versus time $\log(r_f) = (\log(t) - \log(\zeta_1))/\alpha$. The slope in the log-log plot is directly related to $1/\alpha$. The data we need for this kind of analysis are tabulated in Taylor's paper and are reproduced here. The first figure of the data set denotes time in seconds and the second the radius in meters:

```
taylorData = {{0.1 10^-3, 11.1},
  {0.24 10^-3, 19.9}, {0.38 10^-3, 25.4}, {0.52 10^-3, 28.8},
  {0.66 10^-3, 31.9}, {0.80 10^-3, 34.2}, {0.94 10^-3, 36.3},
  {1.08 10^-3, 38.9}, {1.22 10^-3, 41.0}, {1.36 10^-3, 42.8},
  {1.50 10^-3, 44.4}, {1.65 10^-3, 46.0}, {1.79 10^-3, 46.9},
  {1.93 10^-3, 48.7}, {3.26 10^-3, 59.0}, {3.53 10^-3, 61.1},
  {3.80 10^-3, 62.9}, {4.07 10^-3, 64.3}, {4.34 10^-3, 65.6},
  {4.61 10^-3, 67.3}, {15.0 10^-3, 106.5}, {25.0 10^-3, 130.0},
  {34.0 10^-3, 145.0}, {53.0 10^-3, 175.0}, {62.0 10^-3, 185.0}};
<< Graphics`Graphics`
```

The double logarithmic plot demonstrates the linear relation between the fire-ball radius and the time elapsed since the ignition.

```
pl1 = LogLogListPlot[taylorData, GridLines → Automatic,
  Frame → False, Prolog → {PointSize[0.02]},
  PlotStyle → RGBColor[1, 0, 0], AxesLabel → {"t", "r_f"}]
```



The scaling exponent α can be estimated by fitting a linear relation on the logarithmic data:

```
logData = Take[ Log[10, taylorData], {1, Length[taylorData]}];
```

The fit of the *logData* follows by

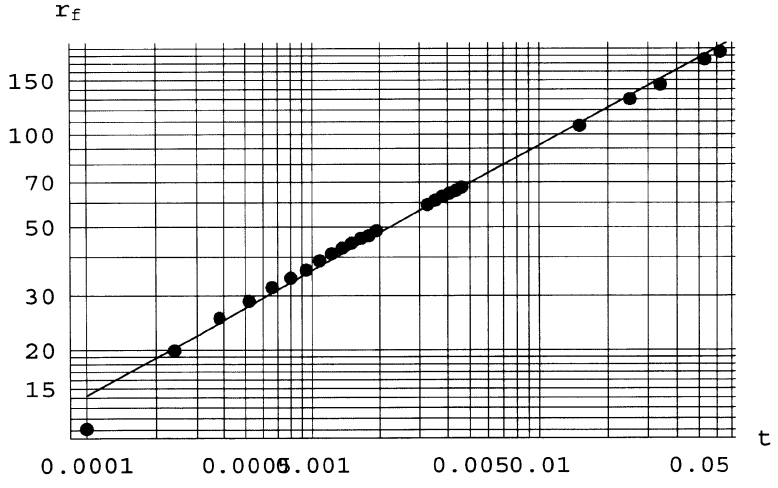
```
fu = Fit[logData, {1, t}, t]
```

```
2.77674 + 0.405823 t
```

The result is a relation connecting time t with the radius r_f in a logarithmic representation. The slope of the straight line is given by $\beta = 0.405 = 1/\alpha$. The exponent α thus takes approximately the value $\alpha = 5/2$ within an error of 1.4%.

Combining both the measurement and the fit in a common picture shows the excellent agreement between experiment and theory:

```
Show[pl1, Plot[fu, {t, -4, -1},
  PlotStyle → RGBColor[0, 0, 1], DisplayFunction → Identity],
 DisplayFunction → $DisplayFunction]
```



The plot shows that the scaling relation $\zeta_1 = t\, r^{-2/5}$ is satisfied over a range of about three decades in time.

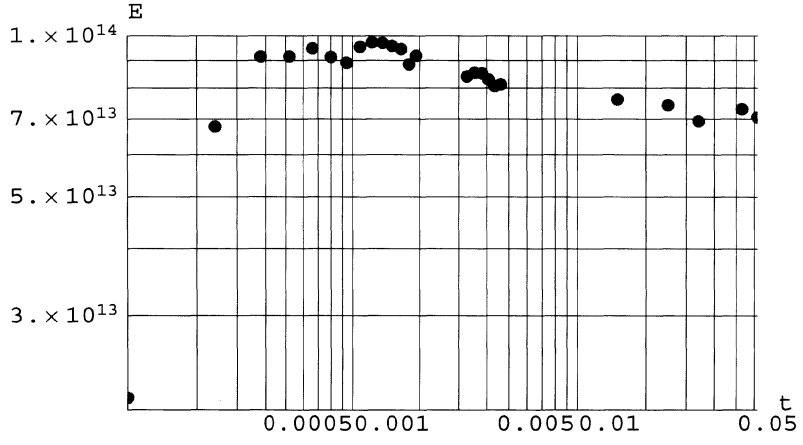
The exponent α was calculated by Taylor by different reasonings. He used a dimensional analysis of the problem and ended up with the value $\alpha = 5/2$ for the scaling exponent. The idea behind a dimensional analysis is that a physical quantity is expressed by other physical quantities which mainly govern the process. The

atomic explosion is mainly determined by the total energy E released at ignition, the density of the surrounding air ρ_0, and the time t. The dimensions of the governing quantities in the length, mass, and time (LMT) system are $[E] = \mathrm{ML}^2\,T^{-2}$, $[t] = T$, and $\rho_0 = \mathrm{ML}^{-3}$. The dimension of the radius r_f expressed by these quantities is $[r_f] = [E]^{1/5}[t]^{2/5}[\rho_0]^{-1/5}$. Since E and ρ_0 are constants, we find $\log(r_f) = \frac{2}{5}\left(\log(t) - \log(\frac{K\rho_0}{E})^{1/2}\right)$. Comparing this result with the formula derived from the similarity analysis, we can identify $\zeta = (K\,\rho_0 / E)^{1/2}$. This relation also suggests that

$$E = K\,\rho_0\,r_f^5\,t^{-2} \tag{5.87}$$

is a constant in time. Assuming that the density $\rho_0 = 1.25\ \mathrm{kg/m^3}$and that K, depending on the pressure, is near unity, we can estimate the energy from Taylor's data. The following plot shows an overview of the energy calculated by (5.87) for all data points:

```
LogLogListPlot[Map[({#[[1]], #[[1]]^-2 #[[2]]^5 * 1.25})&, taylorData],
 GridLines → Automatic,
 Frame → False, Prolog → {PointSize[0.02]},
 PlotStyle → RGBColor[1, 0, 0], AxesLabel → {"t", "E"},
 PlotRange → {{0.0001, 0.062}, {2.0 * 10^13, 1.0 * 10^14}}]
```



Despite the first point, the energy values oscillate around a mean energy value of about

```
<< Statistics`DescriptiveStatistics`

Mean[Map[(#[[1]]^-2 #[[2]]^5 * 1.25)&, taylorData]]
```

8.28254×10^{13}

The value calculated in joules corresponds to a T.N.T. equivalent of about 19,488 tons. The other information contained in Taylor's model is the dynamic behavior of the density ρ, the velocity u, and the pressure p. Inserting the value of α into the similarity reduction, we find the governing equations for these quantities:

```
LTF[(rtaylor // Flatten) /. {α → 5 / 2} // Simplify] /. zeta1 → ζ1
```

$$\frac{t}{r^{2/5}} - \zeta_1 == 0$$

$$r^{\frac{2\,(3+c)}{5}}\,\rho - F_1 == 0$$

$$\frac{u}{r^{3/5}} - F_2 == 0$$

$$p\,r^{2\,c/5} - F_3 == 0$$

$$-\frac{t\,((-5 + 2\,F_2\,\zeta_1)\,(F_1)_{\zeta_1} + F_1\,((-7 + 2\,c)\,F_2 + 2\,\zeta_1\,(F_2)_{\zeta_1}))}{2\,\zeta_1} == 0$$

$$-\frac{t^{7/2}\,(c\,p0\,F_3 - \frac{1}{2}\,F_1\,(3\,F_2^2 + 5\,(F_2)_{\zeta_1} - 2\,F_2\,\zeta_1\,(F_2)_{\zeta_1}) + p0\,\zeta_1\,(F_3)_{\zeta_1})}{\zeta_2^{7/2}}$$

$$== 0$$

$$\frac{t\,\gamma\,F_3\,(-\frac{5}{2} + F_2\,\zeta_1)\,(F_1)_{\zeta_1} + t\,F_1\,((c\,(-1+\gamma) + 3\,\gamma)\,F_2\,F_3 + (\frac{5}{2} - F_2\,\zeta_1)}{\zeta_1}$$

$$== 0$$

The set of equations contain common factors which are eliminated in the following representation of the first-order coupled ODEs:

```
eq = {((5 - 2 zeta1 F2[zeta1]) F1'[zeta1] +
      F1[zeta1] ((7 - 2 c) F2[zeta1] - 2 zeta1 F2'[zeta1])) == 0,
  F1[zeta1]
     (-3 F2[zeta1]^2 - 5 F2'[zeta1] + 2 zeta1 F2[zeta1] F2'[zeta1]) +
    2 p0 (c F3[zeta1] + zeta1 F3'[zeta1]) == 0,
  γ (-5 + 2 zeta1 F2[zeta1]) F3[zeta1] F1'[zeta1] +
    F1[zeta1] (5 F3'[zeta1] + 2 F2[zeta1] (-c F3[zeta1] +
          3 γ F3[zeta1] + c γ F3[zeta1] - zeta1 F3'[zeta1])) ==
   0};
```

This set of equations contains parameters describing the pressure of air p_0, the ratio of the specific heats γ, and a group parameter c. Inserting numerical values for these three quantities allows us to integrate the ODEs numerically:

```
eq1 = eq /. {c → 1, γ → 1, p0 → 1};
```

For a numerical integration, we need initial conditions for the density ρ, the velocity field u, and the pressure p:

```
eq2 = Join[eq1, {F1[0] == 1.25, F2[0] == 1, F3[0] == 1}];
```

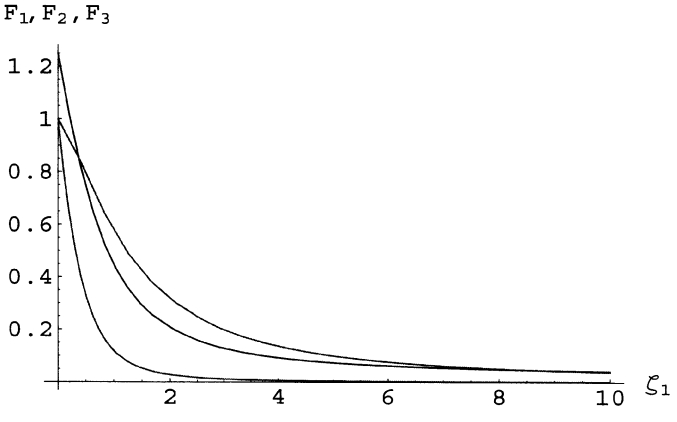
The integration for ζ_1 in the range $0 \leq \zeta_1 \leq 10$ delivers the solution by an interpolated function:

```
nsol = NDSolve[eq2, {F1, F2, F3}, {zeta1, 0, 10}]
```

```
{{F1 → InterpolatingFunction[{{0., 10.}}, <>],
  F2 → InterpolatingFunction[{{0., 10.}}, <>],
  F3 → InterpolatingFunction[{{0., 10.}}, <>]}}
```

The functions are represented by Plot[] for the three variables F_1, F_2, and F_3:

```
Plot[Evaluate[{F1[ζ], F2[ζ], F3[ζ]} /. nsol],
 {ζ, 0, 10}, PlotStyle → {RGBColor[0, 0, 0.996109],
   RGBColor[0.996109, 0, 0], RGBColor[0, 0.500008, 0]},
 PlotRange → All, AxesLabel → {"ζ1", "F1,F2,F3"}]
```



The plot shows that all three quantities decay in ζ_1. This behavior is expected since the total amount of energy is released into free space. In conclusion, we not only estimated the released thermal energy of the explosion but have also the spatial and temporal decay of the physical quantities available.

6

Non-classical Symmetries of Partial Differential Equations

6.1. Introduction

The non-classical method of symmetry analysis is an extension of Lie's classical method. This non-classical method was compiled by Bluman and Cole [1969] in connection with the analysis of the heat equation. This method allows us to derive another type of solutions which are different from solutions derived from Lie's procedure. The method extends the classical approach to find solutions for linear and non-linear PDEs. This chapter contains the theoretical background of the method and demonstrates the application to different problems.

Based on a side condition, Bluman and Cole [1969] introduced the non-classical method. They discussed the diffusion equation as an example. The result of their considerations was that completely different solutions for the heat equation occur. A group-theoretical explanation of this result was given by Olver [1986] and Levi and Winternitz [1989]. The non-classical method was extended to *weak symmetries* and *side conditions* by Olver and Rosenau [1986]. The difference between weak symmetries and symmetries is defined as follows.

Definition: Symmetry of a PDE

A symmetry group G of $\Delta = 0$ is a local transformation with the following properties:

1. The elements of G transform solutions of $\Delta = 0$ into new solutions of this equation.
2. The solutions of $\Delta = 0$ invariant under G follow from a system of PDEs containing a reduced set of independent variables. ○

The above definition rephrases the invariance conditions of Chapter 5 in different words. A weak symmetry, on the other hand, is a symmetry which satisfies the second condition of the definition but does not satisfy the first one. Thus, a weak symmetry transformation does not allow that new solutions of $\Delta = 0$ follow from known solutions. The second condition expand the class of solutions for a PDE. In the following, we will show that this freedom of a symmetry definition is capable to deliver a new kind of solution.

6.2. Mathematical Background of the Non-classical Method

In this section, we are looking for solutions of the general equation

$$\Delta(x, u_{(k)}) = 0 \tag{6.1}$$

under a side condition. Contrary to the classical method of Lie, we are seeking solutions under the condition that the invariant surface condition vanishes. As we know, the solution for the function u invariant under a point transformation is based on the characteristic equations

$$\frac{dx_1}{\xi_1} = \frac{dx_2}{\xi_2} = \cdots = \frac{dx_n}{\xi_n} = \frac{du^1}{\phi_1} = \frac{du^2}{\phi_2} = \cdots = \frac{du^m}{\phi_m}. \tag{6.2}$$

Equivalently, relation (6.2) can be formulated in connection with the vector field $\vec{v}$ by using the infinitesimals as

$$\sum_{i=1}^{n} \xi_i u_i^{\alpha} - \phi_{\alpha} = 0, \quad \alpha = 1, \ldots, m. \tag{6.3}$$

This equation (6.3) is abbreviated with $Q_{\alpha}(x, u_{(1)}) = 0$ by Olver [1986]. A solution of $\Delta = 0$ invariant under the transformation given by $\vec{v}$ satisfies the system of PDEs

$$\Delta(x, u_{(k)}) = 0, \tag{6.4}$$

$$Q_\alpha\,(x,\ u_{(1)}) = 0. \tag{6.5}$$

So far, we recalled the properties of the classical method by Lie. The non-classical method of Bluman and Cole now considers condition (6.5) as an additional side condition for the solution of the original equation $\Delta = 0$.

In a non-classical symmetry analysis of $\Delta = 0$, we are not only interested in the symmetries of the PDE but also in the symmetries of the PDE extended by the characteristic equation (6.5). The solutions of the non-classical method will generate a weak symmetry in the sense discussed in Section 6.1. At the other hand, it is certain that the vector fields of the non-classical method do not need to form a Lie algebra. Hence, there can be a wider class of similarity solutions than in the classical case.

An essential observation by Bluman and Cole [1969] is that an invariant solution $u(x)$ of $\Delta = 0$ does not only solve the PDE itself but also the invariant surface condition or characteristic equation:

$$Q_\alpha = 0, \qquad \alpha = 1, \ldots, q. \tag{6.6}$$

Thus, the invariance condition of the classical method is extended by this additional condition:

$$\mathrm{pr}^{(k)}\,\vec{v}\,\Delta\,|_{\substack{\Delta=0\\ Q_\alpha=0}} = 0, \tag{6.7}$$

$$\mathrm{pr}^{(1)}\,\vec{v}\,Q_\alpha\,|_{\substack{\Delta=0\\ Q_\alpha=0}} = 0. \tag{6.8}$$

Equation (6.8) does not establish any additional restriction in the derivation of the determining equation since it is satisfied identically. The main difference compared with Lie's method is that not only one side condition occurs but also the characteristics must vanish. This second side condition has the consequence that the derivatives of u are strongly coupled. The tight coupling of the derivatives results into a non-linear system of determining equations for the infinitesimals. This fact is the main difference between the classical and non-classical method. However, the non-linearity in the determining equations is a real problem in connection with symbolic calculations.

The above theoretical considerations can be cast into a more suitable formulation for computer calculation. The following discussion shows how the prolongation formulation can be rewritten in terms of the Fréchet formalism. We know from above that the non-classical method applies the classical Lie method to the extended system of equations

$$\Delta^i(x, u_{(k)}) = 0, \tag{6.9}$$

$$Q_\alpha = 0, \tag{6.10}$$

$i = 1, 2, \ldots, m$ and $\alpha = 1, 2, \ldots, m$. The invariance condition expressed by the Fréchet derivative reads

$$\left(\mathfrak{D}_\Delta(Q) + \sum_{i=1}^{p} \xi_i \, D_i \, \Delta\right)\Big|_{\substack{Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i \, u_i^\alpha = 0 \\ \Delta = 0}} = 0. \tag{6.11}$$

Equation (6.11) is quite similar to the invariance condition for point symmetries. The difference is that in the present case the characteristics have to vanish. The added surface condition can be annotated as a side condition or as a conditional equation that introduces new dependencies for the derivatives of u. These relations have to be taken into consideration during the calculations. The side condition $Q_\alpha = 0$ introduces a relation representing a strong connection between the derivatives of the dependent variables. This relationship has to be considered in the elimination of derivatives in the prolongation formula. Equation (6.11) contains all the necessary steps to calculate the determining equations in a nutshell. Let us summarize the algorithm in the following steps:

1. Calculate the prolongation of the equation $\Delta = 0$ as discussed for classical point symmetries.
2. Apply the side conditions to the prolongation formula.
3. Extract the determining equations as discussed for the classical method.

These three steps are the essentials of the non-classical method. Contrary to the classical method, we have as side conditions not only the equations $\Delta = 0$ but also the condition of the vanishing characteristics and the differential consequences of this relation. The main difficulty of this algorithm is contained in the second step. Since the characteristic equation is satisfied identically, we can apply this side condition either to the original equation itself or to the prolongation. Both procedures are discussed in the literature. The method by Levi and Winternitz [1989] prefers the elimination of the side conditions after the calculation of the prolongation. Clarkson and Mansfield [1994] eliminate the terms of the characteristics before the prolongation is calculated. The algorithm implemented in *MathLie* follow the considerations of Clarkson and Mansfield.

The kind of problems occurring during a non-classical calculation are demonstrated by the following example. Let us consider the characteristic equation in 2+1 dimensions. The characteristic equation for this case reads

$$u_x \, \xi_1(x, t, u) + u_t \, \xi_2(x, t, u) - \phi(x, t, u) = 0. \tag{6.12}$$

Let us assume that we are looking for a substitution replacing the u_t term. The resulting expression from (6.12) is given by

$$u_t \rightarrow \phi - u_x \xi_1 . \tag{6.13}$$

We replaced the ratios of ϕ/ξ_2 and ξ_1/ξ_2 simply by ϕ and ξ_1, respectively. This substitution corresponds to a formal coordinate transformation assuming that $\xi_2 = 1$. The derived substitution rules from (6.13) are up to second order given by

$$u_t \rightarrow \phi - u_x \xi_1, \tag{6.14}$$

$$u_{xt} \rightarrow \phi_x + u_x(\phi_u - \xi_{1\,x}) - u_x^2 \xi_{1\,u} - u_{xx} \xi_1, \tag{6.15}$$

$$u_{tt} \rightarrow \phi_t + u_t \phi_u - u_x(\xi_{1\,t} + u_t \xi_{1\,u}) - u_{xt} \xi_1 . \tag{6.16}$$

The last of these relations (6.16) contains terms of u_t and $u_{x,t}$ which have to be replaced by the first two rules (6.14) and (6.15), respectively. Thus, relation (6.16) is only expressible if the preceding substitutions were calculated first. If not, relation (6.16) contains redundant information which affects the invariant condition (6.11). Thus, one has to keep the following suggestions in mind if one solves the side conditions $Q_\alpha = 0$.

1. The derivatives for which the side condition $Q_\alpha = 0$ is solved should occur in the original equation in a simple form. This guarantees that the original equation stays simple after the substitution of the side conditions and thus saves a lot of computing time.

2. If we have more than one dependent variable, we should solve the characteristic equation with respect to a derivative occurring in all side conditions. This allows the introduction of the relation $\xi_i = 1$ for a specific independent variable x_i.

The substitutions derived in the above example indicate that the determining equations are non-linear functions in the infinitesimals. This change from a linear system of determining equations in the case of the classical method to a system of non-linear determining equations for the non-classical method bears very complicated problems. The problem of solving the non-linear determining equations is currently not disclosed. Thus, we cannot automatically find solutions for the infinitesimals in all cases with *MathLie*. At the moment, we use an interactive and a pseudo-automatic method to derive the solutions from the determining equations. The following examples will demonstrate how we can find non-classical symmetries by applying the functions of *MathLie*.

6.3. Applications of the Non-classical Method

The non-classical method has been applied to various PDEs. New classes of solutions which cannot be obtained by the classical method have been found for the heat equation by Bluman and Cole [1969], the Boussinesq equation by Levi and Winternitz [1989], the Burgers equation by Pucci [1992], and the Fitzhugh Nagumo equation by Nucci and Clarkson [1992]. We will demonstrate in this section how the functions of *MathLie* can be exerted to derive the determining equations. In addition, we will solve the determining equations interactively and automatically.

6.3.1 The Heat Equation

Let us start with the well-studied heat equation by Bluman and Cole [1969]. The scaled heat equation in (1+1) dimensions reads

$$u_t - u_{x,x} = 0, \tag{6.17}$$

where u is the field describing the variation of a scaled temperature in a pipe, for example. The heat equation is one of the rare examples which allows the study of different solution procedures. In Section 5.6.1, we examined the heat equation with the classical method of Lie. The result was a six-dimensional discrete symmetry group in connection with an infinite dimensional group. Here, we will examine the heat equation again by using the non-classical method. The determining equations are derived with the help of Lie[] in connection with two options of the function. The equation of motion in *Mathematica* is created by

```
U = u[x, t];

diffus = {∂t U - ∂x,x U}; diffus // LTF

ut - ux,x == 0
```

We first extend our database by a file containing the information on the equation. The left-hand side of the heat equation (6.17) is stored to the file *ncdiffu.dgl* by applying LieEquations[]:

```
LieEquations["ncdiffu.dgl", diffus, {u}, {x, t}, {},
{"Heat equation"}, {{"G. Baumann"}, {"Ulm 1997"}}]
```

The file *ncdiffu.dgl* contains all the necessary information to start the analysis.

The information in the file is contained in global variables. The variable *Title*, for example, contains a headline describing the purpose of the equation. The variable

Source carries, in sublists, information on the origin of the equation. The remaining variables such as *IndepVar*, *DependVar*, *EqList*, *SubsList*, *ParameterS*, *ListXi*, and *ListPhi* contain information on the independent and dependent variables, the equation of motion, the terms for which the side condition in the classical method is solved, a list of parameters, and two lists for the infinitesimals, respectively. All information contained in the file is necessary for the function Lie[] to carry out a symmetry analysis. The non-classical symmetry method is initiated by using Lie[] in connection with the option *NonclassicalSymmetries→True*. The option *NonclassicalCases→{t}* of Lie[] selects those terms of the characteristic equations which contain a derivative with respect to the specified variable, here t, meaning that all higher derivatives containing a partial derivative with respect to t are used to match and eliminate the terms in the original equation. We start the non-classical symmetry analysis for the heat equation by

```
Lie["ncdiffu.dgl", NonclassicalSymmetries → True,
        NonclassicalCases → {t}] // LTF
```

$$(\xi_2)_u == 0$$

$$(\xi_1)_{u,u} == 0$$

$$(\xi_2)_{u,u} == 0$$

$$(\xi_2)_x == 0$$

$$(\xi_2)_{x,u} == 0$$

$$\phi_1 (\xi_2)_u + \xi_1 (\xi_2)_x + \xi_2 (\xi_2)_{x,x} == 0$$

$$-\xi_2^2 (\xi_1)_t + 2 \xi_2 \phi_1 (\xi_1)_u - 2 \xi_1 \xi_2 (\xi_1)_x + \xi_1 \xi_2 (\xi_2)_t + \xi_1 \phi_1 (\xi_2)_u + \xi_1^2 (\xi_2)_x + \xi_2^2 (\xi_1)_{x,x} - 2 \xi_2^2 (\phi_1)_{x,u} == 0$$

$$-2 \xi_2 \phi_1 (\xi_1)_x + \xi_2 \phi_1 (\xi_2)_t + \phi_1^2 (\xi_2)_u + \xi_1 \phi_1 (\xi_2)_x - \xi_2^2 (\phi_1)_t + \xi_2^2 (\phi_1)_{x,x} == 0$$

$$2 \xi_1 (\xi_1)_u - 2 \xi_2 (\xi_1)_{x,u} + \xi_2 (\phi_1)_{u,u} == 0$$

The result is a non-linear coupled system of nine partial differential equations determining the non-classical symmetries of the heat equation. Our task is to solve the determining equations. If we have information on the structure of the infinitesimals, we can incorporate this information into the database file *ncdiffu.dgl*. The extension of the file will simplify the determining equations in the solution procedure. Let us first store the following relations into *ncdiffu.dgl*:

```
ListXi = {xi[1][x, t, u[x, t]], 1};
ListPhi = {phi[1][x, t, u[x, t]]};
Save["ncdiffu.dgl", ListXi, ListPhi];
```

Save[] appends the two relations on the infinitesimals to the file *ncdiffu.dgl*. The gathered information can now be used to simplify the determining equations by starting the analysis a second time. Since we provided the database file with new information, we can activate this information by setting the option *NonclassicalInfo→True*

```
Lie["ncdiffu.dgl", NonclassicalSymmetries → True,
  NonclassicalCases → {t}, NonclassicalInfo → True] //
 LTF
```

$$(\xi_1)_{u,u} == 0$$
$$2\,\xi_1\,(\xi_1)_u - 2\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$
$$-2\,\phi_1\,(\xi_1)_x - (\phi_1)_t + (\phi_1)_{x,x} == 0$$
$$-(\xi_1)_t + 2\,\phi_1\,(\xi_1)_u - 2\,\xi_1\,(\xi_1)_x + (\xi_1)_{x,x} - 2\,(\phi_1)_{x,u} == 0$$

We end up with four non-linear determining equations. We realize that the number of equations is reduced by five. Although the equations are still non-linear, we have a good chance to solve them. First, we observe that not all equations are coupled. Thus, the strategy is to extract those equations which are simple and linear. Solving this subset of equations will provide us with more information on the infinitesimals. This information is essential to solve the remaining non-linear determining equations. The first equation of the determining equations states that *xi*[1] allows a solution which is independent of the dependent variable u. Thus, we can use this information to extend our file *ncdiffu.dgl* with

```
Clear[A]
ListXi = {A[x, t], 1};
ListPhi = {phi[1][x, t, u[x, t]]};
Save["ncdiffu.dgl", ListXi, ListPhi];
```

Running the function Lie[] again and using the new information in the file, we get

```
Lie["ncdiffu.dgl", NonclassicalSymmetries → True,
  NonclassicalCases → {t}, NonclassicalInfo → True] //
 LTF
```

$$(\phi_1)_{u,u} == 0$$
$$-A_t - 2\,A\,A_x + A_{x,x} - 2\,(\phi_1)_{x,u} == 0$$
$$-2\,A_x\,\phi_1 - (\phi_1)_t + (\phi_1)_{x,x} == 0$$

The resulting three equations are again simplified and determine the infinitesimals. A glance at these equations shows that we get a mixed system of linear and non-linear determining equations. The linear equation $\phi_{u,u} = 0$ serves to gain additional information on the structure of the infinitesimals. This equation allows us to represent ϕ as a linear function in u. Thus, we set the infinitesimals to

```
ListXi = {A[x, t], 1};
```

as before, and ϕ as

```
ListPhi = {B[x, t] + C[x, t] u[x, t]};
```

where *B[x,t]* and *C[x,t]* are arbitrary functions. Saving this result,

```
Save["ncdiffu.dgl", ListXi, ListPhi];
```

and starting the calculation again by

```
ncdiffu = Lie["ncdiffu.dgl", NonclassicalSymmetries -> True,
        NonclassicalCases -> {t},
        NonclassicalInfo -> True]; ncdiffu // LTF
```

$$-A_t - 2\,A\,A_x - 2\,C_x + A_{x,x} == 0$$
$$-2\,B\,A_x - B_t + B_{x,x} == 0$$
$$-2\,C[x,\,t]\,A_x - C_t + C_{x,x} == 0$$

reveals a system of three coupled non-linear equations for the unknown functions *A*, *B*, and *C*. This system of equations determines the infinitesimals of the non-classical group. At this stage, we clearly face the problem that starting with a linear PDE, we end up with a non-linear one. This system of non-linear PDEs is difficult to solve. However, in our analysis, we only need special solutions of this equation to find new solutions of the heat equation. We already know a method to analyze such non-linear equations. Applying Lie's point symmetry procedure in the usual way, we get the symmetries of these equations. The symmetries of the three coupled PDEs read

```
incdiffu = Infinitesimals[ncdiffu, {A, B, C}, {x, t}];
incdiffu // LTF
```

```
PDESolve::nsf : Use option Standard->True.
 This may lead to further solutions in case of
 linear Systems
```

$$-(\mathcal{F}_1)_t + (\mathcal{F}_1)_{x,x} == 0$$
$$\xi_1 == k1 - 2\,k5\,t + \frac{k3\,x}{2} - \frac{2\,k7\,t\,x}{5}$$
$$\xi_2 == k2 + k3\,t - \frac{2\,k7\,t^2}{5}$$
$$\phi_1 == \frac{1}{10}\,(-5\,A\,k3 - 20\,k5 + 4\,A\,k7\,t - 4\,k7\,x)$$
$$\phi_2 == B\left(k4 + k6 + k7\,t + k5\,x + \frac{k7\,x^2}{10}\right) + C\,\mathcal{F}_1 - A\,(\mathcal{F}_1)_x - (\mathcal{F}_1)_{x,x}$$
$$\phi_3 == \frac{1}{5}\,(-5\,C\,k3 + 5\,A\,k5 + k7 + 4\,C\,k7\,t + A\,k7\,x)$$

The result is that the three coupled PDEs for the non-classical symmetries posses a seven-dimensional discrete symmetry group. In addition, there exist an infinite dimensional group given by the function *free*[1]. This arbitrary function has to satisfy the heat equation. The symmetries of the non-classical determining equations exhibit a similar structure as the point symmetries of the heat equation. The symmetries can now be used to find special solutions for the functions A, B, and C. First, we select the subgroup with $k1 = 1$:

```
infi = {{xi[1][x, t, A, B, C],
     xi[2][x, t, A, B, C]}, {phi[1][x, t, A, B, C],
phi[2][x, t, A, B, C], phi[3][x, t, A, B, C]}} /. incdiffu[[1]] /.
{k1 -> 0, k2 -> 1, k3 -> 0, k4 -> 0, k5 -> 0, k6 -> 0, k7 -> 0,
   free[_] → Function[{x, t}, 0]}
```

```
{{0, 1}, {0, 0, 0}}
```

The corresponding reduction of the non-classical determining equations follow with this representation of the infinitesimals by

```
red =
 LieReduction[ncdiffu, {A, B, C}, {x, t}, infi[[1]], infi[[2]]];
red // Flatten // LTF /. zeta1 → ζ1
```

$$x - zeta1 == 0$$
$$A - F_1 == 0$$
$$B - F_2 == 0$$
$$C - F_3 == 0$$
$$-2\,F3_{zeta1} - 2\,F_1\,(F_1)_{zeta1} + (F_1)_{zeta1,zeta1} == 0$$
$$-2\,F_2\,F1_{zeta1} + (F_2)_{zeta1,zeta1} == 0$$
$$-2\,F_3\,F1_{zeta1} + (F_3)_{zeta1,zeta1} == 0$$

The similarity representation of the non-linear determining equations is given by three coupled non-linear ODEs. To simplify things, let us assume that the functions F2 and F3 are given by the trivial solutions

```
redh =
 red[[3]] /. {F3 → Function[zeta1, 0], F2 → Function[zeta1, 0]}
```

```
{-2 F1[zeta1] F1'[zeta1] + F1''[zeta1] == 0, True, True}
```

The substitution of $F2 = 0$ and $F3 = 0$ into the reduced equations simplifies the system to a single equation. This equation is solved by DSolve[]

```
sol1 = DSolve[redh[[1]], F1, zeta1]
```

```
{{F1 → (√C[2] Tan[#1 √C[2] + C[1] √C[2] ]&)}}
```

The solution simplifies further if we specify the constants of integration in a certain way:

```
sol2 = ( F1[zeta1] /. sol1 ) /. {C[1] → 0, C[2] → α^2} // Simplify
```

$$\left\{\sqrt{\alpha^2}\ \text{Tan}\left[\text{zeta1}\ \sqrt{\alpha^2}\ \right]\right\}$$

The final solution for the unknown functions A, B, and C of the non-classical determining equations is thus given by

```
rsol = Solve[red[[2]] /. sol1 /.
      {F3 → Function[zeta1, 0], F2 → Function[zeta1, 0]} /.
     {C[1] → 0, C[2] → α^2} // Simplify,
  {A, B, C}]
```

$$\left\{\left\{\text{A} \to \sqrt{\alpha^2}\ \text{Tan}\left[\text{x}\ \sqrt{\alpha^2}\ \right],\ \text{B} \to 0,\ \text{C} \to 0\right\}\right\}$$

We can check that this special solution satisfies the determining equations by rewriting the above results in a pure function form:

```
infired = {A → Function[{x, t}, √α^2 Tan[x √α^2 ]],
  B → Function[{x, t}, 0], C → Function[{x, t}, 0]};
infired // LTF
```

$$\text{A} == \sqrt{\alpha^2}\ \text{Tan}\left[\text{x}\ \sqrt{\alpha^2}\ \right]$$
$$\text{B} == 0$$
$$\text{C} == 0$$

Inserting this representation of the solution into the non-linear determining equations, we find that the equations are identically satisfied:

```
ncdiffu /. infired // Simplify
```

$$\{0,\ 0,\ 0\}$$

The non-classical infinitesimals for the heat equation are thus given by

```
infdifus = {ListXi, ListPhi} /. infired
```

$$\left\{\left\{\sqrt{\alpha^2}\ \text{Tan}\left[\text{x}\ \sqrt{\alpha^2}\ \right],\ 1\right\},\ \{0\}\right\}$$

representing a special transformation under which the heat equation is invariant. The reduction of the heat equation with these infinitesimals follows:

```
rdiffus =
 LieReduction[diffus, {u}, {x, t}, infdifus[[1]], infdifus[[2]]];
rdiffus // Flatten // LTF /. zeta1 → ζ1
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$-\text{zeta1} - \frac{-t\,\alpha^2 + \text{Log}[\text{Sin}[x\,\alpha]]}{\alpha^2} == 0$$

$$u - F_1 == 0$$

$$\left(E^{2\,t\,\alpha^2} - E^{2\,\text{zeta1}\,\alpha^2}\right)\left(\alpha^2\,F1_{\text{zeta1}} + F1_{\text{zeta1,zeta1}}\right) == 0$$

The similarity representation of the heat equation is an expression combining trigonometric and logarithmic functions. The unknown function F1 of this reduction is determined by a second-order ODE. This ODE is solved by DSolve[]:

```
srdiffus = DSolve[(α^2 F1'[zeta1] + F1''[zeta1]) == 0, F1, zeta1]
```

$$\left\{\left\{F1 \to \left(-\frac{E^{-\alpha^2\,\#1}\,C[1]}{\alpha^2} + C[2]\,\&\right)\right\}\right\}$$

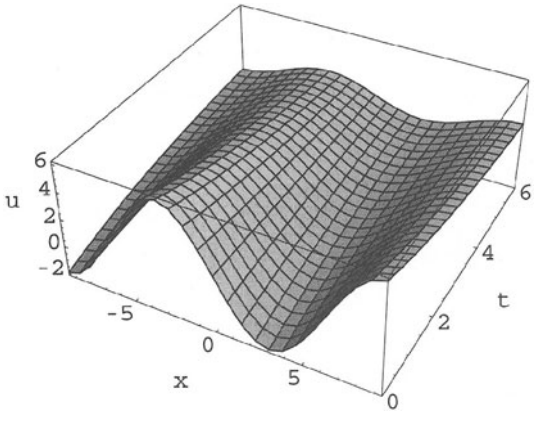
The result is an exponential solution containing two constants of integration $C[1]$ and $C[2]$. In original variables, the solution reads

```
sol = Solve[(rdiffus[[2]] /. srdiffus) // Simplify, u] // Flatten
```

$$\left\{u \to C[2] - \frac{E^{-t\,\alpha^2}\,C[1]\,\text{Sin}[x\,\alpha]}{\alpha^2}\right\}$$

If we specify the parameters α, $C[1]$, and $C[2]$ in an appropriate way, we can graphically represent the solution by

```
Plot3D[Evaluate[u /. sol /. {α → 1 / 2, C[1] → 1, C[2] → 2}],
 {x, -3 π, 3 π}, {t, 0, 6}, AxesLabel → {"x", "t", "u"},
 PlotPoints → 25]
```



The example of the heat equation demonstrates how the basic functions of *MathLie* can be used to find non-classical solutions. The above discussion is based on the combination of batch mode calculations and interactive calculations in *MathLie*. The following calculation will show how non-classical symmetries can be derived by pseudo-automatic calculations in *MathLie*.

6.3.2 The Boussinesq Equation

The Boussinesq [1872] equation

$$u_{t,t} + u\,u_{x,x} + u_x^2 + u_{x,x,x,x} = 0 \tag{6.18}$$

is one of the equations frequently examined in connection with solution procedures for non-linear models. The equation arises in several physical applications ranging from surface waves of rectangular canals to applications in plasma physics. The point symmetries of the Boussinesq equation were examined by Nishitani and Tajiri [1982] and Rosenau and Schwarzmeier [1986]. The discussion of the non-classical symmetries by Levi and Winternitz [1989] revealed that the symmetries of the direct method by Clarkson and Kruskal [1989], Clarkson [1989] follow from a non-classical analysis. We use this equation here to demonstrate the pseudo-automatic calculation of non-classical symmetries.

The Boussinesq equation in *Mathematica* notation reads:

```
boussinesq = ∂t u[x, t] +
   u[x, t] ∂x,x u[x, t] + (∂x u[x, t])^2 + ∂{x,4} u[x, t] == 0;
boussinesq // LTF
```

$$u_t + u_x^2 + u\,u_{x,x} + u_{x,x,x,x} == 0$$

The point symmetries of the Boussinesq equation are

```
liePointSymmetries = Infinitesimals[boussinesq, u, {x, t}];
liePointSymmetries // LTF
```

$$\phi_1 == -2\,k3\,u$$
$$\xi_1 == k2 + k3\,x$$
$$\xi_2 == k1 + 4\,k3\,t$$

This finite dimensional group of order three contains translations and scaling transformations as symmetries. The non-classical symmetries follow by applying the function NonClassicalPointSymmetries[]. This function assumes by default that the option *NonclassicalCases* is set to the last variable of the independent variables. Typically, this variable is the time t. If you prefer to use another variable, for

example, x, you have to specify this change in the option *NonclassicalCases*→{x}. The nonlinear determining equations follow from

```
ncdeter =
 NonClassicalPointSymmetries[boussinesq, {u}, {x, t}];
ncdeter // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_2)_u == 0$$
$$(\xi_1)_{u,u} == 0$$
$$(\xi_2)_{u,u} == 0$$
$$(\xi_1)_{u,u,u} == 0$$
$$(\xi_2)_{u,u,u} == 0$$
$$(\xi_1)_{u,u,u,u} == 0$$
$$(\xi_2)_{u,u,u,u} == 0$$
$$(\xi_2)_x == 0$$
$$(\xi_2)_{x,u} == 0$$
$$(\xi_2)_{x,u,u} == 0$$
$$(\xi_2)_{x,u,u,u} == 0$$
$$(\xi_2)_{x,x} == 0$$
$$(\xi_2)_{x,x,u} == 0$$
$$(\xi_2)_{x,x,u,u} == 0$$
$$(\xi_2)_{x,x,x} == 0$$
$$(\xi_2)_{x,x,x,u} == 0$$
$$\phi_1 + 2\,u\,(\xi_1)_x - 4\,(\xi_1)_{x,x,x} + 6\,(\phi_1)_{x,x,u} == 0$$
$$-3\,(\xi_1)_{x,x,u,u} + 2\,(\phi_1)_{x,u,u,u} == 0$$
$$-3\,(\xi_1)_{x,x,u} + 2\,(\phi_1)_{x,u,u} == 0$$
$$-3\,(\xi_1)_{x,x} + 2\,(\phi_1)_{x,u} == 0$$
$$-4\,(\xi_1)_{x,u,u,u} + (\phi_1)_{u,u,u,u} == 0$$
$$-4\,(\xi_1)_{x,u,u} + (\phi_1)_{u,u,u} == 0$$
$$-4\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$
$$-4\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$
$$\phi_1\,(\xi_2)_u + \xi_1\,(\xi_2)_x - \xi_2\,(\xi_2)_{x,x,x} == 0$$
$$-\xi_2^2\,(\xi_1)_t + 4\,\xi_2\,\phi_1\,(\xi_1)_u - 4\,\xi_1\,\xi_2\,(\xi_1)_x + \xi_1\,\xi_2\,(\xi_2)_t + \xi_1\,\phi_1\,(\xi_2)_u + \xi_1^2\,(\xi_2)_x + 2\,\xi_2^2\,(\phi_1)_x - u\,\xi_2^2\,(\xi_1)_{x,x} + 2\,u\,\xi_2^2\,(\phi_1)_{x,u} - \xi_2^2\,(\xi_1)_{x,x,x} + 4\,\xi_2^2\,(\phi_1)_{x,x,u} == 0$$
$$-4\,\xi_2\,\phi_1\,(\xi_1)_x + \xi_2\,\phi_1\,(\xi_2)_t + \phi_1^2\,(\xi_2)_u + \xi_1\,\phi_1\,(\xi_2)_x - \xi_2^2\,(\phi_1)_t - u\,\xi_2^2\,(\phi_1)_{x,x} - \xi_2^2\,(\phi_1)_{x,x,x} == 0$$
$$-4\,\xi_1\,(\xi_1)_u + 2\,\xi_2\,(\xi_1)_x + \xi_2\,(\phi_1)_u - 2\,u\,\xi_2\,(\xi_1)_{x,u} + u\,\xi_2\,(\phi_1)_{u,u} - 4\,\xi_2\,(\xi_1)_{x,x,u} + 6\,\xi_2\,(\phi_1)_{x,u,u} == 0$$

The result is a system of 29 coupled non-linear determining equations for the infinitesimals ξ_1, ξ_2, and ϕ_1. Only the last four equations are non-linear. The first 25 equations are linear. We can use these linear equations to find a partial solution of the determining equations. Before we apply a function to the total set of equations, let us rewrite the infinitesimals in such a way that $\xi_2 = 1$. Defining a general substitution for *xi*[2], we find

```
ncdeterh = DeleteCases[((ncdeter /. u[x, t] → u)  /.
     xi[2] → Function[{x, t, u}, 1]) /. u → u[x, t], 0];
ncdeterh // LTF
```

$$(\xi_1)_u == 0$$
$$(\xi_1)_{u,u} == 0$$
$$(\xi_1)_{u,u,u} == 0$$
$$(\xi_1)_{u,u,u,u} == 0$$
$$\phi_1 + 2\,u\,(\xi_1)_x - 4\,(\xi_1)_{x,x,x} + 6\,(\phi_1)_{x,x,u} == 0$$
$$-3\,(\xi_1)_{x,x,u,u} + 2\,(\phi_1)_{x,u,u,u} == 0$$
$$-3\,(\xi_1)_{x,x,u} + 2\,(\phi_1)_{x,u,u} == 0$$
$$-3\,(\xi_1)_{x,x} + 2\,(\phi_1)_{x,u} == 0$$
$$-4\,(\xi_1)_{x,u,u,u} + (\phi_1)_{u,u,u,u} == 0$$
$$-4\,(\xi_1)_{x,u,u} + (\phi_1)_{u,u,u} == 0$$
$$-4\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$
$$-4\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$
$$-(\xi_1)_t + 4\,\phi_1\,(\xi_1)_u - 4\,\xi_1\,(\xi_1)_x + 2\,(\phi_1)_x - u\,(\xi_1)_{x,x} + 2\,u\,(\phi_1)_{x,u} - (\xi_1)_{x,x,x,x} + 4\,(\phi_1)_{x,x,x,u} == 0$$
$$-4\,\phi_1\,(\xi_1)_x - (\phi_1)_t - u\,(\phi_1)_{x,x} - (\phi_1)_{x,x,x,x} == 0$$
$$-4\,\xi_1\,(\xi_1)_u + 2\,(\xi_1)_x + (\phi_1)_u - 2\,u\,(\xi_1)_{x,u} + u\,(\phi_1)_{u,u} - 4\,(\xi_1)_{x,x,x,u} + 6\,(\phi_1)_{x,x,u,u} == 0$$

Applying now the general-purpose solver PDESolve[] of *MathLie* to the system of coupled non-linear equations, we can automatically derive solutions. The function PDESolve[] responds with a question concerning the solution branches of the non-linear determining equations. This interruption of the calculation is terminated by providing the number of the condition printed by the function:

```
partsol = PDESolve[ncdeter, {u}, {x, t}]; partsol // LTF

There exists no unique solution of the equations.

Please choose one of the following results
```

```
1, {free[3][t] → 0}

2, {free[1]^(2,0)[x, t] → 0}

There exists no unique solution of the equations.

Please choose one of the following results

1, {free[3][t] → 0}

2, {free[1]^(3,0)[x, t] → 0}
```

$$-2\,\mathcal{F}_1\,(\mathcal{F}_2)_t^2 - \mathcal{F}_2^2\,(\mathcal{F}_1)_{t,t} + \mathcal{F}_2\,(2\,(\mathcal{F}_1)_t\,(\mathcal{F}_2)_t + \mathcal{F}_1\,(\mathcal{F}_2)_{t,t}) == 0$$

$$\xi_1 == \mathcal{F}_2 + \frac{1}{4}\,x\,(\mathcal{F}_1)_t - \frac{x\,\mathcal{F}_1\,(\mathcal{F}_2)_t}{4\,\mathcal{F}_2}$$

$$\xi_2 == \mathcal{F}_1$$

$$\phi_1 == \frac{1}{2}\,u\left(-(\mathcal{F}_1)_t + \frac{\mathcal{F}_1\,(\mathcal{F}_2)_t}{\mathcal{F}_2}\right)$$

We select the second case for each inquiry of PDESolve[]. The result for these choices is a representation of the non-classical infinitesimals depending on two arbitrary functions $\mathcal{F}_i = free[i]$, $i = 1, 2$. We know that for ξ_2 the relation $\xi_2 = 1$ was assumed in the above calculation. Thus, we can set $free[1] = 1$.

```
ppsol = partsol /. free[1] → Function[t, 1];
ppsol // Flatten // LTF
```

$$\phi_1 == \frac{u\,(\mathcal{F}_2)_t}{2\,\mathcal{F}_2}$$

$$\xi_1 == \mathcal{F}_2 - \frac{x\,(\mathcal{F}_2)_t}{4\,\mathcal{F}_2}$$

$$\xi_2 == 1$$

$$-2\,(\mathcal{F}_2)_t^2 + \mathcal{F}_2\,(\mathcal{F}_2)_{t,t} == 0$$

The infinitesimals reduce to a simpler form containing only one undetermined function $\mathcal{F}_2 = free[2]$. This function has to satisfy the last determining equation, which is a second-order non-linear ODE. The partial results so far derived can be used in another calculation with PDESolve[]:

```
isol = PDESolve[ppsol, {u}, {x, t}]; isol // LTF
```

$$\xi_1 == \frac{4 - k1\,x}{4\,k1\,k2 - 4\,k1\,t}$$

$$\xi_2 == 1$$

$$\phi_1 == \frac{u}{2\,k2 - 2\,t}$$

The final result of the calculation is a system of infinitesimals containing two parameters *k1* and *k2*. These two parameters determine the transformation properties

of the non-classical infinitesimal transformation. The above representation of the infinitesimals is represented in a more convenient way by the following transformations:

```
ncinfi = isol /. u[x, t] → u /. Rule[a_[n_][h___], b_] →
    Rule[a[n], Function[$v, $w] /. {$v → {h}, $w → b}] // Flatten;
ncinfi // LTF
```

$$\phi_1 == \frac{u}{2\,k2 - 2\,t}$$

$$\xi_1 == \frac{4 - k1\,x}{4\,k1\,k2 - 4\,k1\,t}$$

$$\xi_2 == 1$$

After the rearrangements of the representation, we know the infinitesimals in a standard form which is helpful in the following calculations. Our aim is to derive solutions for the Boussinesq equation. To uncover the reduction of the Boussinesq equation, we first select a sub-class of the infinitesimal transformations with $k1 = 1$ and $k2 = c$ with c a real constant:

```
infi = {{xi[1][x, t, u], xi[2][x, t, u]},
    {phi[1][x, t, u]}} /. ncinfi /.
  {k1 → 1, k2 → c}
```

$$\left\{\left\{\frac{4 - x}{4\,c - 4\,t}, 1\right\}, \left\{\frac{u}{2\,c - 2\,t}\right\}\right\}$$

Inserting the equation and the infinitesimals into LieReduction[] provides us with a reduction of the Boussinesq equation:

```
red = LieReduction[boussinesq, {u}, {x, t}, infi[[1]], infi[[2]];
(red // Flatten // LTF) /. zeta1 → ζ1
```

$$-\frac{c + t}{(-4 + x)^4} - \zeta_1 == 0$$

$$u\,(-4 + x)^2 - F_1 == 0$$

$$(1 + 2904\,\zeta_1)\,(F_1)_{\zeta_1} + 2\,(5\,F_1^2 + 2\,F_1\,(30 + 13\,\zeta_1\,(F_1)_{\zeta_1} + 4\,\zeta_1^2\,(F_1)_{\zeta_1,\zeta_1}) + 8\,\zeta_1^2\,((F_1)_{\zeta_1}^2 + 351\,(F_1)_{\zeta_1,\zeta_1} + 8\,\zeta_1\,(19\,(F_1)_{\zeta_1,\zeta_1,\zeta_1} + 2\,\zeta_1\,(F_1)_{\zeta_1,\zeta_1,\zeta_1,\zeta_1}))) == 0$$

The resulting reduction consists of a similarity representation of the original variable u containing an unknown function F1. This function has to satisfy a fourth-order ODE, which is easier to read if we rename the variables

```
eqr = red[[3, 1]] /. {F1 → H, zeta1 → ζ}; eqr // LTF
```

$$120\,H + 10\,H^2 + H_\zeta + 2904\,\zeta\,H_\zeta + 52\,H\,\zeta\,H_\zeta + 16\,\zeta^2\,H_\zeta^2 + 5616\,\zeta^2\,H_{\zeta,\zeta} + 16\,H\,\zeta^2\,H_{\zeta,\zeta} + 2432\,\zeta^3\,H_{\zeta,\zeta,\zeta} + 256\,\zeta^4\,H_{\zeta,\zeta,\zeta,\zeta} == 0$$

Surprisingly we can find a first integral of this complicated equation by integrating the left-hand side with respect to ζ.

```
eqhh = ∫ (eqr[[1]]) dζ == K
```

$$(1 + 120\,\zeta)\,H[\zeta] + 10\,\zeta\,H[\zeta]^2 + 16\,\zeta^2\,(87 + H[\zeta])\,H'[\zeta] + 1408\,\zeta^3\,H''[\zeta] + 256\,\zeta^4\,H^{(3)}[\zeta] == K$$

where K is the constant of integration. Now, the question is: Can we solve this third-order equation by Lie's methods? We know from Section 4.4.3 that a solution follows by quadratures if the equation possesses at least n symmetries, where n is the largest order of derivatives. The first step in integrating the third-order ODE is the examination of the symmetries:

```
ihh = Infinitesimals[eqhh, H, ζ, {K}]; ihh // LTF
```

$$\phi_1 == 0$$
$$\xi_1 == 0$$

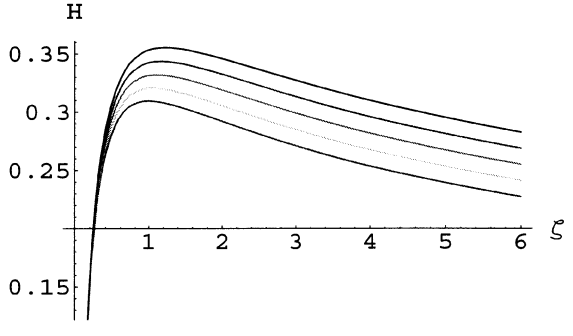
The result is that the third-order ODE resulting from the Boussinesq equation does not allow any symmetry. Thus, the integration procedure by Lie fails. However, we are able to represent a solution by a numerical integration. The validity of the solution will be restricted by initial conditions and the finite interval for the similarity variable ζ. An example for a numerical integration is given below.

```
nboussinesq = Map[NDSolve[{eqhh /. K → #, H[0.1] == 0,
      H'[0.1] == 2, H''[0.1] == 1}, H, {ζ, 0.1, 15}]&,
  {1, 2, 3, 4, 5}]
```

```
{{{H → InterpolatingFunction[{{0.1, 15.}}, <>]}},
 {{H → InterpolatingFunction[{{0.1, 15.}}, <>]}},
 {{H → InterpolatingFunction[{{0.1, 15.}}, <>]}},
 {{H → InterpolatingFunction[{{0.1, 15.}}, <>]}},
 {{H → InterpolatingFunction[{{0.1, 15.}}, <>]}}}
```

The calculation is carried out by changing the constant K in steps of 1 from 1 to 5. The initial conditions are fixed for each integration. The result of the numerical solution is used to graphically represent the different solutions:

```
Plot[Evaluate[H[ζ] /. nboussinesq],
 {ζ, .1, 6}, AxesLabel → {"ζ", "H"}, PlotStyle →
  {RGBColor[1, 0, 0], RGBColor[0.996109, 0.996109, 0],
   RGBColor[0.996109, 0.500008, 0], RGBColor[0, 0.500008, 0],
   RGBColor[0, 0, 1]}]
```



The numerical solution is connected with the similarity solution via the similarity transformation. We can now at least numerically represent the non-classical similarity solution for the Boussinesq equation. The example demonstrates that the functions of *MathLie* support a pseudo-automatic procedure for the non-classical method. The interactions are reduced to a minimal number of steps. We also demonstrated that *MathLie* is capable of solving non-linear determining equations with a decision support provided by the user. The functions NonClassicalPointSymmetries[] and PDESolve[] reduce the solution steps and the interaction to a convenient number.

6.3.3 The Fokker-Planck Equation

The Fokker-Planck (FP) equation is a general equation to describe statistical phenomena in condensed matter physics, quantum optics, chemical physics, and fluctuations in many other physical problems. For a detailed discussion of the Fokker-Planck equation, we refer to the book by Risken [1984]. In this section, we restrict our considerations to the (1+1)-dimensional version of the FP equation

$$u_t = (a(x)\,u)_x + (b(x)\,u)_{x,x}, \tag{6.19}$$

where $a(x)$ denotes the drift and $b(x)$ the diffusion coefficient. Both coefficients are functions of the spatial variable x. Subscripts in equation (6.19) denote partial derivatives with respect to time t and x. Let us assume that we have a system with constant diffusion $b(x) = 1$ and a linear drift term $a(x) = x$. Such a situation is

possible for a Brownian particle in a liquid (Cicogna and Vitali [1990]). Under these conditions, equation (6.19) reduces to

```
fokkerPlanck = ∂t u[x, t] - x ∂x u[x, t] - ∂x,x u[x, t] == 0;
fokkerPlanck // LTF
```

$$u_t - x\,u_x - u_{x,x} == 0$$

The point symmetries of this equation are

```
ifokkerPlanck = Infinitesimals[fokkerPlanck, u, {x, t}];
ifokkerPlanck // LTF
```

$$-(\mathcal{F}_1)_t + x\,(\mathcal{F}_1)_x + (\mathcal{F}_1)_{x,x} == 0$$

$$\xi_1 == E^{-t}\,k1 + E^{t}\,k2 + \frac{1}{2}\,E^{-2t}\,(-E^{4t}\,k3 + k5)\,x$$

$$\xi_2 == -\frac{1}{2}\,E^{2t}\,k3 - \frac{1}{2}\,E^{-2t}\,k5 + k6$$

$$\phi_1 == \frac{1}{2}\,u\,(E^{2t}\,k3 + 2\,k4 - 2\,E^{t}\,k2\,x + E^{2t}\,k3\,x^2) + \mathcal{F}_1$$

The result is a six-dimensional finite group extended by an infinite group given by *free*[*1*]. The arbitrary function *free*[*1*] satisfies the FP equation. The structure of the group has some resemblance to the group of the heat equation. However, the detailed structure is completely different.

The next step of our examination is the determination of the non-classical symmetries. To derive the non-classical determining equations for the FP equation, we apply the operator $n\mathcal{PS}^{\gamma}_{u,x}[\Delta]$ to the equation

```
ncfokkerPlanck = nPS{}{u},{x,t} [fokkerPlanck];
ncfokkerPlanck // LTF
```

$$(\xi_2)_u == 0$$

$$(\xi_1)_{u,u} == 0$$

$$(\xi_2)_{u,u} == 0$$

$$(\xi_2)_x == 0$$

$$(\xi_2)_{x,u} == 0$$

$$\phi_1\,(\xi_2)_u + \xi_1\,(\xi_2)_x + \xi_2\,(\xi_2)_{x,x} == 0$$

$$-\xi_1\,\xi_2^2 - \xi_2^2\,(\xi_1)_t + 2\,\xi_2\,\phi_1\,(\xi_1)_u - 2\,\xi_1\,\xi_2\,(\xi_1)_x - x\,\xi_2^2\,(\xi_1)_x + \xi_1\,\xi_2\,(\xi_2)_t + \xi_1\,\phi_1\,(\xi_2)_u + \xi_1^2\,(\xi_2)_x + \xi_2^2\,(\xi_1)_{x,x} - 2\,\xi_2^2\,(\phi_1)_{x,u} == 0$$

$$-2\,\xi_2\,\phi_1\,(\xi_1)_x + \xi_2\,\phi_1\,(\xi_2)_t + \phi_1^2\,(\xi_2)_u + \xi_1\,\phi_1\,(\xi_2)_x - \xi_2^2\,(\phi_1)_t + x\,\xi_2^2\,(\phi_1)_x + \xi_2^2\,(\phi_1)_{x,x} == 0$$

$$2\,\xi_1\,(\xi_1)_u + 2\,x\,\xi_2\,(\xi_1)_u - 2\,\xi_2\,(\xi_1)_{x,u} + \xi_2\,(\phi_1)_{u,u} == 0$$

Again, we assume that $\xi_2 = 1$ thus we can simplify the non-classical determining equations by

```
deteqFP = DeleteCases[(ncfokkerPlanck /. u[x, t] → u) /.
    xi[2] → Function[{x, t, u}, 1], 0] /. u → u[x, t];
deteqFP // LTF
```

$$(\xi_1)_{u,u} == 0$$
$$-\xi_1 - (\xi_1)_t + 2\,\phi_1\,(\xi_1)_u - x\,(\xi_1)_x - 2\,\xi_1\,(\xi_1)_x + (\xi_1)_{x,x} - 2\,(\phi_1)_{x,u} == 0$$
$$-2\,\phi_1\,(\xi_1)_x - (\phi_1)_t + x\,(\phi_1)_x + (\phi_1)_{x,x} == 0$$
$$2\,x\,(\xi_1)_u + 2\,\xi_1\,(\xi_1)_u - 2\,(\xi_1)_{x,u} + (\phi_1)_{u,u} == 0$$

If we apply PDESolve[] to the derived equations, the solution of the non-classical determining equations follow. This function originally designed for linear PDEs is capable of solving some kind of non-linear equations if some hints are supplied to the function. In the present case, PDESolve[] detects a situation in which either a function *free*[2] or *free*[6] can be set equal to zero. The function asks the user to decide which of these possibilities it should take. The following calculation is carried out under the second possibility:

```
infl = PDESolve[deteqFP, {u}, {x, t}]; infl // LTF
```

```
There exists no unique solution of the equations.

Please choose one of the following results

1,{free[2][x, t] → 0}

2,{free[6][x, t] → 0}
```

$$\xi_1 == \mathcal{F}_1$$
$$\phi_1 == \mathcal{F}_2 + u\,\mathcal{F}_3$$
$$-\mathcal{F}_1 - (\mathcal{F}_1)_t - x\,(\mathcal{F}_1)_x - 2\,\mathcal{F}_1\,(\mathcal{F}_1)_x - 2\,(\mathcal{F}_3)_x + (\mathcal{F}_1)_{x,x} == 0$$
$$-2\,\mathcal{F}_2\,(\mathcal{F}_1)_x - (\mathcal{F}_2)_t + x\,(\mathcal{F}_2)_x + (\mathcal{F}_2)_{x,x} == 0$$
$$-2\,\mathcal{F}_3\,(\mathcal{F}_1)_x - (\mathcal{F}_3)_t + x\,(\mathcal{F}_3)_x + (\mathcal{F}_3)_{x,x} == 0$$

The result of the calculation is a general representation of the infinitesimals depending on three arbitrary functions $\mathcal{F}_i$ = *free*[*i*], i=1, 2, 3 which have to satisfy three nonlinear coupled PDEs. These PDEs again allow some symmetries. In the following, we determine the point symmetries of the non-classical non-linear determining equations. We gain a simpler representation by renaming the arbitrary functions *free*[*i*] by

```
infl1h = infl[[2]] /. {free[1] → g, free[2] → h, free[3] → f};
infl1h // LTF
```

$$-g - 2\,f_x - g_t - 2\,g\,g_x - x\,g_x + g_{x,x} == 0$$
$$-2\,h\,g_x - h_t + x\,h_x + h_{x,x} == 0$$
$$-f_t + x\,f_x - 2\,f\,g_x + f_{x,x} == 0$$

For this set of equations, we determine the infinitesimals by

```
inf1hh = Infinitesimals[inf1h, {g, h, f}, {x, t}]; inf1hh // LTF
```

```
PDESolve::nsf : Use option Standard->True.
 This may lead to further solutions in case of linear Systems
```

$$-(\mathcal{F}_1)_t + x\,(\mathcal{F}_1)_x + (\mathcal{F}_1)_{x,x} == 0$$

$$\xi_1 == E^{-t}\,k1 + E^{t}\,k2 + \frac{1}{2}\,E^{-2\,t}\,(k3 + E^{4\,t}\,k4)\,x$$

$$\xi_2 == -\frac{1}{2}\,E^{-2\,t}\,k3 + \frac{1}{2}\,E^{2\,t}\,k4 + k5$$

$$\phi_1 == -\frac{1}{2}\,E^{-2\,t}\,(2\,E^{t}\,k1 - 2\,E^{3\,t}\,k2 + g\,k3 + E^{4\,t}\,g\,k4 + 2\,k3\,x - 2\,E^{4\,t}\,k4\,x)$$

$$\phi_2 == -E^{-2\,t}\,h\,k3 - \frac{3}{2}\,E^{2\,t}\,h\,k4 + h\,k6 - E^{t}\,h\,k2\,x - \frac{1}{2}\,E^{2\,t}\,h\,k4\,x^2 + f\,\mathcal{F}_1 - (g + x)\,(\mathcal{F}_1)_x - (\mathcal{F}_1)_{x,x}$$

$$\phi_3 == -E^{-2\,t}\,(f\,(k3 + E^{4\,t}\,k4) + E^{3\,t}\,g\,(k2 + E^{t}\,k4\,x) + E^{3\,t}\,(E^{t}\,k4 + k2\,x + E^{t}\,k4\,x^2))$$

The infinitesimals represent a six-dimensional finite group and an infinite one given by the arbitrary function *free*[*1*]. Surprisingly, this function must satisfy the original equation with which we started. Now you see another reason why we changed the names of the three arbitrary functions in the non-classical analysis preventing mismatches of the two calculations. Knowing the point symmetries of the non-classical determining equations, we can reduce the equations by applying LieReduction[]:

```
infi = {{xi[1][x, t, g, h, f], xi[2][x, t, g, h, f]},
      {phi[1][x, t, g, h, f], phi[2][x, t, g, h, f],
       phi[3][x, t, g, h, f]}} /. inf1hh[[1]] /.
   {k1 → 1, k2 → 0, k3 → 0, k4 → 0,
    k5 → 1, k6 → 0, free[___] → Function[{x, t}, 0]} //
  Expand
```

```
{{E^-t, 1}, {-E^-t, 0, 0}}
```

For the sub-group $k1 = k5 = 1$, the reduction is gained by

```
red1 = LieReduction[inf1h, {g, h, f}, {x, t}, infi[[1]], infi[[2]]];
(red1 /. zeta1 → ξ1) // Flatten // LTF
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

$$E^{-t} + x - \zeta_1 == 0$$
$$g + x - F_1 == 0$$
$$h - F_2 == 0$$
$$f - F_3 == 0$$
$$F_1 - 2\,F3_{\zeta_1} - 2\,F_1\,(F_1)_{\zeta_1} + \zeta_1\,(F_1)_{\zeta_1} + (F_1)_{\zeta_1,\zeta_1} == 0$$
$$2\,F_2 - 2\,F_2\,F1_{\zeta_1} + \zeta_1\,(F_2)_{\zeta_1} + (F_2)_{\zeta_1,\zeta_1} == 0$$
$$2\,F_3 - 2\,F_3\,F1_{\zeta_1} + \zeta_1\,(F_3)_{\zeta_1} + (F_3)_{\zeta_1,\zeta_1} == 0$$

We find a coupled non-linear system of ODEs for the similarity functions F1, F2, and F3. A special solution follows by choosing F2 = F3 = 0:

```
red1[[3]] /. {F2 → Function[zeta1, 0], F3 → Function[zeta1, 0]} /.
  zeta1 → ζ1 // LTF
```

$$F_1 - 2\,F_1\,(F_1)_{\zeta_1} + \zeta_1\,(F_1)_{\zeta_1} + (F_1)_{\zeta_1,\zeta_1} == 0$$
```
True
True
```

The remaining ODE for F1 is solved by means of DSolve[]:

```
psol = DSolve[F1[zeta1] + zeta1 F1'[zeta1] -
    2 F1[zeta1] F1'[zeta1] + F1''[zeta1] == 0, F1,
  zeta1]
```

$$\left\{\left\{F1 \to \left(\frac{-2\,E^{\frac{\#1^2}{2}}\sqrt{\#1^2} + 2\,\#1^2\,C[1] + \sqrt{2\pi}\,\#1^2\,\text{Erfi}\left[\frac{\sqrt{\#1^2}}{\sqrt{2}}\right]}{\#1\left(2\,C[1] + \sqrt{2\pi}\,\text{Erfi}\left[\frac{\sqrt{\#1^2}}{\sqrt{2}}\right]\right)}\&\right)\right\}\right\}$$

The solution for F1 contains the special function Erfi[]. The inversion of the used transformations provides the solution for the unknown functions *free*[*1*], *free*[*2*], and *free*[*3*] of the non-classical symmetries:

```
solg =
 {free[1] → Function[{x, t}, g], free[2] → Function[{x, t}, h],
   free[3] → Function[{x, t}, f]} /.
  Solve[(red1[[2]] /. psol /. {F2 → Function[zeta1, 0],
       F3 → Function[zeta1, 0]}) // Flatten,
   {g, h, f}]
```

$$\{\{\text{free}[1] \to \text{Function}\Big[\{x, t\}, -x + \Big(-2\, E^{\frac{1}{2}(E^{-t}+x)^2}\sqrt{(E^{-t}+x)^2} + 2\,(E^{-t}+x)^2\, C[1] + \sqrt{2\pi}\,(E^{-t}+x)^2\, \text{Erfi}\Big[\frac{\sqrt{(E^{-t}+x)^2}}{\sqrt{2}}\Big]\Big) \Big/ \Big((E^{-t}+x)\Big(2\,C[1] + \sqrt{2\pi}\,\text{Erfi}\Big[\frac{\sqrt{(E^{-t}+x)^2}}{\sqrt{2}}\Big]\Big)\Big)\Big],$$
$$\text{free}[2] \to \text{Function}[\{x, t\}, 0],$$
$$\text{free}[3] \to \text{Function}[\{x, t\}, 0]\}\}$$

The check of the initial determining equations reveals that, in fact, the given functions satisfy the determining equations:

```
inf1[[2]] /. solg // Simplify
```

```
{{0, 0, 0}}
```

Finally, the non-classical symmetry transformations for the FP equation under the condition that the non-linear determining equations allow point symmetries are given by

```
ncinfi = Append[inf1[[1]] /. solg /. C[1] → k1 /. u[x, t] → u /.
    Rule[a_[n_][h___], b_] →
     Rule[a[n], Function[$v, $w] /. {$v → {h}, $w → b}] //
   Flatten,
  xi[2] → Function[{x, t, u}, 1]];
ncinfi // LTF
```

$$\phi_1 == 0$$

$$\xi_1 == -x + \frac{2\,k1\,(E^{-t}+x)^2 - 2\,E^{\frac{1}{2}(E^{-t}+x)^2}\sqrt{(E^{-t}+x)^2} + \sqrt{2\pi}\,(E^{-t}+x)^2\,\text{Erfi}\Big[\frac{\sqrt{(E^{-t}+x)^2}}{\sqrt{2}}\Big]}{(E^{-t}+x)\Big(2\,k1 + \sqrt{2\pi}\,\text{Erfi}\Big[\frac{\sqrt{(E^{-t}+x)^2}}{\sqrt{2}}\Big]\Big)}$$

$$\xi_2 == 1$$

At this stage of the calculation, we find a very special representation of the non-classical symmetries. The next step in the calculation should be the reduction of the FP equation. A glance at the above result reveals that this last step is very difficult because the infinitesimals contain special functions with very complicated arguments. Since the infinitesimals are part of a first-order PDE, we face the problem of solving these equations to find the invariants. Currently, neither *Mathematica* nor *MathLie* can solve this kind of equations. However, the result derived via a mixed application of Lie's classical and non-classical method demonstrates that a simple equation allows a very complicated structure of symmetries.

However, solutions of the FP equation are derivable if we make an ansatz for the non-classical infinitesimals of the form

```
inf2 = inf1 /. {free[1] → Function[{x, t}, free[1][x]],
   free[2] → Function[{x, t}, 0], free[3] → Function[{x, t}, 0]};
inf2 // LTF
```

$$\xi_1 == \mathcal{F}_1$$
$$\phi_1 == 0$$
$$-\mathcal{F}_1 - x\,(\mathcal{F}_1)_x - 2\,\mathcal{F}_1\,(\mathcal{F}_1)_x + (\mathcal{F}_1)_{x,x} == 0$$
True
True

assuming that the arbitrary functions *free*[*2*] and *free*[*3*] are vanishing constants. The substitution into the non-linear determining equations of the FP equation reveals that only one equation for the function *free*[*1*] remains. So that the non-classical infinitesimals are determined by

```
inf3 = {Append[inf2[[1]], xi[2][x, t, u[x, t]] → 1],
  DeleteCases[inf2[[2]], 0]};
inf3 // LTF
```

$$-\mathcal{F}_1 - x\,(\mathcal{F}_1)_x - 2\,\mathcal{F}_1\,(\mathcal{F}_1)_x + (\mathcal{F}_1)_{x,x} == 0$$
$$\xi_1 == \mathcal{F}_1$$
$$\phi_1 == 0$$
$$\xi_2 == 1$$

Since the unknown function *free*[*1*] is a function of x only, we can again apply PDESolve[] to solve the determining equation for *free*[*1*]:

```
inf4 = PDESolve[inf3, {u}, {x, t}]; inf4 // LTF
```

$$\xi_1 == \frac{-2\sqrt{x^2} + k1\sqrt{2\pi}\,x\,\text{Erfi}\left[\frac{\sqrt{x^2}}{\sqrt{2}}\right]}{2\sqrt{x^2}\left(E^{\frac{x^2}{2}}\,k1 + x\right) - k1\sqrt{2\pi}\,x^2\,\text{Erfi}\left[\frac{\sqrt{x^2}}{\sqrt{2}}\right]}$$
$$\phi_1 == 0$$
$$\xi_2 == 1$$

The result is an explicit representation of the non-classical symmetries for the FP equation depending on a single parameter *k1*. This parameter is responsible for the transformation properties. Because PDESolve[] returns the infinitesimals in a special representation, we need to transform the infinitesimals into a pure function.

```
ncifokkerPlanck = inf4 /. u[x, t] → u /. Rule[a_[n_][h___], b_] →
    Rule[a[n], Function[$v, $w] /. {$v → {h}, $w → b}] // Flatten;
ncifokkerPlanck // LTF
```

$$\phi_1 == 0$$

$$\xi_1 == \frac{-2\sqrt{x^2} + k1\sqrt{2\pi}\, x\, \text{Erfi}\left[\frac{\sqrt{x^2}}{\sqrt{2}}\right]}{2\sqrt{x^2}\left(E^{\frac{x^2}{2}} k1 + x\right) - k1\sqrt{2\pi}\, x^2\, \text{Erfi}\left[\frac{\sqrt{x^2}}{\sqrt{2}}\right]}$$

$$\xi_2 == 1$$

In the following, we will restrict our considerations to a sub-class of transformations for which $k1 = 0$. The non-classical infinitesimals reduce to the simple form

```
infi = {{xi[1][x, t, u], xi[2][x, t, u]},
    {phi[1][x, t, u]}} /. ncifokkerPlanck /.
  k1 → 0
```

$$\left\{\left\{-\frac{1}{x}, 1\right\}, \{0\}\right\}$$

For this sub-class of infinitesimal transformation, we reduce the original FP equation by

```
red =
 LieReduction[fokkerPlanck, {u}, {x, t}, infi[[1]], infi[[2]]];
red /. zeta1 → ζ1 // Flatten // LTF
```

$$\frac{1}{2}(2t + x^2) - \zeta_1 == 0$$

$$u - F_1 == 0$$

$$-x^2 (F1_{\zeta_1} + F1_{\zeta_1,\zeta_1}) == 0$$

The resulting second-order ODE is solved by DSolve[]:

```
rsol = DSolve[red[[3, 1]], F1, zeta1]
```

$$\{\{F1 \to (-E^{-\#1} C[1] + C[2]\&)\}\}$$

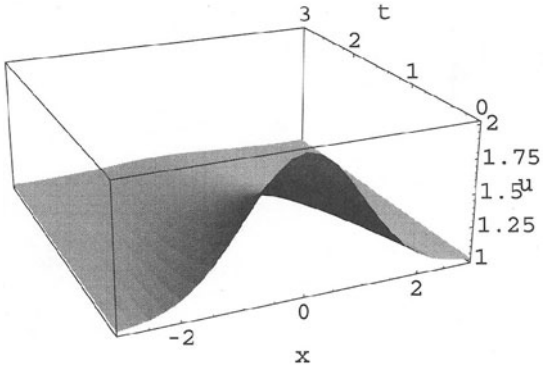
The solution is a decaying exponential function in the similarity variable $\zeta_1 = (2t + x^2)/2$. The solution for the original variable u follows by

```
sol = Solve[ red[[2]] /. rsol, u] // Flatten
```

$$\left\{u \to -E^{\frac{1}{2}(-2t-x^2)} C[1] + C[2]\right\}$$

representing a Gaussian in the spatial variable multiplied by an exponential decrease in time. The constants $C[1]$ and $C[2]$ are constants of integration and can be used to implement initial conditions. For a fixed value of $C[1]$ and $C[2]$, we plot the non-classical solution for the FP equation:

```
Plot3D[Evaluate[u /. sol /. {C[1] → -1, C[2] → 1}], {x, -3, 3},
 {t, 0, 3}, AxesLabel → {"x", "t", "u"}, PlotPoints → 25,
 PlotRange → All, ViewPoint -> {-1.253, -2.896, 1.221},
 Mesh → False]
```



We clearly observe that the solution represents a decaying solution in time. The initial shape of a Gaussian in x decreases in time but does not change its shape. The examination of the FP equation demonstrated that the non-classical method can be combined with the classical method. The results for the non-classical symmetries of the FP equation are, however, too complicated for *MathLie* to carry out the final stage of the solution procedure.

Up to now, we discussed local symmetries of a given equation. In the following chapter, we will describe the derivation of non-local symmetries. We will show that the Fréchet derivative is a useful tool for determining this type of symmetry.

7

Potential Symmetries of Partial Differential Equations

7.1. Introduction

The last two chapters discussed point symmetries and non-classical symmetries. These types of symmetry are local symmetries because the coordinates are involved in the local transformations in a direct way. This chapter discusses a completely different type of symmetry. We not only consider the original PDEs $\Delta = 0$ but also derived systems of PDEs whose solutions are solutions of the original equations. The new associated system of PDEs is treated by the methods discussed in the previous sections. The result of this treatment are symmetries not only depending on the local variables of the original equation but also on variables of the affiliated system of PDEs. Thus, we get a new type of symmetry depending on an extended set of variables. Such symmetries are generally called non-local symmetries. A special type of non-local symmetry is a potential symmetry. Our interest in this chapter are potential symmetries of PDEs.

The following sections discuss how the associated PDEs are derivable in a systematic way. For this reason, we need to introduce some new terminology. The new terms have the advantage that a generalization to other kinds of symmetry are possible.

7.2. Basics of Potential Symmetries

In Chapter 6, we distinguished weak and strong symmetries. If we have a strong symmetry, solutions of the equation $\Delta = 0$ are transformed into solutions of this equation, and a reduction of the original PDEs are possible. In case of weak symmetries, only a reduction is possible. Based on strong symmetries, we can generalize the meaning of a symmetry.

Definition: Topological symmetry

A symmetry of a system of PDEs $\Delta = 0$ is a transformation which transforms solutions of $\Delta = 0$ into other solutions of $\Delta = 0$. ○

The symmetry of the PDE given above is of a topological nature because it is completely free of any coordinates. We see that a symmetry primarily is not connected to a system of coordinates. However, if we need to carry out practical calculations, we have to consider the coordinates of the problem.

The classical symmetries of a PDE are point transformations which guarantee the invariance of the solution space. These kinds of symmetry are point symmetries as we know them. Point symmetries are created by infinitesimal transformations

$$x_i' = x_i + \epsilon \xi_i(x, u) + O(\epsilon^2), \quad i = 1, 2, \ldots, p, \tag{7.1}$$

$$u'^{\alpha} = u^{\alpha} + \epsilon \phi_{\alpha}(x, u) + O(\epsilon^2), \quad \alpha = 1, 2, \ldots, q. \tag{7.2}$$

The vector field of the infinitesimal transformation is not uniquely defined. Contrary to Chapter 5, we can represent the vector field by

$$\vec{v} = \sum_{\alpha=1}^{q} \eta_{\alpha}(x, u, u_{(1)}) \, \partial_{u^{\alpha}} \, . \tag{7.3}$$

The infinitesimal components η of the vector field $\vec{v}$ are given by

$$\eta_{\alpha} = \phi_{\alpha}(x, u) - \sum_{i=1}^{p} \xi_i(x, u) \, u_i^{\alpha}, \quad \alpha = 1, 2, \ldots, q, \tag{7.4}$$

which are equivalent to the characteristics Q_{α} introduced in Chapter 6 in connection with non-classical symmetries. Based on the characteristics, the kth prolongation of the vector field simplifies to

$$\mathrm{pr}^{(k)}\,\vec{v} \;=\; \vec{v} + \sum_{\alpha=1}^{q}\sum_{J}(D_J\,\eta_\alpha)\,\partial_{u_J^\alpha}\,. \tag{7.5}$$

We call a transformation local if the characteristics η_α depend on derivatives of the dependent variables. The invariance condition from which solutions for the characteristics follow is given by

$$\mathfrak{D}_\Delta(\eta)\,|_{\Delta=0} \;= 0, \tag{7.6}$$

where $\mathfrak{D}$ denotes the Fréchet derivative. Relation (7.6) is nothing more than the invariance condition discussed in Chapter 5 for point symmetries. We note that this type of invariance condition does not change if the characteristics depend on higher derivatives. This situation occurs if we consider generalized symmetries. In Chapter 9, we will discuss this type of symmetry. The Fréchet derivative, on the other hand, is the basic tool for the systematic calculation of potential symmetries. How we access the potential symmetry will be discussed next.

7.3. Calculation of Potential Symmetries

When describing non-local symmetries, it is convenient to introduce new variables $v(x)$ by additional equations which are connected with the original equations in the variables $u(x)$. One condition on the new system of PDEs is that the original system must be derivable from the new. In other words, if $(u(x), v(x))$ satisfy the extended equations, $u(x)$ also has to be a solution of the original system,

$$\Delta(x, u_{(k)}) \;= 0. \tag{7.7}$$

An auxiliary system with new variables can be introduced if at least one PDE, say $\Delta_m = 0$, of the system $\Delta = 0$ can be written as a conservation law (cf. Bluman and Kumei [1989]).

Suppose one PDE of the system $\Delta = 0$, without loss of generality $\Delta_m = 0$, can be expressed as a conservation law

$$\sum_{i=1}^{n} D_i\, f^i(x, u_{(k-1)}) \;= 0. \tag{7.8}$$

Then, the system $\Delta = 0$ can be written in the form

$$\Delta_\nu(x, u_{(k)}) \;= 0, \qquad \nu = 1, 2, \ldots, m-1, \tag{7.9}$$

$$\sum_{i=1}^{n} D_i\, f^i(x, u_{(k-1)}) \;= 0. \tag{7.10}$$

Relation (7.10) allows us to introduce $n-1$ new variables $v = (v^1, v^2, \ldots, v^{n-1})$. These $n-1$ variables have to satisfy $n-1$ equations. In connection with the remaining equations (7.9), they built up a new system of equations $\Psi(x, u_{(k)}, v_{(1)}) = 0$, which is the potential system. Explicitly, we find

$$f^1(x, u_{(k-1)}) = v^1_{x_2}, \tag{7.11}$$

$$f^l(x, u_{(k-1)}) = (-1)^{l-1}\left\{v^l_{x_{l+1}} + v^{l-1}_{x_{l-1}}\right\}, \qquad 1 < l < n, \tag{7.12}$$

$$\Delta_\nu(x, u_{(k)}) = 0, \qquad \nu = 1, 2, \ldots, m-1. \tag{7.13}$$

The original system Δ is closely connected to the potential system Ψ, meaning that some properties of Δ are still contained in Ψ. One of these properties is that if (u, v) are solutions of Ψ then u is also a unique solution of Δ. On the other hand if u is a solution of Δ then there exists a v so that (u, v) is also a solution of Ψ. However, v in this case is not unique. These properties of the solutions are also present in the symmetries of both systems. A symmetry of Ψ is connected with a symmetry of Δ and vice versa. The symmetries, however, have different meanings for the two systems. For example, a point symmetry of Ψ may be a non-local symmetry of Δ. This kind of symmetry is called potential symmetry.

Definition: Potential symmetry

Let us assume that Ψ is a potential system of Δ. A point symmetry of Ψ given by

$$\begin{aligned}\vec{v}_\Psi = &\sum_{\alpha=1}^{m}\left\{\phi_\alpha(x, u, v) - \sum_{i=1}^{p}\xi_i(x, u, v)\, u^\alpha_i\right\}\partial_{u^\alpha} \\ &+ \sum_{\beta=1}^{n-1}\left(\chi_\beta(x, u, v) - \sum_{i=1}^{n}\xi_i(x, u, v)\, v^\beta_i\right)\partial_{v^\beta}\end{aligned} \tag{7.14}$$

with ξ_i, ϕ_α, and χ_β the infinitesimals of x, u, and v, respectively, is a potential symmetry of Δ, if the infinitesimals ξ_i and ϕ_α depend on the new variables v^β and are not reducible to a point symmetry of Δ. ○

This definition introduces a potential symmetry as a point symmetry of an auxiliary system, the potential system. We already know how point symmetries of a PDE are determined. In Chapter 5, we discussed the procedure for different examples. Thus, we face the problem of finding the potential system for which we can calculate the point symmetries.

The main problem in determining potential symmetries is to find useful potential systems Ψ. To express one PDE of the system (7.7), $\Delta_i = \sum_{i=1}^{n} D_i f^i$ in a conserved

form is not the only possibility to rewrite a system of PDEs. Additionally, a number of PDEs of the system (7.7) can be multiplied by integrating factors to become conserved quantities, as Bluman [1993] remarks. There exists a set of integrating multipliers

$$\{\lambda^{\nu}(x, u_{(p)})\} \tag{7.15}$$

which allows the representation

$$\sum_{\nu=1}^{m} \lambda^{\nu}(x, u_{(p)})\, \Delta_{\nu}(x, u_{(k)}) = \sum_{i=1}^{n} D_i\, f^{i}(x, u_{(k)}). \tag{7.16}$$

In replacing the PDE Δ_{ν} by relation (7.16), we must be cautious. The resulting new system Ψ may gain a solution which is not a solution of the original system Δ. However, every solution of Δ is a solution of Ψ, but each solution of the system

$$\Delta_{\nu} = 0 \quad \nu = 1, \ldots, \mu - 1, \mu + 1, \ldots, m, \tag{7.17}$$

$$\lambda^{\mu}(x, u_{(p)}) = 0 \tag{7.18}$$

satisfies the potential system Ψ. So there may exist solutions of Ψ which do not solve the original system. In fact, we have to exclude such solutions to prevent trouble in the calculation. So let us define the following:

Definition: Equivalence of conserved representations

A set of integrating factors $\{\lambda^{\nu}\}$ creates an equivalent conserved representation of Δ if for a single μ, $1 \leq \mu \leq m$, the systems Δ and Ψ allow a common space of solutions. Potential symmetries exist only for equivalent conserved representations. ○

Experience shows that integrating multipliers λ depending on derivatives do not create equivalent conserved representations. Thus, we restrict our considerations to multipliers depending only on the independent and dependent variables

$$\lambda(x, u) = \{\lambda^{1}(x, u), \lambda^{2}(x, u), \ldots, \lambda^{m}(x, u)\}. \tag{7.19}$$

The last step in our theoretical considerations is the determination of the integrating multipliers. So we need a formula allowing us the systematic determination of the λ^{ν}'s. If we apply the invariance criterion (7.6) to the potential system (7.11)–(7.13) in connection with relation (7.16), we end up with the condition

$$\mathfrak{D}^*_\Delta(\lambda)\,|_{\Delta=0} \;= 0, \tag{7.20}$$

where $\mathfrak{D}^*$ denotes the adjoint Fréchet derivative. The solutions for the λ's follow by calculating the invariance condition with the adjoint Fréchet derivative and replacing the characteristics η in (7.6) by the integrating multipliers. Since *MathLie* is able to treat such a system of equations, we have a tool providing the solutions.

The solutions of (7.20) may be either a finite set of solutions or a continuous set, meaning that for the finite set, a finite number of solutions exists, whereas in the continuous case, arbitrary functions occur in the integrating multipliers λ. Both cases are useful in classifying the point symmetries of the related potential system Ψ.

The theory so far discussed can again be applied to a derived potential system Ψ. This recursive application of the procedure delivers a tree of the potential system if the integrating multipliers are chosen in an appropriate way.

Thus, we are able to construct new solutions by means of the potential symmetries. These new solutions are quite different from the solutions derived from point symmetries because the non-local properties of the equation are the key for this construction. Solution (u, v) of the potential system allows us to find solutions of the original system Δ. These solutions are new solutions, different from the solutions derived via the classical or non-classical method of Lie. Completely new solutions may be expected if we analyze the potential system by means of the non-classical method.

We use the procedure of integrating factors extensively in our package *MathLie* to derive the potential systems. The algorithmic procedure to derive potential symmetries is summarized in the following four steps:

1. Determine the integrating multipliers from relation (7.20).
2. Extract these integrating multipliers which allow an equivalent conservation law.
3. Create for each integrating factor from step 3 the potential representation Ψ.
4. Examine the potential system either with the classical or the non-classical method.

These four steps are the basis of the following examples demonstrating the automatic determination of potential symmetries. The calculations are supported by the function PotentialSymmetries[], which is part of *MathLie*. The function PotentialSymmetries[] allows us to derive the possible potential systems and calculate the point symmetries of these systems.

7.4. Applications of Potential Symmetries

In the following sections, we apply the theory to several examples. The discussed theory allows us to derive new solutions for a scalar PDE as well as for systems of PDEs. The program *MathLie* calculates automatically all potential systems and determines the corresponding point symmetries of the potential system. The application of the non-classical method to potential systems is novel. As we will see, there are a lot of problems when non-classical symmetries are used to find solutions.

Let us demonstrate some of the possible calculations carried out by *MathLie* in connection with potential symmetries. The following examples demonstrate the application of the function PotentialSymmetries[].

7.4.1 A Non-linear Reaction Diffusion Equation

The first example demonstrated how the potential systems are derived by *MathLie*. We examine a non-linear reaction diffusion equation of the type

```
reaktionDiffusion = ∂t u[x, t] + ∂x,x (1/u[x, t] + β x^2) == 0;
reaktionDiffusion // LTF
```

$$2\beta + u_t + \frac{2u_x^2}{u^3} - \frac{u_{x,x}}{u^2} == 0$$

where β is a real constant. Our primary interest is to detect a potential representation of the equation. The function PotentialSymmetries[] in connection with the option *PotentialSystemsOnly→True* allows us to derive these systems. The function needs the equation, the dependent and independent variables, and the parameters as input quantities.

```
pSystemOfReaktionDiffusion =
 PotentialSymmetries[ reaktionDiffusion,
  u, {x, t}, {β}, PotentialSystemsOnly → True];
pSystemOfReaktionDiffusion // LieTraditionalForm
```

$$\left\{\left\{\frac{1}{u} + x^2\beta + V10_t,\ -V_8 - V10_x,\ -u - (V_8)_x\right\},\right.$$
$$\{u, V_8, V10\}, \{x, t\}\Big\}, \left\{\left\{\frac{1}{u\,x} + x\beta + V11_t,\right.\right.$$
$$\left.\left.-\frac{V_9}{x^2} - V11_x,\ -u\,x - (V_9)_x\right\}, \{u, V_9, V11\}, \{x, t\}\right\},$$
$$\left\{\left\{-\frac{E^{-\beta V_9}}{u\,x\,\beta} + V12_t,\ -\frac{E^{-\beta V_9}}{x^2\,\beta^2} - V12_x,\ -u\,x - (V_9)_x\right\}, \{u, V_9, V12\},\right.$$
$$\left.\{x, t\}\right\},$$

$$\left\{\left\{\frac{1}{u\,x}+x\,\beta+V14_t,\ -\frac{V_9}{x^2}-V14_x,\ -\frac{E^{-\beta V_9}}{u\,x\,\beta}+V13_t,\ -\frac{E^{-\beta V_9}}{x^2\,\beta^2}-V13_x\right\},\right.$$
$$\left.\{u,\ V_9,\ V13,\ V14\},\ \{x,\ t\}\right\}\}$$

The result is a nested list containing four lists. Each of these sub-lists contains information on a specific potential system. Added to each potential system are lists containing the dependent and independent variables of the system. We observe that three of the potential systems are formulated for three dependent variables and that one of the potential systems contains four dependent variables. This behavior indicates that the fourth potential system is a second-stage potential system derived from a precursor. We learn that not only the original equation is checked on a possible potential representation but also all potential systems derived from the original one.

7.4.2 Cylindrical Korteweg-de Vries Equation

The second example examines the symmetries of the potential systems for the cylindrical KdV equation. The cylindrical KdV (cKdV) equation is given by

```
cKdV =
 ∂t u[x, t] + u[x, t]/(2 t) + u[x, t] ∂x u[x, t] + ∂x,x,x u[x, t] == 0;
cKdV // LTF
```

$$\frac{u}{2\,t}+u_t+u\,u_x+u_{x,x,x}==0$$

The potential systems of the cKdV equation are calculated by

```
pcKdV = PotentialSymmetries[cKdV, u, {x, t},
  PotentialSystemsOnly → True, OrderReduce → False]; pcKdV //
 TableForm[LieTraditionalForm[#], TableSpacing -> {1, 1}]&
```

$$-\frac{1}{2}\sqrt{t}\,u^2+V14_t-\sqrt{t}\,u_{x,x}$$
$$-\sqrt{t}\,u-V14_x \qquad \begin{matrix} u & x \\ V14 & t \end{matrix}$$

$$-\frac{t\,u^3}{3}+\frac{t\,u_x^2}{2}+V15_t-t\,u\,u_{x,x}$$
$$-\frac{t\,u^2}{2}-V15_x \qquad \begin{matrix} u & x \\ V15 & t \end{matrix}$$

$$-\frac{1}{3}t^{3/2}u^3+\frac{1}{4}\sqrt{t}\,u^2\,x-\frac{\sqrt{t}\,u_x}{2}+\frac{1}{2}t^{3/2}u_x^2$$
$$+V16_t-\left(t^{3/2}\,u-\frac{\sqrt{t}\,x}{2}\right)u_{x,x} \qquad \begin{matrix} u & x \\ V16 & t \end{matrix}$$
$$-\frac{1}{2}\,t^{3/2}\,u^2+\frac{1}{2}\sqrt{t}\,u\,x-V16_x$$

The result is a set of three potential systems of first stage. For this equation PotentialSymmetries[] does not find derived potential systems of higher order. We used the option *OrderReduce→False* for the above calculation to find all reductions of the original equation. The function PotentialSymmetries[] assumes by default that the auxiliary system is reduced in order and thus only returns a special class of potential systems. However, we can control this behavior by the option *OrderReduce* and can choose the suitable method.

The next step is the solution of the potential systems to find the related potential symmetries. Let us start with the first potential system. We calculate the infinitesimal transformations of this system by selecting the first argument of *pcKdV* and feed them into the function Infinitesimals[]:

```
infpcKdV =
 Infinitesimals[pcKdV[[1, 1]], pcKdV[[1, 2]], pcKdV[[1, 3]]];
infpcKdV // LTF
```

$$\phi_1 == \frac{-k3 - 4\,k1\,\sqrt{t}\,u + 8\,k4\,t\,u - 2\,k4\,x}{\sqrt{t}}$$

$$\phi_2 == k2 + k1\,V1 + x\,(k3 + k4\,x)$$

$$\xi_1 == k5 - 2\,k3\,\sqrt{t} + 2\,(k1 - 2\,k4\,\sqrt{t})\,x$$

$$\xi_2 == 6\,k1\,t - 8\,k4\,t^{3/2}$$

The result represents a five-dimensional finite symmetry group which is similar to the point symmetries of the cKdV equation. The difference exists in the dimension of the group and in the number of elements. The point symmetry of the cKdV equation consists of three expressions and the potential symmetries of four. Recalling the definition of a potential symmetry, we have to check the dependence of the infinitesimals for the original equation on variables introduced in the derivation of the potential system. If we examine the infinitesimals *xi[1]*, *xi[2]*, and *phi[1]* on the dependence of variables starting with capital letter *V*, we quickly realize that these infinitesimals are independent of the new potential variables. In conclusion, the first potential system of the cKdV equation does not contribute to potential symmetries. In order to check all possibilities for the above potential system, we carry out the calculation by

```
allInfPcKdV =
 Map[Infinitesimals[#[[1]], #[[2]], #[[3]]]&, pcKdV]; allInfPcKdV /.
    {Rule → Equal, HoldPattern[Function[x_, y_]] → y} /.
   TraditionalLieForm // Flatten //
 TableForm
```

$$\phi_1 == \frac{-k3 - 4\,k1\,\sqrt{t}\,u + 8\,k4\,t\,u - 2\,k4\,x}{\sqrt{t}}$$

$$\xi_1 == k5 - 2\,k3\,\sqrt{t} + 2\,(k1 - 2\,k4\,\sqrt{t})\,x$$

$$\xi_2 == 6\,k1\,t - 8\,k4\,t^{3/2}$$

$$\phi_2 == k2 + k1\,V1 + x\,(k3 + k4\,x)$$

$$\phi_1 == -2\,k3\,u$$

$$\phi_2 == k1$$

$$\xi_1 == k2 + k3\,x$$

$$\xi_2 == 3\,k3\,t$$

$$\phi_1 == 0$$

$$\xi_1 == 0$$

$$\xi_2 == 0$$

$$\phi_2 == k1$$

Checking the dependencies of the infinitesimals on the potential variables, we realize that none of the infinitesimals satisfies the criterion for potential symmetries. However, the function PotentialSymmetries[] makes it easy to verify the definition and to decide whether a given equation allows potential symmetries or not. Experience with this function tells us that potential symmetries are rare symmetries but occur in connection with some equations. The above two-step calculation can be done in one step by suppressing the option *PotentialSystemsOnly→True*:

```
PotentialSymmetries[cKdV, u, {x, t}, OrderReduce → False]
```

$$\Big\{\Big\{-\frac{1}{2}\sqrt{t}\,u[x,t]^2 + V17^{(0,1)}[x,t] - \sqrt{t}\,u^{(2,0)}[x,t],$$

$$-\sqrt{t}\,u[x,t] - V17^{(1,0)}[x,t]\Big\},$$

$$\{u, V17\}, \{x, t\}, \Big\{\text{phi}[1] \to \text{Function}\Big[$$

$$\{x, t, u, V17\}, \frac{-k3 - 4\,k1\,\sqrt{t}\,u + 8\,k4\,t\,u - 2\,k4\,x}{\sqrt{t}}\Big],\ \text{xi}[1] \to$$

$$\text{Function}[\{x, t, u, V17\}, k5 - 2\,k3\,\sqrt{t} + 2\,(k1 - 2\,k4\,\sqrt{t})\,x],$$

$$\text{xi}[2] \to \text{Function}[\{x, t, u, V17\}, 6\,k1\,t - 8\,k4\,t^{3/2}],$$

$$\text{phi}[2] \to \text{Function}[\{x, t, u, V17\}, k2 + k1\,V17 + x\,(k3 + k4\,x)]\}\Big\},$$

$$\Big\{\Big\{-\frac{1}{3}\,t\,u[x,t]^3 + V18^{(0,1)}[x,t] + \frac{1}{2}\,t\,u^{(1,0)}[x,t]^2 -$$

$$t\,u[x,t]\,u^{(2,0)}[x,t], -\frac{1}{2}\,t\,u[x,t]^2 - V18^{(1,0)}[x,t]\Big\},$$

$$\{u, V18\}, \{x, t\}, \{\text{phi}[1] \to \text{Function}[\{x, t, u, V18\}, -2\,k3\,u],$$

$$\text{phi}[2] \to \text{Function}[\{x, t, u, V18\}, k1],$$

$$\text{xi}[1] \to \text{Function}[\{x, t, u, V18\}, k2 + k3\,x],$$

$$\text{xi}[2] \to \text{Function}[\{x, t, u, V18\}, 3\,k3\,t]\}\Big\},$$

$$\left\{\left\{\frac{1}{4}\sqrt{t}\,x\,u[x,t]^2 - \frac{1}{3}t^{3/2}u[x,t]^3 + V19^{(0,1)}[x,t] - \frac{1}{2}\sqrt{t}\,u^{(1,0)}[x,t] + \frac{1}{2}t^{3/2}u^{(1,0)}[x,t]^2 - \left(-\frac{\sqrt{t}\,x}{2} + t^{3/2}u[x,t]\right)u^{(2,0)}[x,t],\right.\right.$$
$$\left.\left.\frac{1}{2}\sqrt{t}\,x\,u[x,t] - \frac{1}{2}t^{3/2}u[x,t]^2 - V19^{(1,0)}[x,t]\right\},$$
$$\{u, V19\}, \{x, t\}, \left\{\text{phi}[1] \to \text{Function}\left[\{x, t, u, V19\}, -\frac{2\,(-6\,k2 + k4\sqrt{t}\,u + 36\,k3\,t\,u - 9\,k3\,x)}{3\sqrt{t}}\right],\right.$$
$$\text{xi}[1] \to \text{Function}\left[\{x, t, u, V19\}, \frac{k4\,x}{3} + 4\sqrt{t}\,(2\,k2 + 3\,k3\,x)\right],$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u, V19\}, (k4 + 24\,k3\sqrt{t})\,t],$$
$$\left.\text{phi}[2] \to \text{Function}\left[\{x, t, u, V19\}, k1 + k5 + 3\,k3\,t + \frac{k4\,V19}{2} + k2\,x^2 + k3\,x^3\right]\right\}\}\}$$

The resulting list now contains the information on the potential systems extended by the infinitesimals. The outcome is the same as before. No potential symmetries are present.

7.4.3 The Burgers Equation

Another example which allows potential symmetries is the Burgers equation:

```
Burgers = ∂t u[x, t] + u[x, t] ∂x u[x, t] + γ ∂x,x u[x, t] == 0;
Burgers // LTF
```

$$u_t + u\,u_x + \gamma\,u_{x,x} == 0$$

where γ is a real constant and measures the strength of second-order dispersion effects. The potential system and the potential symmetries are calculated with

```
pBurgers = PotentialSymmetries[Burgers, u, {x, t}, {γ},
  OrderReduce → False]
```

$$\left\{\left\{\left\{-\frac{1}{2}u[x,t]^2 + V24^{(0,1)}[x,t] - \gamma\,u^{(1,0)}[x,t], -u[x,t] - V24^{(1,0)}[x,t]\right\}, \{u, V24\}, \{x, t\},\right.\right.$$
$$\left.\left.\left\{\text{xi}[1] \to \text{Function}\left[\{x, t, u, V24\}, k3 - k6\,t + \frac{1}{2}(k2 - 4\,k7\,t)\,x\right],\right.\right.\right.$$
$$\text{xi}[2] \to \text{Function}[\{x, t, u, V24\}, k1 + t\,(k2 - 2\,k7\,t)],$$
$$\left.\left.\left.\text{phi}[2] \to \text{Function}\left[\{x, t, u, V24\}, k4 + k5 + k6\,x + k7\,x^2 - 2\,k7\,t\,\gamma + 2\,E^{\frac{V24}{2\gamma}}\,\gamma\,\text{free}[1][x,t]\right]\right.\right.\right.$$

$$\text{phi}[1] \to \text{Function}\Big[\{x,\ t,\ u,\ V24\},\ -k6 - \frac{k2\,u}{2} + 2\,k7\,t\,u - 2\,k7\,x + E^{\frac{V24}{2\gamma}}\,u\,\text{free}[1][x,\ t] - 2\,E^{\frac{V24}{2\gamma}}\,\gamma\,\text{free}[1]^{(1,0)}[x,\ t]\Big\},$$

$$\Big\{\frac{\text{free}[1]^{(0,1)}[x,\ t]}{\gamma} + \text{free}[1]^{(2,0)}[x,\ t]\Big\}\Big\}\Big\}$$

For the potential system of the Burgers equation, we find a seven-dimensional finite group and an infinite group. The infinite part of the group was derived by Vinogradov and Krasil'shchik [1984]. In addition, the auxiliary function *free[1]* of this continuous group has to satisfy the diffusion equation.

8

Approximate Symmetries of Partial Differential Equations

8.1. Introduction

The theory of approximate symmetries was developed by Baikov, Gazizov, and Ibragimov [1989] in the 1980s. The idea behind this development was the extension of Lie's theory to situations in which a small perturbation of the original equation is encountered. For such cases, the question arises of how the point symmetries or the group of the equations are altered if a small perturbation is added to the original equation. This question initiated the development of a group analysis method that is stable under small perturbations of the differential equation. The present chapter discusses the method of approximate symmetries. The method is based on the concept of an approximate group of transformations. Approximate symmetries are useful for partial differential equations depending on a small parameter ϵ. This parameter is usually used in the standard theories to examine the differential equation in some limit. On the other hand, this parameter is also useful in the examination of Lie point symmetries.

The basics of the theory were recently developed by Baikov et al. [1991]. These authors showed that the main part of Lie's theory can be used in an approximate calculus taking into account the smallness of the critical parameter in the theory. The new theory maintains the essential features of the standard Lie theory. This chapter

provides a concise introduction to the theory of approximate transformation groups and regular approximate symmetries of differential equations with a small parameter.

8.2. Approximations

Discussing approximate symmetries, we first have to define the term approximation. The following terms are used to fix the notation. We assume that $x = (x_1, x_2, \ldots, x_N)$ are the independent coordinates of functions which are analytic in their arguments. Let us also assume that ϵ is a small parameter on which our functions additionally depend. We will denote the involved infinitesimal small functions of order ϵ^{p+1} by $\theta_p(x, \epsilon)$, where $p \leq 0$ is a positive constant. This condition is expressed by $\theta_p(x, \epsilon) = o(\epsilon^p)$. An alternate representation of this condition is

$$\lim_{\epsilon \to 0} \frac{\theta_p(x, \epsilon)}{\epsilon^p} = 0. \tag{8.1}$$

Using this notation, we can state what we mean by an approximation.

Definition: Approximation

Let f and g be analytic functions in x. We define an approximation of order p, $f \sim g$, by the relation

$$f(z, \epsilon) = g(z, \epsilon) + o(\epsilon^p) \tag{8.2}$$

for some fixed value of $p \leq 0$. ○

This definition is the basis of all the calculations we will carry out in the following sections.

8.3. One-Parameter Approximation Group

Following the discussion of Baikov et al. [1991], we define a one-parameter approximation group for a set of vector functions $f_i(x, \epsilon)$, $i = 0, \ldots, p$, with coordinates $f_i^j(x, \epsilon)$, $j = 1, \ldots, N$. The one-parameter family $\mathbf{G}$ of approximate transformations is thus

$$x^{*j} \sim \sum_{i=0}^{p} \delta^i f_i^j(x, \epsilon), \quad j = 1, \ldots, N, \tag{8.3}$$

where $x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$ are the old coordinates and $x^* = (x_1^*, x_2^*, \ldots, x_N^*)$ are the new coordinates, and ϵ and δ are the group parameter and the perturbation parameter, respectively. This transformation satisfies the following conditions:

$$f(x, \epsilon = 0, \delta) \sim x, \tag{8.4}$$

the approximate identity element. Furthermore, it is assumed that the transformation

$$x^* = f(x, \epsilon, \delta) \tag{8.5}$$

is defined for any value of ϵ of a small neighborhood of $\epsilon = 0$ and that this neighborhood allows the relation $f(x, \epsilon, \delta) \sim x$ at $\epsilon = 0$.

The set $\mathcal{G}$ of transformation is called a local one-parameter approximate transformation group if

$$f(f(x, \epsilon, \delta), \gamma, \delta) \sim F(x, \epsilon + \gamma, \delta) \tag{8.6}$$

for all transformations $x = f$.

Example 1

Let us consider an example with $N = 1$. The following two functions are equal in a first-order approximation. The functions f and g depend on the independent variable x and on the two parameters ϵ and δ. ϵ denotes the group parameter and δ the small perturbation parameter.

```
f[x_, ϵ_, δ_] := x + ϵ (1 + δ x + δ ϵ / 2)
```

```
g[x_, ϵ_, δ_] := x + ϵ (1 + δ x) (1 + ϵ δ / 2)
```

The difference between the two functions is given by

```
f[x,ϵ,δ]-g[x,ϵ,δ]//Expand
```

$$-\frac{1}{2} x \delta^2 \epsilon^2$$

which is a function proportional to the square of the small approximation parameter δ. Thus, in first-order approximation, the two functions are equal. The two functions f and g satisfy, in addition, the relation

```
f[g[x,ϵ,δ],ϕ,δ] - f[x,ϵ+ϕ,δ] //Expand
```

$$\frac{1}{2} x \delta^2 \epsilon^2 + x \delta^2 \epsilon \phi + \frac{1}{2} \delta^2 \epsilon^2 \phi + \frac{1}{2} x \delta^3 \epsilon^2 \phi$$

which is the approximate association relation of the related group. The difference is a quadratic function of the group parameters ϵ and ϕ and at least of order two in the small perturbation δ. This example demonstrates the general behavior of an approximate group in first-order approximation. The essential point is that the above relations are satisfied up to the order of approximation; i.e., the approximation order (here, first-order) does not occur in the relations. Note that only higher orders in the perturbation parameter are present.

To determine the group properties of an approximate group transformation, we use similar tools as in the case of Lie point groups. The applied mathematical objects in Lie's theory are the group generator, Lie's equation, infinitesimal transformations, etc. In the following, we will discuss these objects for approximate groups.

8.4. Approximate Group Generator

In close analogy to Lie point groups, we introduce the main tool of symmetry analysis at this point. The group generator or vector field of an approximate group is given by a first-order differential operator of the form

$$\vec{v} = \sum_{i=1}^{p} \xi_i(x, \delta) \frac{\partial}{\partial x_i} \tag{8.7}$$

such that

$$\xi_i(x, \delta) \sim \xi_i^0(x) + \delta \xi_i^1(x) + \cdots + \delta^p \xi_i^p(x), \tag{8.8}$$

where the components of the vector field $(\xi^0, \xi^1, \ldots, \xi^p)$ are given by the expansion coefficients of the transformation

$$\xi_i^\nu = \frac{\partial f_i^\nu(x, \epsilon, \delta)}{\partial \epsilon}\bigg|_{\epsilon=0} \quad \nu = 0, 1, \ldots, p; \quad i = 1, 2, \ldots, N. \tag{8.9}$$

Thus, an approximate vector field is given by

$$\vec{v} \sim (\xi_i^0(x) + \delta \xi_i^1(x) + \cdots + \delta^p \xi_i^p(x)) \frac{\partial}{\partial x_i}. \tag{8.10}$$

The main difference between an ordinary vector field $\vec{v}$ of Lie point symmetries and an approximate vector field is an expansion of the coefficients of the vector field with respect to the perturbation parameter δ. These coefficients follow from a Taylor expansion of the transformation f with respect to the group parameter ϵ and with respect to the perturbation parameter δ.

8.5. The Determining Equations and an Algorithm of Calculation

The purpose of this section is to discuss the connection between the classical Lie theory and the theory of approximate group analysis. We sketch the main steps of the algorithm for calculating approximate symmetries. A detailed presentation and proofs of the statements are contained in the work of Baikov et al. [1989].

Let G be an approximate group of transformations given by equation (8.3). Let us further assume that the order of approximation is $p \geq q$. The approximate equation may be given by

$$\Delta(x, \epsilon) = \Delta_0(x) + \epsilon \Delta_1(x) + \cdots + \epsilon^q \Delta_q(x) = o(\epsilon^q). \tag{8.11}$$

Relation (8.11) is said to be an approximate invariant with respect to G if

$$\Delta(f(x, \delta, \epsilon), \epsilon) = o(\epsilon^q) \tag{8.12}$$

whenever $x = (x_1, \ldots, x_N)$ satisfies equation (8.11).

Assume the approximate vector field $\vec{v}$ is given by equation (8.7). Then, equation (8.11) is approximately invariant under the approximate group G if and only if

$$\mathrm{pr}^{(k)} \vec{v}\, \Delta(x, \epsilon)\,|_{\Delta(x,\epsilon)=o(\epsilon^q)} = o(\epsilon^p). \tag{8.13}$$

This relation is called the determining equation for approximate symmetries. Comparing this expression with Lie's theory, we realize that the original condition of invariance is altered in such a way that the exact vanishing is dropped. In relation (8.13), only an approximate vanishing is needed to derive the determining equations for the infinitesimals. If the determining equation (8.13) is satisfied, we say that equation (8.12) admits the approximate operator $\vec{v}$.

To demonstrate the relations discussed so far, let us consider the simple case with $q = p = 1$. Relations (8.11) and (8.7) simplify to

$$\Delta_0(x) + \epsilon \Delta_1(x) \approx 0 \tag{8.14}$$

and

$$\vec{v} = \vec{v}^0 + \epsilon \vec{v}^1 \equiv \xi_i^0(x) \frac{\partial}{\partial x_i} + \epsilon \xi_i^1(x) \frac{\partial}{\partial x_i}, \tag{8.15}$$

respectively. The determining equation (8.13) then reduces to the relation

$$\mathrm{pr}^{(k)} \left(\vec{v}^0 + \epsilon \vec{v}^1 \right) (\Delta_0(x) + \epsilon \Delta_1(x))\,|_{\Delta_0(x) + \epsilon \Delta_1(x) = 0} = o(\epsilon). \tag{8.16}$$

This relation contains an algorithm for the calculation of first-order approximate symmetries. The algorithm is based on a theorem stated in Baikov et al. [1989].

Theorem: First-order approximations

In the first-order approximation, the determining equations for approximate symmetries follow from the system of relations

$$\mathrm{pr}^{(k)}\,\vec{v}^{\,0}\,\Delta_0(x) \;=\; \lambda(x)\,\Delta_0(x), \tag{8.17}$$

and

$$\mathrm{pr}^{(k)}\,\vec{v}^{\,1}\,\Delta_0(x) \;+\; \mathrm{pr}^{(k)}\,\vec{v}^{\,0}\,\Delta_1(x) \;=\; \lambda(x)\,\Delta_1(x). \tag{8.18}$$

The auxiliary factor $\lambda(x)$ is determined from (8.17) and afterward substituted into (8.18), where (8.18) itself must hold for the solution x of the unperturbed equation $\Delta_0(x) = 0$. ○

The above theorem provides an algorithm for the calculation of first-order approximate symmetries. The algorithm consists of the following four steps:

1. Find the exact symmetries generated by $\vec{v}^{\,0}$ of the unperturbed equation. This step is equivalent to the classical theory of Lie. In equations, we have

$$\mathrm{pr}^{(k)}\,\vec{v}^{\,0}\,\Delta_0(x)\,|_{\Delta_0=0} = 0. \tag{8.19}$$

2. If we know the symmetries in the zero approximation $\vec{v}^{\,0}$, we can use them to calculate the auxiliary function $\lambda(x)$ if the perturbation $\epsilon\Delta_1(x)$ is given. The deficiency $\lambda = H$ follows by

$$H \;\approx\; \frac{1}{\epsilon}\,\mathrm{pr}^{(k)}\,\vec{v}^{\,0}(\Delta_0(x) \;+\; \epsilon\,\Delta_1(x))\,|_{\Delta_0+\epsilon\Delta_1=0}\,. \tag{8.20}$$

3. The symmetries of the first-order approximation then follow from the relation

$$\mathrm{pr}^{(k)}\,\vec{v}^{\,1}(\Delta_0(x))\,\big|_{\Delta_0=0} \;+\; H \;=\; 0. \tag{8.21}$$

4. Check the consistency of the approximation at the end.

We remark that in the approximate group analysis of differential equations, the prolongation formulas are the same as in the classical Lie algorithm. Thus, it is straightforward to use the functions of *MathLie* to derive approximate symmetries. The package *MathLie* offers the function ApproximateSymmetries[] to carry out the calculations in a single step. The following examples will demonstrate the application of the function.

8.6. Examples

The following two examples are taken from Baikov et al. [1989]. They serve to illustrate the calculation of approximate symmetries in connection with computer algebra. Our results are identical with those published by Baikov and co-workers.

8.6.1 Isentropic Liquid

Let us consider the problem of a liquid in a pipe (Baikov et al. [1989]). The system of equations of motion for the fluid density ρ and the velocity field v in a one-dimensional space is given by

$$\rho_t + (\rho v)_x = 0 \tag{8.22}$$

and

$$\rho v_t + \rho v v_x + p_x - \epsilon \rho v = 0, \tag{8.23}$$

where p is the pressure of the liquid and ϵ is the hydraulic-friction coefficient. In Lagrange coordinates t and $q = \int \rho \, dx$, the system becomes

```
eqOfMotion = {∂t (1/ρ[q, t]) - ∂q v[q, t] == 0,
   ∂t v[q, t] + ∂q p[q, t] - ϵ v[q, t] == 0};
eqOfMotion // LTF
```

$$-v_q - \frac{\rho_t}{\rho^2} == 0$$

$$-v\epsilon + p_q + v_t == 0$$

hence, for

```
rule1 = ρ → Function[{q, t}, 1 / u[q, t]];
```

we obtain the equations

```
eqation = eqOfMotion /. rule1; eqation // LTF
```

$$u_t - v_q == 0$$

$$-v\epsilon + p_q + v_t == 0$$

Differentiating the first equation with respect to t and the second with respect to q, and replacing the spatial derivative of v by the temporal derivative of u, we end up with the relation

```
equation = ((∂t eqation[[1, 1]] + ∂q eqation[[2, 1]]) /.
     Solve[eqation[[1]], v^(1,0) [q, t]] )[[1]] == 0;
equation // LTF
```

$$-\epsilon\, u_t + p_{q,q} + u_{t,t} == 0$$

If we now represent the pressure by the expression

```
rule2 = p → Function[{q, t}, u[q, t]^(σ+1) / (σ + 1)];
```

with σ a constant, we can represent the equation as

```
equation = equation /. rule2 // Simplify; equation // LTF
```

$$u^{-1+\sigma}\,\sigma\,u_q^2 - \epsilon\, u_t + u^{\sigma}\, u_{q,q} + u_{t,t} == 0$$

This equation depends on two parameters σ and ϵ denoting the isentropic exponent and the perturbation parameter, respectively. In the following, we consider the hydraulic-friction coefficient ϵ as a small quantity. Thus, we can discuss the symmetries of the equation under the condition that ϵ creates a disturbance of the original fluid equations. The approximate symmetries of the equation follow by applying the function ApproximateSymmetries[] to the derived equation:

```
infinitesimals =
 ApproximateSymmetries[equation, {u}, {q, t}, {σ, ϵ}, ϵ];
infinitesimals // LTF
```

$$\phi_1 == \frac{4\,k4\,u}{4+\sigma} + \frac{k4\,u\,\sigma}{4+\sigma} + \epsilon\left(-\frac{k4\,t\,u\,\sigma}{4+\sigma} + \frac{u\,(2\,k3\,t + k8\,(4+\sigma))}{4+\sigma}\right)$$

$$\xi_1 == k2 + k3\,q + \epsilon\left(k5 + k7\,q + \frac{k8\,q\,\sigma}{2}\right)$$

$$\xi_2 == k1 + t\left(k3 - \frac{k4\,\sigma}{2}\right) + \epsilon\left(k6 + k7\,t + \frac{t^2\,\sigma\,(-2\,k3 + k4\,\sigma)}{4\,(4+\sigma)}\right)$$

The result is a finite eight-dimensional symmetry group depending on the small parameter ϵ and the parameter σ. It was essential in the above calculation that both parameters σ and ϵ are given in the parameters list. The last argument of the function ApproximateSymmetries[] contains only the name of the perturbation parameter ϵ. In this way, we are able to select one of the parameters as a small quantity. The coefficients of the eight different vector fields are accessible by

```
cases = {{xi[1][q, t, u], xi[2][q, t, u]},
     {phi[1][q, t, u]}} /. infinitesimals /.
   (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7, k8} → #])&,
     Permutations[{1, 0, 0, 0, 0, 0, 0, 0}]]) //
  Simplify
```

$$\{\{\{0, 1\}, \{0\}\}, \{\{1, 0\}, \{0\}\}, \{\{q, t - \frac{t^2 \epsilon \sigma}{2 (4 + \sigma)}\}, \{\frac{2 t u \epsilon}{4 + \sigma}\}\},$$
$$\{\{0, \frac{t \sigma (-8 + (-2 + t \epsilon) \sigma)}{4 (4 + \sigma)}\}, \{\frac{u (4 + \sigma - t \epsilon \sigma)}{4 + \sigma}\}\}, \{\{\epsilon, 0\}, \{0\}\},$$
$$\{\{0, \epsilon\}, \{0\}\}, \{\{q \epsilon, t \epsilon\}, \{0\}\}, \{\{\frac{q \epsilon \sigma}{2}, 0\}, \{u \epsilon\}\}\}$$

We select one sublist from the above list to reduce the original equation and find a similarity representation. Let us examine, for example, the fourth subgroup which contains the approximation parameter ϵ. The reduction follows by

```
red1 = LieReduction[equation,
   {u}, {q, t}, cases[[4, 1]], cases[[4, 2]]] // Simplify;
red1 /. zeta1 → ζ1 // Flatten // LTF
```

$$q - \zeta_1 == 0$$
$$t^{2/\sigma}\, u\, (-8 - 2\sigma + t \epsilon \sigma)^{2/\sigma} - F_1 == 0$$
$$(-8 - 2\sigma + t \epsilon \sigma)^{2/\sigma}$$
$$(4 (64 + 64\sigma - 4 (-5 + 2 t^2 \epsilon^2) \sigma^2 + (2 - 2 t^2 \epsilon^2 + t^3 \epsilon^3) \sigma^3) F_1^2 +$$
$$\sigma^3 F_1^{\sigma} (F_1)^2_{\zeta_1} + \sigma^2 F_1^{1+\sigma} (F_1)_{\zeta_1, \zeta_1}) ==$$
$$0$$

The result shows that the similarity variable $\zeta_1 = q$ and the similarity representation is given by $u = t^{-2/\sigma} (-8 - 2\sigma + t \epsilon \sigma)^{-2/\sigma} F_1 (q)$. This expression depends on ϵ in a certain power of ϵ. Since our approximation order in ϵ is one, we have to expand the similarity solution in ϵ to first order. The expansion is carried out by

```
v1 =
 Series[u /. Solve[red1[[2]], u], {ε, 0, 1}] // Normal // Simplify
```

$$\{\frac{t^{-2/\sigma} (-8 - 2\sigma)^{-2/\sigma} (4 + t \epsilon + \sigma) F1[q]}{4 + \sigma}\}$$

```
rule3 = u → Function@@{{q, t}, v1[[1]]}
```

$$u \to \text{Function}[\{q, t\}, \frac{t^{-2/\sigma} (-8 - 2\sigma)^{-2/\sigma} (4 + t \epsilon + \sigma) F1[q]}{4 + \sigma}]$$

On the other hand, the arbitrary function $F1$ has to satisfy a second-order ODE. Examining this equation, we realize that the second-order ODE contains coefficients depending on t. However, this dependence will directly lead to inconsistent results. Since we are looking for solutions linear in ϵ, we have to eliminate the time-dependent terms. The coefficients containing t contribute terms in ϵ of the order two or higher. A consistent similarity reduction is only possible if we eliminate those terms by choosing $\epsilon = 0$. The second-order ODE reduces thus to

```
redh = red1[[3]] /. ε → 0; redh /. zeta1 → ζ1 // LTF
```

$$(-8-2\sigma)^{2/\sigma}\,(4\,(64+64\sigma+20\sigma^2+2\sigma^3)\,F_1^2+\sigma^3\,F_1^{\sigma}\,(F_1)^2_{\zeta_1}+\sigma^2\,F_1^{1+\sigma}\,(F_1)_{\zeta_1,\zeta_1})==0$$

free of ϵ and t. The solution of this approximation equation follows by

```
sol1 = DSolve[redh, F1, zeta1]
```

$$\left\{\left\{F1\to\left(\text{InverseFunction}\left[\right.\right.\right.$$

$$C[2]-(\#1^{-1-\sigma}\,(C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}))\Big/$$

$$\left(8\,\sigma\,(4+\sigma)^2\sqrt{\left(\frac{1}{\sigma^2}\,(\#1^{-2\sigma}\,(C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}))\right)}\right)-$$

$$\left((-2-\sigma)\,C[1]\,\text{Hypergeometric2F1}\left[\frac{1}{-2-\sigma},\frac{1}{2},1+\frac{1}{-2-\sigma},-\frac{(-256-128\sigma-16\sigma^2)\,\#1^{2+\sigma}}{C[1]}\right]\#1^{-1+\frac{1}{4}(-2-\sigma)-\sigma+\frac{2+\sigma}{4}}\right.$$

$$\sqrt{C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}}$$

$$\left.\sqrt{1+\frac{(-256-128\sigma-16\sigma^2)\,\#1^{2+\sigma}}{C[1]}}\right)\Big/$$

$$\left(8\,\sigma\,(2+\sigma)\,(4+\sigma)^2\sqrt{\left(\frac{1}{\sigma^2}\,(\#1^{-2\sigma}\,(C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}))\right)}\right.$$

$$\left.\sqrt{C[1]+(-256-128\sigma-16\sigma^2)\,\#1^{2+\sigma}}\right)\&\left.\right][$$

$$\left.\#1]\&\right)\Big\},$$

$$\left\{F1\to\left(\text{InverseFunction}\left[\right.\right.\right.$$

$$C[2]+(\#1^{-1-\sigma}\,(C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}))\Big/$$

$$\left(8\,\sigma\,(4+\sigma)^2\sqrt{\left(\frac{1}{\sigma^2}\,(\#1^{-2\sigma}\,(C[1]-256\,\#1^{2+\sigma}-128\,\sigma\,\#1^{2+\sigma}-16\,\sigma^2\,\#1^{2+\sigma}))\right)}\right)+$$

$$\left((-2-\sigma)\,C[1]\,\text{Hypergeometric2F1}\left[\frac{1}{-2-\sigma},\right.\right.$$

$$\left.\left.\frac{1}{2},1+\frac{1}{-2-\sigma},-\frac{(-256-128\sigma-16\sigma^2)\,\#1^{2+\sigma}}{C[1]}\right]\right.$$

$$\#1^{-1+\frac{1}{4}(-2-\sigma)-\sigma+\frac{2+\sigma}{4}}$$

$$\sqrt{C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma}}$$

$$\sqrt{1 + \frac{(-256 - 128\,\sigma - 16\,\sigma^2)\,\#1^{2+\sigma}}{C[1]}}\Bigg) \Big/ \Big(8\,\sigma\,(2+\sigma)\,(4+\sigma)^2$$

$$\sqrt{\Big(\frac{1}{\sigma^2}\,(\#1^{-2\,\sigma}\,(C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma}))\Big)}$$

$$\sqrt{C[1] + (-256 - 128\,\sigma - 16\,\sigma^2)\,\#1^{2+\sigma}}\Big)\&\Big][$$

$$\#1]\&\Big)\Big\}\Big\}$$

The results are two solutions for function F_1. Both representations contain special functions of the hypergeometric type ${}_2F_1$ depending on the isentropic exponent σ. The representation of the solution in the original variables q and t follows by substituting the results into the approximated similarity solution. We choose here the first solution for the representation:

```
solution = u[q, t] /. rule3 /. sol1[[1]]
```

$$\frac{1}{4+\sigma}\Bigg(t^{-2/\sigma}\,(-8-2\,\sigma)^{-2/\sigma}\,(4+t\,\epsilon+\sigma)\,\text{InverseFunction}\Big[$$

$$C[2] - (\#1^{-1-\sigma}\,(C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma})) \Big/$$

$$\Big(8\,\sigma\,(4+\sigma)^2\,\sqrt{\Big(\frac{1}{\sigma^2}$$

$$(\#1^{-2\,\sigma}\,(C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma}))\Big)\Big) -$$

$$\Bigg((-2-\sigma)\,C[1]\,\text{Hypergeometric2F1}\Big[\frac{1}{-2-\sigma},\ \frac{1}{2},\ 1+\frac{1}{-2-\sigma},$$

$$-\frac{(-256 - 128\,\sigma - 16\,\sigma^2)\,\#1^{2+\sigma}}{C[1]}\Big]\,\#1^{-1+\frac{1}{4}(-2-\sigma)-\sigma+\frac{2+\sigma}{4}}$$

$$\sqrt{C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma}}$$

$$\sqrt{1 + \frac{(-256 - 128\,\sigma - 16\,\sigma^2)\,\#1^{2+\sigma}}{C[1]}}\Bigg) \Big/$$

$$\Big(8\,\sigma\,(2+\sigma)\,(4+\sigma)^2$$

$$\sqrt{\Big(\frac{1}{\sigma^2}\,(\#1^{-2\,\sigma}\,(C[1] - 256\,\#1^{2+\sigma} - 128\,\sigma\,\#1^{2+\sigma} - 16\,\sigma^2\,\#1^{2+\sigma}))\Big)}$$

$$\sqrt{C[1] + (-256 - 128\,\sigma - 16\,\sigma^2)\,\#1^{2+\sigma}}\Big)\&\Big][$$

$$q]\Bigg)$$

$C[1]$ and $C[2]$ are constants of integration and InverseFunction[] represents the inverse of the function f.

The original equation of the isentropic fluid in Lagrange coordinates can be examined either for specific values of the isentropic parameter σ or for different models for the pressure. Let us discuss two cases for a specific value of σ. The change of pressure in the model is left to the reader as an exercise. The following calculation of the approximate symmetries assume that $\sigma = -4/3$:

```
infinitesimals = ApproximateSymmetries[equation /. σ → -4 / 3,
  {u}, {q, t}, {ϵ}, ϵ, SubstitutionRules → {∂t,t u[q, t]}];
infinitesimals // LTF
```

$$\phi_1 == \frac{3\,k4\,u}{2} - \frac{3\,k6\,u}{2} - 3\,k7\,q\,u + \left(-\frac{3\,k2\,u}{2} + \frac{3\,k9\,u}{2} - 3\,k3\,q\,u + \frac{3\,k4\,t\,u}{4}\right)\epsilon$$

$$\xi_1 == k5 + q\,(k6 + k7\,q) + (k10 + q\,(k2 + k3\,q))\,\epsilon$$

$$\xi_2 == k1 + k4\,t + \left(k8 + k9\,t + \frac{k4\,t^2}{4}\right)\epsilon$$

The result of the calculation is a 10-dimensional finite symmetry group in the first-order approximation. The coefficients of the generating vector fields are

```
cases2 = {{xi[1][q, t, u], xi[2][q, t, u]},
    {phi[1][q, t, u]}} /. infinitesimals /.
  (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7, k8, k9, k10} → #]) &,
    Permutations[{1, 0, 0, 0, 0, 0, 0, 0, 0, 0}]])
```

$$\left\{\{\{0, 1\}, \{0\}\}, \left\{\{q\epsilon, 0\}, \left\{-\frac{3\,u\,\epsilon}{2}\right\}\right\},\right.$$
$$\{\{q^2\,\epsilon, 0\}, \{-3\,q\,u\,\epsilon\}\}, \left\{\left\{0, t + \frac{t^2\,\epsilon}{4}\right\}, \left\{\frac{3\,u}{2} + \frac{3\,t\,u\,\epsilon}{4}\right\}\right\},$$
$$\{\{1, 0\}, \{0\}\}, \left\{\{q, 0\}, \left\{-\frac{3\,u}{2}\right\}\right\}, \{\{q^2, 0\}, \{-3\,q\,u\}\},$$
$$\left.\{\{0, \epsilon\}, \{0\}\}, \left\{\{0, t\epsilon\}, \left\{\frac{3\,u\,\epsilon}{2}\right\}\right\}, \{\{\epsilon, 0\}, \{0\}\}\right\}$$

A specific reduction for the third vector field with $\vec{v}_3 = \epsilon q\,\partial_q - 3\,\epsilon q\,u\,\partial_u$ follows by

```
red2 = LieReduction[equation /. σ → -4 / 3,
   {u}, {q, t}, cases2[[3, 1]], cases2[[3, 2]]] // Simplify;
red2 /. zeta1 → ζ1 // Flatten // LTF
```

$$t - \zeta_1 == 0$$
$$q^3\,u - F1_{\zeta_1} == 0$$
$$-\epsilon\,F1_{\zeta_1} + F1_{\zeta_1,\zeta_1} == 0$$

The similarity representation is given by the similarity solution $u = F_1(t)/q^3$ with similarity variable $\zeta_1 = t$. The auxiliary function F_1 has to satisfy a second-order ODE. We realize that the similarity solution does not depend on ϵ. However, ϵ occurs in the determining equation of F_1. The solution of the equation for F_1 is given by

```
solh = DSolve[red2[[3]], F1, zeta1]
```

$$\left\{\left\{\text{F1} \to \left(\frac{\text{E}^{\epsilon\,\#1}\,\text{C[1]}}{\epsilon} + \text{C[2]}\&\right)\right\}\right\}$$

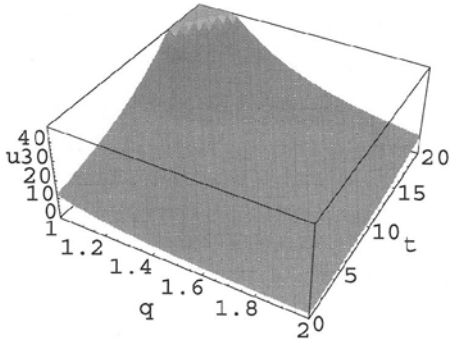
This function actually does not depend linearly on ϵ and, thus, it does not fit into the approximation scheme of a first-order approximation. However, it is a solution of the reduced equation. The representation of this expression in Lagrange coordinates reads

```
solution2 = Flatten[Solve[red2[[2]] /. solh, u]]
```

$$\left\{\text{u} \to \frac{\text{E}^{\text{t}\,\epsilon}\,\text{C[1]} + \epsilon\,\text{C[2]}}{\text{q}^3\,\epsilon}\right\}$$

A graphical representation of the solution for specific values of the parameters is given as follows:

```
Plot3D[Evaluate[u /. solution2 /. {C[1] → 1, C[2] → 2, ϵ → .1}],
 {q, 1, 2}, {t, 0, 20}, AxesLabel → {"q", "t", "u"},
 Mesh → False, PlotPoints → 35]
```



Another case discussed by Baikov et al. [1989] is the isentropic motion with $\sigma = -4$. The equation of motion for this case reads

```
eq = equation /. σ → -4; eq // LTF
```

$$-\frac{4\,u_q^2}{u^5} - \epsilon\,u_t + \frac{u_{q,q}}{u^4} + u_{t,t} == 0$$

The first-order approximate symmetries of this equation follows by

```
infinitesimals = ApproximateSymmetries[eq, {u},
  {q, t}, {ϵ}, ϵ, SubstitutionRules → {∂x,x u[q, t]}];
infinitesimals // LTF
```

$$\phi_1 == -\frac{k4\,u}{2} + \frac{1}{2}\,(-k2 + k6 + 2\,k7\,t)\,u\,\epsilon$$

$$\xi_1 == k3 + k4\,q + (k8 + k2\,q)\,\epsilon$$

$$\xi_2 == k1 + (k5 + t\,(k6 + k7\,t))\,\epsilon$$

representing an eight-dimensional finite symmetry group. Again, we can use the infinitesimals to derive analytic solutions for the isentropic model. The generating vector fields of the subgroups read

```
Map[(Fold[Plus, 0, Map[Fold[NonCommutativeMultiply, 1, #]&,
        Transpose[{Flatten[#], {"∂q", "∂t", "∂u"}}]] /.
      _ ** 0 ** _ → 0] /.
    1 ** a_ ** b_ → a ** b)&,
  {{xi[1][q, t, u], xi[2][q, t, u]},
    {phi[1][q, t, u]}} /. infinitesimals /.
   (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7, k8} → #])&,
     Permutations[{1, 0, 0, 0, 0, 0, 0, 0}]])] //
 TableForm
```

$$1 ** \partial_t$$

$$(q\,\epsilon) ** \partial_q + (-\tfrac{u\,\epsilon}{2}) ** \partial_u$$

$$1 ** \partial_q$$

$$q ** \partial_q + (-\tfrac{u}{2}) ** \partial_u$$

$$\epsilon ** \partial_t$$

$$(t\,\epsilon) ** \partial_t + \tfrac{u\,\epsilon}{2} ** \partial_u$$

$$(t^2\,\epsilon) ** \partial_t + (t\,u\,\epsilon) ** \partial_u$$

$$\epsilon ** \partial_q$$

Here, we used the function NonCommutativeMultiply[] (**) to keep the ordering of the operators in the representation of the vector fields. The coefficients of these differential equations are

```
cases3 = {{xi[1][q, t, u], xi[2][q, t, u]},
    {phi[1][q, t, u]}} /. infinitesimals /.
  (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7, k8} → #])&,
    Permutations[{1, 0, 0, 0, 0, 0, 0, 0}]])
```

$$\left\{\{\{0,\ 1\},\ \{0\}\},\ \left\{\{q\epsilon,\ 0\},\ \left\{-\frac{u\epsilon}{2}\right\}\right\},\ \{\{1,\ 0\},\ \{0\}\},\right.$$

$$\left.\{\{q,\ 0\},\ \left\{-\frac{u}{2}\right\}\right\},\ \{\{0,\ \epsilon\},\ \{0\}\},\ \left\{\{0,\ t\epsilon\},\ \left\{\frac{u\epsilon}{2}\right\}\right\},\right.$$

$$\left.\{\{0,\ t^2\epsilon\},\ \{tu\epsilon\}\},\ \{\{\epsilon,\ 0\},\ \{0\}\}\right\}$$

One of the possible reductions can be calculated by combining two sub-groups. For the present calculation, we combine the third and seventh sub-group:

```
red3 = LieReduction[eq, {u}, {q, t}, cases3[[7, 1]] + cases3[[3, 1]],
   cases3[[7, 2]] + cases3[[3, 2]]] // Simplify;
red3 /. zeta1 → ζ1 // Flatten // LTF
```

$$\frac{1}{t} + q\epsilon - \zeta_1 == 0$$

$$-\frac{u}{t} - F_1 == 0$$

$$t^3\epsilon F_1^6 + 4\epsilon^2 (F_1)^2_{\zeta_1} - \epsilon^2 F_1 (F_1)_{\zeta_1,\zeta_1} - F_1^5 (t^2\epsilon (F_1)_{\zeta_1} + (F_1)_{\zeta_1,\zeta_1})$$
$$== 0$$

Solving for the unknown field u, we find an explicit similarity representation of the solution:

```
Solve[-u/t == F1[1/t + qϵ], u]
```

$$\left\{\left\{u \to -t\, F1\left[\frac{1}{t} + q\epsilon\right]\right\}\right\}$$

The resulting similarity representation is given by $u = -t\, F_1(1/t + q\,\epsilon)$, where F_1 has to satisfy a second-order ODE depending on t and ϵ. We eliminate this dependence by choosing $\epsilon = 0$ in the determining equation of F_1. The solution of the resulting equation follows with

```
solh = DSolve[red2[[3]] /. ϵ → 0 , F1, zeta1]
```

```
{{F1 → (C[1] + C[2] #1&)}}
```

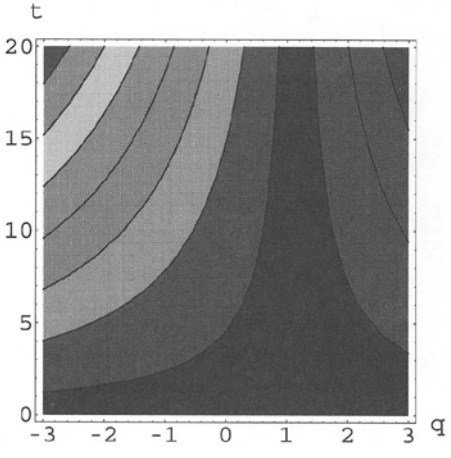
representing a linear function in *zeta1*. The solution in Lagrange coordinates t and q reads

```
solution3 = Solve[Flatten[red3[[2]] /. solh], u] // Flatten
```

```
{u → -t C[1] - C[2] - q t ϵ C[2]}
```

This solution depends linearly on ϵ and thus is consistent with the approximation order of the procedure. The choice of numerical values for the integration constants $C[1]$, $C[2]$ and the approximation parameter ϵ allows us to outline this solution in a contour plot.

```
ContourPlot[
 Evaluate[u /. solution3 /. {C[1] → 1, C[2] → -1, ϵ → .9}],
 {q, -3, 3}, {t, 0, 20}, AxesLabel → {"q", "t"},
 ColorFunction → Hue, Axes → True]
```



8.6.2 Perturbed Korteweg-de Vries Equation

One of the frequently discussed equations in soliton theory is the Korteweg-de Vries equation (KdV)

$$u_t + u\,u_x + u_{x,x,x} = 0. \tag{8.24}$$

The KdV equation is one of the rare equations which is solvable and possesses an infinite number of integrals of motion. The equation first derived by Korteweg and de Vries in 1895 describes shallow water waves in narrow channels. Korteweg and de Vries showed that periodic solutions, which they called cnoidal waves, could be found in closed form and without further approximation. Our interest here is the approximation aspect of the fluid dynamics. It is well known that the KdV equation is an approximated equation in a certain limit incorporating the effects of dispersion and surface tension which stabilizes a wave. If now we incorporate terms in the equation which are actually present in nature but are dropped in equation (8.24), we may gain information on the influence of such terms. For example, let us extend the

KdV equation by a dispersive term u_{xx} that Burgers [1948] used in his turbulent theory. In the following, we examine the equation

```
KdVepsilon = ∂t u[x, t] +
   u[x, t] ∂x u[x, t] + ∂{x,3} u[x, t] + ϵ ∂{x,2} u[x, t] == 0;
KdVepsilon // LTF
```

$$u_t + u\,u_x + \epsilon\,u_{x,x} + u_{x,x,x} == 0$$

where ϵ is a small parameter measuring the influence of second-order dispersion. For short we call this equation the KdV-ϵ equation. The approximate symmetries of this equation follow by

```
KdVinfinitesimals = ApproximateSymmetries[KdVepsilon,
  {u}, {x, t}, {ϵ}, ϵ, SubstitutionRules → {∂t u[x, t]}];
KdVinfinitesimals // LTF
```

$$\phi_1 == k2 + (k5 - 2\,k7\,u)\,\epsilon$$

$$\xi_1 == k3 + k2\,t + (k6 + k5\,t + k7\,x)\,\epsilon$$

$$\xi_2 == k1 + (k4 + 3\,k7\,t)\,\epsilon$$

representing a seven-dimensional approximate symmetry group. The generating vector fields for this model read

```
Map[(Fold[Plus, 0, Map[Fold[NonCommutativeMultiply, 1, #]&,
        Transpose[{Flatten[#], {"∂x", "∂t", "∂u"}}]] /.
       _ ** 0 ** _ → 0] /.
     1 ** a_ ** b_ → a ** b)&,
  {{xi[1][x, t, u], xi[2][x, t, u]},
     {phi[1][x, t, u]}} /. KdVinfinitesimals /.
   (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7} → #])&,
     Permutations[{1, 0, 0, 0, 0, 0, 0}]])] //
 TableForm
```

$$1 ** \partial_t$$

$$1 ** \partial_u + t ** \partial_x$$

$$1 ** \partial_x$$

$$\epsilon ** \partial_t$$

$$\epsilon ** \partial_u + (t\,\epsilon) ** \partial_x$$

$$\epsilon ** \partial_x$$

$$(3\,t\,\epsilon) ** \partial_t + (-2\,u\,\epsilon) ** \partial_u + (x\,\epsilon) ** \partial_x$$

The coefficients of these vector fields are given by

```
casesKdV = {{xi[1][x, t, u], xi[2][x, t, u]},
    {phi[1][x, t, u]}} /. KdVinfinitesimals /.
  (Map[(Thread[{k1, k2, k3, k4, k5, k6, k7} → #])&,
    Permutations[{1, 0, 0, 0, 0, 0, 0}]])
```

```
{{{0, 1}, {0}}, {{t, 0}, {1}}, {{1, 0}, {0}}, {{0, ϵ}, {0}},
 {{t ϵ, 0}, {ϵ}}, {{ϵ, 0}, {0}}, {{x ϵ, 3 t ϵ}, {-2 u ϵ}}}
```

The group of seven vector fields contains symmetries like translations and scalings depending on the perturbation parameter ϵ. Let us choose a linear combination of three subgroups to reduce the original KdV-ϵ equation. The following line creates the similarity reduction in connection with the first, third, and fifth sub-groups.

```
redKdV = LieReduction[KdVepsilon, {u}, {x, t}, casesKdV[[1, 1]] +
    c casesKdV[[3, 1]] + casesKdV[[5, 1]], casesKdV[[1, 2]] +
    c casesKdV[[3, 2]] + casesKdV[[5, 2]]] // Simplify;
redKdV /. zeta1 → ζ1 // Flatten // LTF
```

$$-c\,t + x - \frac{t^2\,\epsilon}{2} - \zeta_1 == 0$$

$$-c + u - t\,\epsilon - F_1 == 0$$

$$\epsilon + F_1\,(F_1)_{\zeta_1} + \epsilon\,(F_1)_{\zeta_1,\zeta_1} + (F_1)_{\zeta_1,\zeta_1,\zeta_1} == 0$$

The result is a similarity representation of the solution depending linearly on ϵ. The similarity variable $\zeta_1 = x - ct - \epsilon t^2/2$ also depends linearly on ϵ. The determining equation of the similarity function F_1 also shows a linear dependence on ϵ. All these linear dependencies on ϵ will result into a non-linear dependence on ϵ of the solution. Thus, let us examine the determining equation without the ϵ terms. The reduced KdV-ϵ equation is integrated by two quadratures to the form

```
b1 = Integrate[∂zeta1 F1[zeta1]
    ( Integrate[redKdV[[3, 1, 1]] /. ϵ → 0 // Expand, zeta1]),
  zeta1]
```

$$\frac{F1[zeta1]^3}{6} + \frac{1}{2}\,F1'[zeta1]^2$$

This result can be treated in two different ways. First, let us assume that the expression equals a constant K. For this case, we find by DSolve[]

```
case1 = DSolve[b1 == K, F1, zeta1]
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.
```

```
Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.

Solve::tdep :
 The equations appear to involve transcendental functions
   of the variables in an essentially non-algebraic way.

General::stop : "Further output of \!\(Solve::" tdep
 \) will be suppressed during this calculation.
```

$$\left\{\text{Solve}\left[C[1] - \frac{1}{\sqrt{-F1^3 + 6K}}\left(2\,I\,2^{1/3}\,3^{7/12}\right.\right.\right.$$
$$\sqrt{(-1)^{5/6}\left(-1 + \frac{F1}{6^{1/3}K^{1/3}}\right)}\sqrt{1 + \frac{F1^2}{6^{2/3}K^{2/3}} + \frac{F1}{6^{1/3}K^{1/3}}}\,K^{1/3}$$
$$\left.\left.\left.\text{EllipticF}\left[\text{ArcSin}\left[\frac{\sqrt{-(-1)^{5/6} - \frac{I\,F1}{6^{1/3}K^{1/3}}}}{3^{1/4}}\right], (-1)^{1/3}\right]\right) ==\right.$$
$$\#1,$$
$$F1\Big],$$
$$\text{Solve}\left[C[1] + \frac{1}{\sqrt{-F1^3 + 6K}}\left(2\,I\,2^{1/3}\,3^{7/12}\right.\right.$$
$$\sqrt{(-1)^{5/6}\left(-1 + \frac{F1}{6^{1/3}K^{1/3}}\right)}\sqrt{1 + \frac{F1^2}{6^{2/3}K^{2/3}} + \frac{F1}{6^{1/3}K^{1/3}}}\,K^{1/3}$$
$$\left.\left.\text{EllipticF}\left[\text{ArcSin}\left[\frac{\sqrt{-(-1)^{5/6} - \frac{I\,F1}{6^{1/3}K^{1/3}}}}{3^{1/4}}\right], (-1)^{1/3}\right]\right) ==$$
$$\#1,$$
$$F1\Big]\Big\}$$

that the solution is given by elliptic functions. Since Solve[] is unable to invert the expression, we only get the solution in an implicit form. The second case we can examine is when $K = 0$. In this case, the solution is given by

```
case2 = DSolve[b1 == 0, F1, zeta1]
```

$$\left\{\left\{F1 \to \left(-\frac{12}{(\#1 - C[1])^2}\&\right)\right\}\right\}$$

The similarity representation of the solution follows from this relation by

```
sol = Solve[redKdV[[2]] /. case2[[1]], u]
```

$$\left\{\left\{u \to c + t\,\epsilon - \frac{12}{\left(-c\,t + x - \frac{t^2\,\epsilon}{2} - C[1]\right)^2}\right\}\right\}$$

Since the solution should depend linearly on ϵ, we have to expand the result up to first order in ϵ around $\epsilon = 0$:

```
solution = Series[u /. sol, {ϵ, 0, 1}]
```

$$\left\{\left(c - \frac{12}{(-c\,t + x - C[1])^2}\right) + \left(t - \frac{12\,t^2}{(-c\,t + x - C[1])^3}\right)\epsilon + O[\epsilon]^2\right\}$$

This expression gives us an approximate solution of the KdV-ϵ equation. We can check relation (8.11) by inserting the above solution into the original equation. In the first step, we convert the solution into a pure function representation:

```
srule = u → Apply[Function, {{x, t}, solution[[1]] // Normal}]
```

$$u \to \text{Function}\left[\{x, t\}, c + \epsilon\left(t - \frac{12\,t^2}{(-c\,t + x - C[1])^3}\right) - \frac{12}{(-c\,t + x - C[1])^2}\right]$$

Inserting the solution into the original equation demonstrates that the equation is satisfied in first-order approximation:

```
KdVepsilon /. srule // Simplify
```

$$\epsilon\left(1 + \frac{432\,t^4\,\epsilon}{(c\,t - x + C[1])^7} + \frac{144\,t^2\,\epsilon}{(c\,t - x + C[1])^5} - \frac{72}{(c\,t - x + C[1])^4} + \frac{36\,t^3\,\epsilon}{(c\,t - x + C[1])^4}\right) == 0$$

The two examples presented are a small representation of the huge amount of equations depending on a small perturbation parameter. We note that the derivation of an approximate solution is not unique. As the reader has noticed in the solution step, there is a great flexibility in choosing the stage of approximation. However, at the end of the calculation, we have to satisfy relation (8.11) defining the order of approximation.

9

Generalized Symmetries

9.1. Introduction

The previous sections discussed the classical method, the non-classical method, and some extensions of Lie's theory. We demonstrated the classification and solution of some types of ordinary as well as partial differential equations. Although very general, these methods mainly consider the geometric aspects of the transformations related to the equations. Essentially, we discussed invariance under point transformations and demonstrated that some kind of contact transformations are related to point transformations in case of first-order partial differential equations. However, it turns out that much wider classes of transformations leave differential equations invariant, including those considered by Lie.

Emmi Noether [1918] and Felix Klein [1918] demonstrated that a system of differential equations derivable from a variational principle allows a much greater set of transformations under which a given system of differential equations is invariant. They significantly extended the application of symmetry group methods by including derivatives of the dependent variables in the transformations. Moreover, they were able to offer a regular procedure to construct the related conservation laws which are based on the investigation of the invariance properties of the variational integral under the action of transformation groups. With the publication of her theorem, Emmi Noether demonstrated that Lie point symmetries cannot provide the total

number of symmetries for a given system of differential equations. Consequently, the required integrals of motion and the solutions are not available for many systems.

In recent years, the symmetry methods have become more attractive, especially in the field of non-linear dynamics. In this chapter, we will consider a generalized method of Lie's classical theory to determine the symmetries for non-linear PDEs and ODEs. We will demonstrate that the method is applicable to a large number of different models. Let us first outline the mathematical theory to generate a firm basis on which *MathLie* is grounded.

9.2. Elements of Generalized Symmetries

The setting of our program *MathLie* is an entirely general one. Its algorithm is well known and described, for example. in the books by Olver [1986] or Bluman and Kumei [1989]. We consider the general case of a non-linear system of differential equations for an arbitrary number q of unknown functions u^α which may depend on p independent variables x_i. We denote these sets of variables simply by $u = (u^1, u^2, \ldots, u^q)$ and $x = (x_1, x_2, \ldots, x_p)$, respectively. The most general case is given by a system of m non-linear differential equations

$$\Delta^i(x, u_{(k)}) = 0, \qquad i = 1, 2, \ldots, m, \tag{9.1}$$

of order k. The term $u_{(k)}$ is understood as the kth derivative of u with respect to x. We note that m, k, p, and q are arbitrary, positive integers.

In order to find the properties of (9.1) different from point symmetries, non-classical symmetries, or potential symmetries, it is suitable to apply a transformation of the independent and dependent variables as well as the derivatives. This kind of transformation determines the attributes of the corresponding group G.

First, let us consider transformations of (9.1) depending on a single parameter ϵ which are given by

$$x^* = \Xi(x, u_{(k)}; \epsilon), \tag{9.2}$$

$$u^* = \Phi(x, u_{(k)}; \epsilon) \tag{9.3}$$

with $k = 0, 1, 2 \ldots$ denoting the order of the derivatives. The functions Ξ and Φ are assumed to be differentiable with respect to the dependent and independent variables. The identity transformation of the coordinates in (9.2) and (9.3) is given by $\epsilon = 0$. In analogy to Lie's theory, the transformations (9.2)–(9.3) are generated by the vector field

$$\vec{v} = \sum_{i=1}^{p} \xi_i(x, u_{(k)}) \frac{\partial}{\partial x_i} + \sum_{\alpha=1}^{q} \phi_\alpha(x, u_{(k)}) \frac{\partial}{\partial u^\alpha}. \tag{9.4}$$

The infinitesimals ξ_i and ϕ_α are derived from equations (9.2) and (9.3) by differentiating the transformations with respect to the group parameter ϵ. Considering these expressions for $\epsilon = 0$, we get

$$\xi_i(x, u_{(k)}) = \frac{\partial \Xi(x, u_{(k)}; \epsilon)}{\partial \epsilon}\bigg|_{\epsilon=0}, \tag{9.5}$$

$$\phi_\alpha(x, u_{(k)}) = \frac{\partial \Phi(x, u_{(k)}; \epsilon)}{\partial \epsilon}\bigg|_{\epsilon=0}. \tag{9.6}$$

The infinitesimal transformations corresponding to (9.2) and (9.3) are then given by

$$x_i^* = x_i + \epsilon \xi_i(x, u_{(k)}) + o(\epsilon^2), \quad i = 1, 2, \ldots, p, \tag{9.7}$$

$$u^{\alpha *} = u^\alpha + \epsilon \phi(x, u_{(k)}) + o(\epsilon^2), \quad \alpha = 1, 2, \ldots, q, \tag{9.8}$$

with ϵ being the group parameter, a small quantity. This representation is a straight generalization of Lie's classical procedure. On the other hand, it is beneficial to introduce coordinates in which the infinitesimal transformations become the simple representation

$$x_i^* = x_i, \quad i = 1, 2, \ldots, p, \tag{9.9}$$

$$u^{\alpha *} = u^\alpha + \epsilon \left(\phi_\alpha(x, u_{(k)}) - \sum_{i=1}^{p} u_i^\alpha \, \xi_i(x, u_{(k)}) \right) + o(\epsilon^2), \quad \alpha = 1, 2, \ldots, q, \tag{9.10}$$

meaning that the complete transformations are represented by a transformation of the dependent variables. The corresponding representation of the vector field $\vec{v}_Q$ is then

$$\vec{v}_Q = \sum_{\alpha=1}^{q} Q_\alpha(x, u_{(k)}) \frac{\partial}{\partial u^\alpha}, \tag{9.11}$$

where $Q_\alpha = Q_\alpha(x, u_{(k)})$ is the characteristic of the vector field $\vec{v}_Q$. The characteristics Q_α are related to the infinitesimals by

$$Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi_i \, u_i^\alpha, \quad \alpha = 1, 2, \ldots, q. \tag{9.12}$$

It is obvious from these relations that point symmetries are a subset of generalized symmetries. Restricting the dependencies of the infinitesimals on the variables x and u, we are back in the classical theory of Lie. The determination of the characteristics follow by a similar algorithm as used in the determination of point symmetries. The determining equations for the characteristics Q_α are consequences of the relation

$$\mathrm{pr}\,\vec{v}_Q\,\Delta\,|_{\Delta=0} = 0 \tag{9.13}$$

where $\mathrm{pr}\,\vec{v}_Q$ denotes the prolongation of the vector field $\vec{v}_Q$. The general expression of the prolongation reads

$$\mathrm{pr}\,\vec{v}_Q = \sum_{\alpha=1}^{p}\sum_{J} D_J\,Q_\alpha\,\frac{\partial}{\partial u_J^\alpha} \tag{9.14}$$

with D_J the total derivative depending on the multi-index J. The invariance condition (9.13) is based on the fact that the equation $\Delta = 0$ and their differential consequences vanish on the solution manifold. From equation (9.13), we get a system of linear coupled PDEs for the characteristics Q_α if we extract the coefficients of the derivatives in u up to a certain order. We note that this order can be infinite. However, in practical calculations, we restrict the order of derivatives in Q_α to a finite number. Equation (9.13) contains all the information to derive the generalized symmetries in a nutshell. The following section discusses the procedure of how equation (9.13) is algorithmically accessible.

9.3. Algorithm for Calculation of Generalized Symmetries

The procedure to calculate the generalized symmetries proceeds in the same way as for the classical symmetries. The following steps are based on (9.13):

1. Calculate the prolongation of the system of differential equations up to kth order by

$$\mathrm{pr}\,\vec{v}_Q\,\Delta = 0, \tag{9.15}$$

   where k specifies the order of differentiation in the characteristics.

2. Use the equations and their differential consequences to eliminate redundant information of the prolongation

$$\mathrm{pr}\,\vec{v}_Q\,\Delta\,|_{\Delta=0} = 0. \tag{9.16}$$

3. Extract the determining equations from the prolongation. The determining equations follow as coefficients of the derivatives in u, i.e., u_x, $u_{x,x}$, etc. The order of derivatives used in the extraction should be greater than k.
4. Solve the resulting linear determining equations.

A selected number of examples will demonstrate the application of the algorithm. The four steps are realized in the function Baecklund[]. This function fosters the determination of the generalized symmetries. The symmetries are represented by the characteristics Q_i which are labeled in *MathLie* by $QC[i]$. Since the characteristics depend, in general, on derivatives of an infinite order, we need to restrict this order to a finite number. At the moment, this number must be specified by the user. The function Baecklund[] assumes by default that the largest order of derivatives is one. Thus, Baecklund[] determines so-called contact transformations by default. By increasing the order of derivatives $n > 1$ in the characteristics, we can force Baecklund[] to determine generalized symmetries of order n. The following examples illustrate the application of Baecklund[] to examples like the diffusion equation, the potential Burgers equation, the generalized KdV equation, and a coupled system of wave equations.

9.4. Examples

This section is concerned with application of Baecklund[] to PDEs. The examples display the capabilities and the flexibility incorporated in the function.

9.4.1 Diffusion Equation

Let us start with the diffusion equation. The four steps of the algorithm are carried out by the function Baecklund[] automatically. We first demonstrate the simplest case of application. The standard case occurs when the equation is free of any parameter. Then, only the equation, the dependent variables, and independent variables are supplied to the function Baecklund[]. In addition to these three input quantities, the side condition of equation (9.13) is needed, meaning that Baecklund[] must know a term for which the equation of motion is solvable. This term must be provided as the fourth argument. The fifth argument is not necessary for first-order generalized symmetries. The generalized symmetries of first-order for the diffusion equation thus follow by

```
generalSymm = Baecklund[
  ∂t u[x, t] - ∂x,x u[x, t] == 0, {u}, {x, t}, {∂t u[x, t]}];
generalSymm // LTF
```

$$QC[1][t, x, u, u_t, u_x] == u\left(k1 + \frac{k3\,x}{2}\right) + (k2 + k3\,t)\,u_x + \mathcal{F}_1 - (\mathcal{F}_1)_t + (\mathcal{F}_1)_{x,x} == 0$$

The result of the calculation is the representation of the characteristic QC[1] depending on the independent and the dependent variables and on first-order derivatives. Thus, Baecklund[] calculates generalized symmetries of first order by default. The result also shows that the first-order generalized symmetries consist of an infinite and a finite part of transformations. The infinite part must satisfy the original equation. This behavior is expected for a linear equation. Generalized symmetries different from contact transformations follow by increasing the order *n*. The order of the largest derivative in the characteristics is specified by the fifth argument of Baecklund[]. For example, if we are interested in second-order generalized symmetries, we initiate the calculation by

```
generalSymm = Baecklund[
  ∂t u[x, t] - ∂x,x u[x, t] == 0, {u}, {x, t}, {∂t u[x, t]}, 2];
generalSymm // LTF
```

$$QC[1][t, x, u, u_t, u_{t,t}, u_x, u_{x,t}, u_{x,x}] == \frac{1}{4}u\,(4\,k1 - 4\,k7 + 2\,k6\,t + 2\,k3\,x + k6\,x^2) + \left(k2 + \frac{k5\,x}{2} + t\,(k3 + k6\,x)\right)u_x + \mathcal{F}_1 + (k4 + t\,(k5 + k6\,t))\,u_{x,x} - (\mathcal{F}_1)_t + (\mathcal{F}_1)_{x,x} == 0$$

Now, the characteristic Q_1 depends on the second-order derivative of u. The group parameters k_i denote the six different characteristics under which the diffusion equation is invariant. The invariance can be checked by Baecklund[] if we provide additional information to the function. The additional information is carried by the option *CharExpression*→{*list*}, where *list* contains the information on the characteristics. Let us check the invariance of the diffusion equation with the second-order characteristics derived above. For the following calculation, we first need to extract the characteristics from the result *generalSymm*:

```
char = QC[1][t, x, u[x, t], u^(0,1)[x, t], u^(0,2)[x, t],
    u^(1,0)[x, t], u^(1,1)[x, t], u^(2,0)[x, t]] /. generalSymm[[1]];
char // LieTraditionalForm
```

$$\frac{1}{4}u\,(4\,k1 - 4\,k7 + 2\,k6\,t + 2\,k3\,x + k6\,x^2) + \left(k2 + \frac{k5\,x}{2} + t\,(k3 + k6\,x)\right)u_x + \mathcal{F}_1 + (k4 + t\,(k5 + k6\,t))\,u_{x,x}$$

The expression *char* is inserted into the option given as the seventh argument of Baecklund[]. The sixth argument contains the parameters of the equation if any. The check of the equation is carried out by

```
generalSymm = Baecklund[∂t u[x, t] - ∂x,x u[x, t] == 0, {u},
  {x, t}, {∂t u[x, t]}, 2, {}, CharExpression → {char}];
generalSymm // LTF
```

$$(\mathcal{F}_1)_t - (\mathcal{F}_1)_{x,x} == 0$$

We find that the arbitrary function *free*[*1*] = $\mathcal{F}_1$ has to satisfy the diffusion equation itself. The result tells us that the finite transformation properties in *char* are generalized symmetries of the diffusion equation. If the transformation group consists only of a finite group, the result would be an empty list. At this stage of our examinations, we are able to derive the generalized symmetries for a certain order of derivatives and we can check a given symmetry group to be a generalized symmetry of a specific equation.

9.4.2 Potential Burgers Equation

Another example frequently discussed in non-linear dynamics is the Burgers equation. Let us examine the potential form of this equation. In a previous section, we examined the prolongation formula of the Burgers equation. Here, we calculate the generalized symmetries. The potential Burgers equation in scaled variables read

$$u_t - u_x^2 + \beta u_{x,x} = 0. \tag{9.17}$$

Subscripts in (9.17) denote differentiations. β is a real constant measuring the dispersion strength. The generalized symmetries of first-order for the potential Burgers equation follow by

```
generalSymmBurgers =
 Baecklund[∂t u[x, t] - (∂x u[x, t])^2 + β ∂x,x u[x, t] == 0,
  {u}, {x, t}, {∂t u[x, t]}, 1, {β}];
generalSymmBurgers // LTF
```

$$QC[1][t, x, u, u_t, u_x] == C[1] + k1\, u_x + E^{u/\beta}\, \beta\, \mathcal{F}_1$$

$$\frac{(\mathcal{F}_1)_t}{\beta} + (\mathcal{F}_1)_{x,x} == 0$$

Since equation (9.17) contains the parameter β, we must tell Baecklund[] that β has a special meaning in the calculation. Because the parameters are supplied as the sixth argument, we also need to specify the order of derivatives on which the characteristics depend. This number is given in the fifth argument of Baecklund[].

The resulting characteristic depends on two parameters *k1* and *C[1]*, and an auxiliary function *free[1]* which must satisfy the diffusion equation.

9.4.3 Generalized Korteweg-de Vries Equations

The Korteweg-de Vries equation (KdV) is a standard equation with a broad range of applications. In this example, we examine the generalized symmetries of the KdV-Burgers equation (KdVB) for different geometries. The equation containing the plane, cylindrical, and spherical coordinates of the KdVB equation is

$$u_t + \frac{j}{2}\frac{u}{t} + \gamma u^m u_x - \mu u_{x,x} + \beta u_{x,x,x} = 0. \tag{9.18}$$

j determines the geometry in which the equation resides ($j = 0$ plane geometry, $j = 1$ cylindrical geometry, and $j = 2$ spherical geometry). m, γ, μ, and β are constants. For $\mu = 0$, the KdV equation follows, and for $\beta = 0$, equation (9.18) reduces to the Burgers equation. The power m distinguishes between the standard and the generalized potential form of the equation. γ measures the strength of non-linearity. Let us first examine the generalized symmetries for $m = 1$, $j = 1$, and arbitrary γ, μ, and β. Applying the function Baecklund[] to this subordinate equation, we find

```
generalKdVSymm =
 Baecklund[∂t u[x, t] + j/2 u[x, t]/t + γ u[x, t]^m ∂x u[x, t] -
     μ ∂x,x u[x, t] + β ∂x,x,x u[x, t] == 0 /. {j → 1, m → 1},
   {u}, {x, t}, {∂t u[x, t]}, 1, {γ, μ, β}];
generalKdVSymm // Flatten // LTF
```

$$QC[1][t, x, u, u_t, u_x] == -\frac{k1}{\sqrt{t}\,\gamma} + (k2 + 2\,k1\,\sqrt{t})\,u_x$$

The generalized symmetry in first-order representation consists of two symmetries: a constant term and a term describing a translation in x. The reader may check the KdVB equation for other choices of parameters and higher-order representations of the characteristics. Other third-order-equations of KdV-type include

$$u_t + u_{x,x,x} - \tfrac{1}{8}u_x^3 + u_x(\alpha e^u + \beta e^{-u} + \gamma) = 0, \tag{9.19}$$

where α, β, and γ are real constants. Equation (9.19) was discussed by Calogero and Degasperis [1981] and Fokas [1980] in connection with exactly solvable equations. The generalized symmetries of this equation follow by

```
generalKdVSymm =
 Baecklund[∂t u[x, t] + ∂x,x,x u[x, t] - 1/8 (∂x u[x, t])^3 +
```

```
∂x u[x, t] (α Exp[u[x, t]] + β Exp[-u[x, t]] + γ) == 0,
 {u}, {x, t}, {∂t u[x, t]}, 1, {α, β, γ}];
generalKdVSymm // LTF
```

```
QC[1][t, x, u, ut , ux] == k1 ux
```

Again, we find, in lowest order approximation of the characteristic, a single expression representing a translation. Another kind of KdV equation demonstrating the application of Baecklund[] to a system of equations is the Hirota and Satsuma [1981] (HS) equation, which is discussed in connection with soliton solutions. The model equations are

$$u_t + u_{x,x,x} + 6\,uu_x - 6\,vv_x = 0, \tag{9.20}$$

$$v_t - 2\,v_{x,x,x} - 6\,u\,v_x = 0. \tag{9.21}$$

The generalized symmetries of this system follow by

```
generalKdVSymm = Baecklund[{∂t u[x, t] + ∂x,x,x u[x, t] +
      6 u[x, t] ∂x u[x, t] + 2 v[x, t] ∂x v[x, t] == 0,
   ∂t v[x, t] - 2 ∂x,x,x v[x, t] - 6 u[x, t] ∂x v[x, t] == 0},
  {u, v}, {x, t}, {∂t u[x, t], ∂t v[x, t]}, 2];
generalKdVSymm // LTF
```

```
QC[1][t, x, u, v, ut , vt , ut,t , vt,t , ux , vx , ux,t , vx,t ,
  ux,x , vx,x] == k1 ux
QC[2][t, x, u, v, ut , vt , ut,t , vt,t , ux , vx , ux,t , vx,t ,
  ux,x , vx,x] == k1 vx
```

Surprisingly, the HS system has only a one-parameter finite symmetry group depending on first-order derivatives. This happens even though we examined characteristics depending on second-order derivatives.

9.4.4 Coupled System of Wave Equations

Another example to demonstrate the capabilities of Baecklund[] is the application to a coupled system of simple wave equations. This example serves to demonstrate that Baecklund[] is capable of finding symmetries of systems of equations. We generalized the example given by Olver [1986] to a system of two equations. The two fields u and v are coupled to each other via the gradients. The generalized symmetries for the system of equations

$$u_t - v\,u_x = 0, \tag{9.22}$$

$$v_t - u\,v_x = 0 \tag{9.23}$$

follow by

```
generalSymmWave = Baecklund[{∂t u[x, t] - v[x, t] ∂x u[x, t] == 0,
    ∂t v[x, t] - u[x, t] ∂x v[x, t]},
  {u, v}, {x, t}, {∂t u[x, t], ∂t v[x, t]}];
generalSymmWave // Flatten // LTF
```

$$QC[1][t, x, u, v, u_t, v_t, u_x, v_x] ==$$
$$-k4 - k1\,u - k2\,u^2 - k3\,u_x - k4\,t\,u_x + k1\,x\,u_x + k2\,v\,x\,u_x +$$
$$u_x \mathcal{F}_1 + u_x \mathcal{F}_2 + v\,u_x \mathcal{F}_3 - u_x \mathcal{F}_4 - v\,u_x \mathcal{F}_5 - u\,(\mathcal{F}_5)_t$$

$$QC[2][t, x, u, v, u_t, v_t, u_x, v_x] == -k4 - k1\,v - k2\,v^2 - k3\,v_x -$$
$$k4\,t\,v_x + k1\,x\,v_x + k2\,u\,x\,v_x + v_x \mathcal{F}_1 + v_x \mathcal{F}_2 + u\,v_x \mathcal{F}_3 - v_x \mathcal{F}_4 -$$
$$u\,v_x \mathcal{F}_5 + u\,v_x\,(\mathcal{F}_1)_v - v\,v_x\,(\mathcal{F}_1)_v - u\,v_x\,(\mathcal{F}_4)_v + v\,v_x\,(\mathcal{F}_4)_v - v\,(\mathcal{F}_5)_t$$

$$\frac{-k3 + k5 - k4\,t + k6\,t - \mathcal{F}_4 - v\,\mathcal{F}_5 + t\,\mathcal{F}_6 + \mathcal{F}_7}{t} == 0$$

$$\frac{(\mathcal{F}_2)_u}{u} + (\mathcal{F}_3)_u == 0$$

We realize that the characteristics depend on seven auxiliary functions *free[i]*. These functions are connected by one algebraic and one differential expression. The dependence in this algebraic form could not be resolved by the function PDESolve[]. However, we find information on the generalized symmetries in a straightforward way.

So far, we discussed the application of generalized symmetries to PDEs. The following section will demonstrate the application of generalized symmetries in connection with ODEs.

9.5. Second-Order ODEs and the Euler-Lagrange Equation

A great number of problems in physics can be described by second-order ordinary differential equations

$$\Delta_k\left(t, \dot{q}_j, \ddot{q}_j\right) = 0, \quad k, j = 1, 2, \ldots, N, \tag{9.24}$$

where $q_j = q_j(t)$ are a set of dependent variables. Dots denote derivatives with respect to time t. The Δ_k's are given functions of the dependent and independent variables. For example, we can think of these expressions as Newton's or Lagrange's equations.

Let us assume that (9.24) can be derived from a generating functional S. The action S and the calculus of variations are the two keys of Hamilton's principle and allow us

to derive the equations of motion (9.24). The variation of an individual path q_j in the action

$$S = \int_{t_1}^{t_2} \mathfrak{L}(t, q_j, q_j') \, dt \tag{9.25}$$

provides us with the equations of motion. We already know from the discussion in Section 3.6.1 that the calculus of variations applied to (9.25) is equivalent with the Euler derivative. The resulting equations (9.24) are known as the Euler-Lagrange equations

$$\mathfrak{E}_k(\mathfrak{L}) = \Delta_k = \frac{\delta S}{\delta q_k} = 0, \quad k = 1, 2, \ldots, N. \tag{9.26}$$

The symbol $\frac{\delta}{\delta q_k} = \sum_{n=0}^{\infty} (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial}{\partial q_{k;(n)}} \right)$ denotes the variational derivative and the Euler derivative is labeled by $\mathfrak{E}_k$. At this stage of the discussion, we know the equations of motion and certainly the physical interpretation. However, it is unknown to us under which conditions solutions of (9.24) can be found. The following section will outline how solutions of a second-order system of ODEs can be calculated by examining generalized symmetries of (9.24).

9.5.1 Generalized Symmetries and Second-Order ODEs

It is known that Lie's original procedure $\mathrm{pr}^{(2)} \vec{v} \, \Delta \, |_{\Delta=0} = 0$ does not deliver all possible symmetries of a second-order ODE. For example, Abraham-Shrauner and Guol [1993] and Olver [1986] discuss so-called *hidden symmetries* in connection with ordinary differential equations. At the turn of the century, Emmy Noether [1918] examined ODEs in connection with generalized point transformations. In turn, we have to use a generalized method to uncover the "hidden" symmetries. A generalization of the point transformations can be gained if the infinitesimals ξ and ϕ_α do not depend only on the dependent and independent variables but also on derivatives of the q_i's. In the case of second-order ODEs, the space of coordinates needs to be extended by the first derivatives. All higher derivatives can be eliminated by the differential equations themselves or differential consequences of the equations. For a shorthand notation to denote the extended dependencies including derivatives, we use square brackets $[q] = (t, q_i, \dot{q}_i)$ in the infinitesimals $\xi = \xi[q]$ and $\phi_i = \phi_i[q]$.

The corresponding generalized vector field for such transformations can formally be given by

$$\vec{v} = \xi[q] \, \partial_t + \sum_{i=1}^{N} \phi_i[q] \, \partial_{q_i} \, . \tag{9.27}$$

In a second step, we can introduce the characteristics for second-order ODEs by $Q_i[q] = \phi_i[q] - \xi[q]\,\dot{q}$. The corresponding vector field has the representation

$$\vec{v}_Q = \sum_{i=1}^{N} Q_i[q]\,\partial_{q_i} \,. \tag{9.28}$$

The symmetries contained in (9.4) and (9.28) are essentially the same. The only difference is that (9.28) considers transformations of the dependent variables and derivatives in the extended form while (9.4) additionally takes into account a transformation of the independent variables. The main advantage of the evolutionary representation is its simplicity in the representation of the prolongation

$$\overline{\mathrm{prv}}_Q = \sum_{i=1}^{N} \sum_{J=0}^{\infty} \frac{d^J}{dt^J} (Q_j[q])\,\partial_{q_{i;(J)}} \,. \tag{9.29}$$

The assumption in the calculation of the related symmetries of (9.1) is the same as in the case of point symmetries. Thus, the computation of the generalized symmetries proceeds in essentially the same way as the computation of Lie point symmetries. Since we consider the variables and derivatives as independent up to a fixed order, we not only take equations (9.1) into account but also consider the derivatives of (9.1) as concerns the invariance condition. The invariance condition remains nearly the same and reads

$$\mathrm{pr}\,\vec{v}_Q\,\Delta_k \,|_{\frac{d^J \Delta_k}{dt^J}=0} = 0 \quad \text{with } J = 0, 1, 2, \ldots \text{ and } k = 1, 2, \ldots . \tag{9.30}$$

Contrary to point symmetries, differential equations can be used to eliminate higher derivatives and also the differential consequences of equations (9.1). Equation (9.30) provides the determining equations for the characteristics Q_i. Although linear, it is very tricky to solve this system of equations, as it yields a large overdetermined system of equations. With some skill, solutions of (9.30) can be found by an ad hoc ansatz for the Q_i's; for example, by assuming that Q_i has a special form (polynomial or power form) in the arguments. By this method, equation (9.30) will split into several equations that may or may not have a non-zero solution.

In Baecklund[], we use an ansatz of polynomial form in the coordinates q_i and momenta $\dot{q}_i$. This ansatz is exclusively used for second-order ODEs. If we can solve the resulting system of linear equations in the coefficients introduced by the ansatz, we will have determined the characteristics up to some power. This information can, in turn, be used to construct conservation laws and invariant solutions.

9.5.2 Conservation Laws

With equation (9.26), we assume that the system of equations (9.1) is derivable by a variation of the path q_i. In close analogy to the question of invariance of (9.1), we can ask for the invariance of the action itself. This means that we have to assume

$$\int_{t_1}^{t_2} \mathfrak{L}(t, q_i, \dot{q}_i)\, dt = \int_{t_1^*}^{t_2^*} \mathfrak{L}(t^*, q_i^*, \dot{q}_i^*)\, dt^* . \tag{9.31}$$

Two questions arise at this point:

1. Are the group elements $g \in G$ of (9.31) derivable from infinitesimal conditions?
2. How are the symmetries of the Euler-Lagrange equations related to these variational symmetries?

It is obvious that local transformations preserving the invariance of a functional transform one solution of the variational problem into another. This must be the case if relation (9.31) is correct. Thus, we can use Lie's procedure applied to the Euler-Lagrange equations corresponding to (9.26), taking into account that not all symmetries of equations (9.1) are symmetries of (9.26). To uncover the variational symmetries, we can use the methods discussed so far but, in addition, need to check the relation

$$\mathrm{pr}\, \vec{v}_Q \Delta_k + \mathfrak{D}_Q^* \Delta_k = 0, \quad k = 1, 2, \ldots, N, \tag{9.32}$$

where $\mathfrak{D}_Q^*$ denotes the adjoint Fréchet derivative of the characteristic Q. The evolutionary vector field $\vec{v}_Q$ with its characteristic Q is a variational symmetry if equation (9.32) holds for all t and q_i. We now have a criterion singling out these symmetries from the generalized symmetries which are also variational symmetries. Noether's theorem establishes the connection between the variational symmetries with their characteristics Q_i and the conserved quantities. The case of one independent variable can be summarized in the formula

$$\frac{dI}{dt} = \sum_{i=1}^{N} Q_i \Delta_i = 0, \tag{9.33}$$

which is a special case of the more general divergence formula given by Olver [1986]. The essential points in determining integrals of motion for a given Euler-Lagrange system are, first, to find the characteristics Q_i for the variational symmetries, and, second, to integrate equation (9.33). These two steps among others are implemented in *MathLie*. To show how the outlined analysis can be used in

practical applications, we will give some examples demonstrating the application of Baecklund[]. However, we first collect the essential steps of the algorithm to calculate integrals for second-order equations.

9.6. Algorithm for Conservation Laws of Second-Order ODEs

The algorithm uses the notation by Olver [1986]. The work of Olver is instrumental in a direct implementation. Using the earlier notations, the algorithm involves the following steps:

1. Applying the prolongation operator $\mathrm{pr}\,\vec{v}_Q$ to each equation $\Delta_i(x, u_{(k)})$ and requiring that equation (9.30) be satisfied.
2. Consider equation (9.30) as an algebraic system where derivatives become the new variables. Provided this system of algebraic equations can be solved by Solve[], we can gain m solutions w^β with $\beta = 1, 2, \ldots, m$ which contain the equations $\Delta = 0$. The variables w^β and derivatives of these relations are used as side conditions in equation (9.30).
3. Extract the determining equations for the characteristics Q_α by equating the coefficients of all functionally independent expressions in the derivatives u_j^α to zero.
4. Simplifying the total number of equations by using the first-order derivatives and homogeneous higher-order derivatives for the characteristics. This step is repeated until the number of determining equations reaches a stable value.
5. If the determining equations are known, it may be possible to solve these equations by using an ansatz of polynomials for the characteristics. We note that this step will not provide the most general solution of ordinary differential equations, if such one exists, but delivers at least a sub-class of solutions.

This procedure works expediently as long as the variables w^β are obtainable by Solve[] from the equations of motion (9.1) and if Solve[] can find a solution from the linear equations for the constants used in the ansatz of the characteristics. The manual procedure, feeding back some information on the characteristics gained by partly solving the determining equations, is completely independent of the automatic solution procedure. If one is convinced that the ansatz used in Baecklund[] does not provide the most general solution, one can use a manual and iterative procedure to find the solutions of the determining equations.

9.7. Examples for Second-Order ODEs

In this section, we will deal with the application of the generalized symmetry method previously introduced. We discuss the Hénon-Heiles model, a quartic anharmonic oscillator, and the problem of two ions in a Paul trap.

9.7.1 The Hénon-Heiles Model

The Hénon-Heiles model for gravitating stars in a cylindrical galaxy is described by a set of two coupled non-linear second-order ODEs. The model in its original form with fixed parameters was used by Hénon and Heiles [1964] to examine the regular and chaotic motion of a star in a galaxy. Here, we are interested in the regular motion of the Hénon-Heiles system. The two equations of motion are

$$\ddot{q}_1 + A q_1 + 2 D q_1 q_2 = 0, \tag{9.34}$$

$$\ddot{q}_2 + B q_2 + D q_1^2 - C q_2^2 = 0, \tag{9.35}$$

where A, B, C, and D are model parameters. In the original equations of Hénon and Heiles, these parameters were fixed to a certain value. For these values, Hénon and Heiles found chaotic behavior of the model if the total energy of the system is increased. Our interest here is to find those parameter combinations for which equations (9.34) and (9.35) possess symbolic solutions. It is known that the two equations of motion follow from the Lagrangian

$$L = \int \left(\frac{1}{2} \left(\dot{q}_1^2 + \dot{q}_2^2 \right) - \frac{A}{2} q_1^2 - \frac{B}{2} q_2^2 - D q_1^2 q_2 + \frac{C}{3} q_2^3 \right) dt \,. \tag{9.36}$$

Applying the Euler derivative to the density of the Lagrangian, we can write down the equations of motion (9.34) and (9.35) by

```
HenonHeiles =
 Thread[ℰ{q1,q2}^t [(1/2 ((∂t q1[t])^2 + (∂t q2[t])^2) - A/2 q1[t]^2 -
       B/2 q2[t]^2 - D q1[t]^2 q2[t] + C/3 q2[t]^3)] == {0, 0}];
HenonHeiles // LTF
-A q1 - 2 D q1 q2 - (q1)t,t == 0
-D q1^2 - B q2 + C q2^2 - (q2)t,t == 0
```

Given an arbitrary choice of the parameters A, B, C, and D, chaotic evolution occurs for the coordinates q_1 and q_2. The non-chaotic or regular cases occur if the motion of

the star is controlled at least by two integrals of motion. These two integrals fix the path of the system in the two-dimensional phase space. The special combinations of the parameters for which integrals of motion exists were derived with a Painlevé test by Bountis et al. [1982]. Here, we use Baecklund[] to find the special parameter combinations under which integrals of motion exist. To find the integrals of motion, we apply the theory presented above. The integrals of motion are accessible by

```
HHIntegrals = Baecklund[HenonHeiles, {q1, q2},
  {t}, {∂t,t q1[t], ∂t,t q2[t]}, 1, {A, B, C, D}, AnsatzPoly → {1, 1}];
HHIntegrals // LieTraditionalForm // TableForm
```

0 $(q_2)_t$	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$-2 A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} B q_2^2 + \frac{C q_2^3}{3} - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to A$ $C \to -6 D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to A$ $C \to -D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} - \frac{D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to C$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} + \frac{C q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to -6 D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$

$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to -\frac{2D}{5}$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} - \frac{2 D q_2^3}{15} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_2)_t$ $(q_1)_t$	$A \to A$ $B \to A$ $C \to -D$ $D \to D$	$-\frac{1}{3} D q_1^3 - A q_1 q_2 - D q_1 q_2^2 - (q_1)_t (q_2)_t$
$-2 q_2 (q_1)_t + q_1 (q_2)_t$ $q_1 (q_1)_t$	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$q_2 (-A q_1^2 + (q_1)_t^2) - q_1 (q_1)_t (q_2)_t$
$-2 q_2 (q_1)_t + q_1 (q_2)_t$ $q_1 (q_1)_t -$ $(3 A (q_2)_t) / (2 D)$	$A \to A$ $B \to A$ $C \to -6 D$ $D \to D$	$-\frac{1}{4} D q_1^4 + \frac{(3 A^2 - 4 D^2 q_1^2) q_2^2}{4 D} + 3 A q_2^3 +$ $\frac{1}{2} q_2 (A q_1^2 + 2 (q_1)_t^2) - q_1 (q_1)_t (q_2)_t + \frac{3 A (q_2)_t^2}{4 D}$
$-2 q_2 (q_1)_t + q_1 (q_2)_t$ $q_1 (q_1)_t +$ $((-4 A + B) (q_2)_t) / (2 D)$	$A \to A$ $B \to B$ $C \to -6 D$ $D \to D$	$-\frac{1}{4} D q_1^4 - \frac{(-4 A B + B^2 + 4 D^2 q_1^2) q_2^2}{4 D} + (4 A - B) q_2^3 +$ $\frac{1}{2} q_2 (2 A q_1^2 - B q_1^2 + 2 (q_1)_t^2) - q_1 (q_1)_t (q_2)_t +$ $\frac{(4 A - B) (q_2)_t^2}{4 D}$

The result is a list containing several sub-lists. The sub-lists gather the information on the symmetries, the parameter combinations, and the related integral of motion. For example, the ninth element of the result

```
HHIntegrals[[9]] // LieTraditionalForm
```

$$\left\{\{(q_1)_t, (q_2)_t\}, \{A \to A, B \to B, C \to C, D \to D\}, -\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} + \frac{C q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2\right\}$$

states that the characteristics are given by the pair ($q1'[t]$, $q2'[t]$) and that all parameters are arbitrary. The related integral of motion is just the total energy of the system. Thus we find that for arbitrary parameter combinations, energy is a conserved quantity. The total energy is one of the two needed integrals to fix a path of the particle. Another integral of motion is given as the seventh element in

```
HHIntegrals[[7]] // LieTraditionalForm
```

$$\left\{\{(q_1)_t, (q_2)_t\}, \{A \to A, B \to A, C \to -6 D, D \to D\}, -\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2\right\}$$

Here, the characteristics are the same as before. However, the special combination of parameters (A and D arbitrary, $B = A$, and $C = -6\,D$) allows a second integral of motion in addition to the total energy. These two conserved quantities completely fix the motion of the star and the galaxy.

In the above calculations, the option *AnsatzPoly→{1,1}* of Baecklund[] was used. This option affects Baecklund[] in two ways. First, it creates a polynomial ansatz for

the characteristics, and second, it determines the coefficients of this polynomial in such a way that integrals of motion exist. The two values in the list specify the polynomial degree in the coordinates and in the first derivatives, respectively. If you are convinced that the characteristics depend on the third order of the coordinates and on third order in the derivatives, you can call Baecklund[] by

```
HHIntegrals = Baecklund[HenonHeiles, {q1, q2},
  {t}, {∂t,t q1[t], ∂t,t q2[t]}, 1, {A, B, C, D}, AnsatzPoly → {3, 3}];
HHIntegrals // LieTraditionalForm // TableForm
```

0 $(q_2)_t$	$A \to 0$ $B \to 0$ $C \to C$ $D \to 0$	$\frac{C q_2^3}{3} - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to 0$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{2} B q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to 0$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} B q_2^2 + \frac{C q_2^3}{3} - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to \frac{A}{9}$ $C \to 0$ $D \to 0$	$-\frac{1}{18} A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to \frac{A}{4}$ $C \to 0$ $D \to 0$	$-\frac{1}{8} A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to A$ $C \to C$ $D \to 0$	$-\frac{1}{2} A q_2^2 + \frac{C q_2^3}{3} - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$-2 A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to 9 A$ $C \to 0$ $D \to 0$	$-\frac{9}{2} A q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{2} B q_2^2 - \frac{1}{2} (q_2)_t^2$
0 $(q_2)_t$	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} B q_2^2 + \frac{C q_2^3}{3} - \frac{1}{2} (q_2)_t^2$

0 $\frac{1}{9} A q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to \frac{A}{9}$ $C \to 0$ $D \to 0$	$-\frac{1}{324} A^2 q_2^4 - \frac{1}{18} A q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $\frac{1}{4} A q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to \frac{A}{4}$ $C \to 0$ $D \to 0$	$-\frac{1}{64} A^2 q_2^4 - \frac{1}{8} A q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $A q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_2^4 - \frac{1}{2} A q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $4 A q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$-4 A^2 q_2^4 - 2 A q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $9 A q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to 9 A$ $C \to 0$ $D \to 0$	$-\frac{81}{4} A^2 q_2^4 - \frac{9}{2} A q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $B q_2^2 (q_2)_t + (q_2)_t^3$	$A \to 0$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{4} B^2 q_2^4 - \frac{1}{2} B q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $B q_2^2 (q_2)_t + (q_2)_t^3$	$A \to A$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{4} B^2 q_2^4 - \frac{1}{2} B q_2^2 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $-\frac{2}{3} C q_2^3 (q_2)_t + (q_2)_t^3$	$A \to 0$ $B \to 0$ $C \to C$ $D \to 0$	$-\frac{1}{9} C^2 q_2^6 + \frac{1}{3} C q_2^3 (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $\left(A q_2^2 - \frac{2 C q_2^3}{3}\right) (q_2)_t +$ $(q_2)_t^3$	$A \to A$ $B \to A$ $C \to C$ $D \to 0$	$-\frac{1}{4} A^2 q_2^4 + \frac{1}{3} A C q_2^5 - \frac{1}{9} C^2 q_2^6 -$ $\frac{1}{6} q_2^2 (3 A - 2 C q_2) (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $\left(B q_2^2 - \frac{2 C q_2^3}{3}\right) (q_2)_t +$ $(q_2)_t^3$	$A \to 0$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{4} B^2 q_2^4 + \frac{1}{3} B C q_2^5 - \frac{1}{9} C^2 q_2^6 -$ $\frac{1}{6} q_2^2 (3 B - 2 C q_2) (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
0 $\left(B q_2^2 - \frac{2 C q_2^3}{3}\right) (q_2)_t +$ $(q_2)_t^3$	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{4} B^2 q_2^4 + \frac{1}{3} B C q_2^5 - \frac{1}{9} C^2 q_2^6 -$ $\frac{1}{6} q_2^2 (3 B - 2 C q_2) (q_2)_t^2 - \frac{1}{4} (q_2)_t^4$
$(q_1)_t$ 0	$A \to 0$ $B \to 0$ $C \to C$ $D \to 0$	$-\frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to 0$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to 0$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} (q_1)_t^2$

$(q_1)_t$ 0	$A \to A$ $B \to \frac{A}{9}$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to \frac{A}{4}$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to A$ $C \to C$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to 4\,A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to 9\,A$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ 0	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{2} A q_1^2 - \frac{1}{2} (q_1)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to 0$ $B \to 0$ $C \to -16\,D$ $D \to D$	$-D q_1^2 q_2 - \frac{16 D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to 0$ $B \to 0$ $C \to -6\,D$ $D \to D$	$-D q_1^2 q_2 - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to A$ $C \to C$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} + \frac{C q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to A$ $C \to -6\,D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to A$ $C \to -D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{A q_2^2}{2} - \frac{D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to 16\,A$ $C \to -16\,D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - 8 A q_2^2 - \frac{16 D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$

$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to C$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} + \frac{C q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to -16 D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} - \frac{16 D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to -6 D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} - 2 D q_2^3 - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$A \to A$ $B \to B$ $C \to -D$ $D \to D$	$-\frac{1}{2} A q_1^2 - D q_1^2 q_2 - \frac{B q_2^2}{2} - \frac{D q_2^3}{3} - \frac{1}{2} (q_1)_t^2 - \frac{1}{2} (q_2)_t^2$
$(q_1)_t^3$ 0	$A \to 0$ $B \to 0$ $C \to C$ $D \to 0$	$-\frac{1}{4} (q_1)_t^4$
$(q_1)_t^3$ 0	$A \to 0$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{4} (q_1)_t^4$
$(q_1)_t^3$ 0	$A \to 0$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to \frac{A}{9}$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to \frac{A}{4}$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to A$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to A$ $C \to C$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to 4 A$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to 9 A$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to B$ $C \to 0$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$

$A q_1^2 (q_1)_t + (q_1)_t^3$ 0	$A \to A$ $B \to B$ $C \to C$ $D \to 0$	$-\frac{1}{4} A^2 q_1^4 - \frac{1}{2} A q_1^2 (q_1)_t^2 - \frac{1}{4} (q_1)_t^4$
$(q_2)_t$ $(q_1)_t$	$A \to A$ $B \to A$ $C \to -D$ $D \to D$	$-\frac{1}{3} D q_1^3 - A q_1 q_2 - D q_1 q_2^2 - (q_1)_t (q_2)_t$
$(\frac{3A}{2D} - 2 q_2) (q_1)_t +$ $q_1 (q_2)_t$ $q_1 (q_1)_t$	$A \to A$ $B \to A$ $C \to -6 D$ $D \to D$	$-\frac{3 A^2 q_1^2}{4 D} - \frac{D q_1^4}{4} - D q_1^2 q_2^2 - \frac{3 A (q_1)_t^2}{4 D} +$ $q_2 (-A q_1^2 + (q_1)_t^2) - q_1 (q_1)_t (q_2)_t$
$(\frac{4A-B}{2D} - 2 q_2) (q_1)_t +$ $q_1 (q_2)_t$ $q_1 (q_1)_t$	$A \to A$ $B \to B$ $C \to -6 D$ $D \to D$	$-\frac{A (4A-B) q_1^2}{4 D} - \frac{D q_1^4}{4} - D q_1^2 q_2^2 - \frac{(4A-B) (q_1)_t^2}{4 D} +$ $q_2 (-A q_1^2 + (q_1)_t^2) - q_1 (q_1)_t (q_2)_t$
$-2 q_2 (q_1)_t + q_1 (q_2)_t$ $q_1 (q_1)_t$	$A \to 0$ $B \to 0$ $C \to -6 D$ $D \to D$	$-\frac{1}{4} D q_1^4 - D q_1^2 q_2^2 + q_2 (q_1)_t^2 - q_1 (q_1)_t (q_2)_t$
$2 D q_1^2 q_2 (q_1)_t + (q_1)_t^3 -$ $\frac{1}{3} D q_1^3 (q_2)_t$ $-\frac{1}{3} D q_1^3 (q_1)_t$	$A \to 0$ $B \to 0$ $C \to -16 D$ $D \to D$	$\frac{1}{18} D^2 q_1^6 + \frac{1}{3} D^2 q_1^4 q_2^2 - D q_1^2 q_2 (q_1)_t^2 -$ $\frac{1}{4} (q_1)_t^4 + \frac{1}{3} D q_1^3 (q_1)_t (q_2)_t$
$(A q_1^2 + 2 D q_1^2 q_2) (q_1)_t +$ $(q_1)_t^3 - \frac{1}{3} D q_1^3 (q_2)_t$ $-\frac{1}{3} D q_1^3 (q_1)_t$	$A \to A$ $B \to 16 A$ $C \to -16 D$ $D \to D$	$-\frac{1}{4} A^2 q_1^4 +$ $\frac{1}{18} D^2 q_1^6 + \frac{1}{3} D^2 q_1^4 q_2^2 - \frac{1}{2} A q_1^2 (q_1)_t^2 -$ $\frac{1}{4} (q_1)_t^4 + \frac{1}{3} D q_1^2 q_2 (A q_1^2 - 3 (q_1)_t^2) +$ $\frac{1}{3} D q_1^3 (q_1)_t (q_2)_t$
$\left(A q_1^2 + 2 D q_1^2 q_2 + A q_2^2 + \frac{2 D q_2^3}{3}\right) (q_1)_t +$ $(q_1)_t^3 + (-\frac{1}{3} D q_1^3 - A q_1 q_2 - D q_1 q_2^2)$ $(q_2)_t$ $(-\frac{1}{3} D q_1^3 - A q_1 q_2 - D q_1 q_2^2) (q_1)_t +$ $\left(A q_1^2 + 2 D q_1^2 q_2 + A q_2^2 + \frac{2 D q_2^3}{3}\right)$ $(q_2)_t +$ $(q_2)_t^3$	$A \to A$ $B \to A$ $C \to -D$ $D \to D$	$-\frac{1}{4} A^2 q_1^4 + \frac{1}{18} D^2 q_1^6 + \frac{1}{12} (-3 A^2 - 2 D^2 q_1^2) q_2^4 -$ $\frac{1}{3} A D q_2^5 - \frac{1}{9} D^2 q_2^6 - \frac{1}{2} A q_1^2 (q_1)_t^2 -$ $\frac{1}{4} (q_1)_t^4 + \frac{1}{3} D q_2^3 (-A q_1^2 - (q_1)_t^2) -$ $\frac{1}{3} D q_1^2 q_2 (2 A q_1^2 + 3 (q_1)_t^2) +$ $\frac{1}{6} q_2^2 (-4 D^2 q_1^4 - 3 A (q_1)_t^2) +$ $\frac{1}{3} q_1 (D q_1^2 + 3 A q_2 + 3 D q_2^2) (q_1)_t (q_2)_t +$ $\frac{1}{6} (-3 A q_1^2 - 6 D q_1^2 q_2 - 3 A q_2^2 - 2 D q_2^3) (q_2)_t^2 -$ $\frac{1}{4} (q_2)_t^4$

We realize that the structure of the characteristics change and a large number of new integrals occur. These integrals represent the solution of the equation in implicit form.

9.7.2 Two-Dimensional Quartic Oscillators

When two or more optical waves copropagate inside a fiber, they can interact with each other through the fiber non-linearity. In general, such an interaction can generate new waves under appropriate conditions through a variety of non-linear phenomena such as stimulated Raman and Brillouin scattering. The same non-linearity also provides a coupling between the incident waves through a phenomenon referred to as cross-phase modulation. Cross-phase modulation is always accompanied by self-phase modulation and occurs because the effective refractive index of a wave depends not only on the intensity of that wave but also on the intensity of the copropagating wave.

In this subsection, we will consider the coupling of two polarized waves with different frequencies. We assume that we have a polarization-preserving fiber so that the two waves maintain their polarization during propagation. This model, consisting of two coupled non-linear Schrödinger equations, can be reduced to a Hamiltonian system (cf. Baumann [1991]). The reduced model shows regular solutions for some parameter values. On the other hand, the Hamiltonian system can show chaos for an arbitrary choice of parameter values. The equations of motion follow from the Hamiltonian

$$H = \frac{1}{2}\sum_{i=1}^{2}\left(p_i^2 + \frac{1}{2}\alpha_i q_i^2 + \Gamma_i q_i^4\right) + q_1^2 q_2^2, \tag{9.37}$$

where p_i denotes the momenta and q_i the coordinates of the reduced equations. The α_i's and Γ_i's are real constants. The equivalent formulation in Lagrange's dynamic starts from the Lagrangian

$$L = \frac{1}{2}\sum_{i=1}^{2}\left(\dot{q}_i^2 - \frac{1}{2}\alpha_i q_i^2 - \Gamma_i q_i^4\right) - q_1^2 q_2^2 . \tag{9.38}$$

Applying the Euler derivative to this Lagrangian, we find the equations of motion as

```
quarticOscillators =
 Thread[ℰ^t_{q1,q2}[(1/2 ((∂t q1[t])^2 + (∂t q2[t])^2 -
          1/2 (α1 q1[t]^2 + α2 q2[t]^2) - Γ1 q1[t]^4 - Γ2 q2[t]^4) -
      q1[t]^2 q2[t]^2)] ==
   {0, 0}];
quarticOscillators // LTF
```

$$-\frac{\alpha 1\, q_1}{2} - 2\,\Gamma 1\, q_1^3 - 2\, q_1\, q_2^2 - (q_1)_{t,t} == 0$$

$$-\frac{\alpha 2\, q_2}{2} - 2\, q_1^2\, q_2 - 2\,\Gamma 2\, q_2^3 - (q_2)_{t,t} == 0$$

These coupled second-order equations represent a similarity reduction of two coupled non-linear Schrödinger equations. The question now is under what circumstances is the given system of equations solvable? The answer is given by Baecklund[], which delivers

```
QOIntegrals = Baecklund[quarticOscillators, {q1, q2}, {t},
  {∂t,t q1[t], ∂t,t q2[t]}, 1, {α1, α2, Γ1, Γ2}, AnsatzPoly → {4, 3}];
QOIntegrals // LieTraditionalForm // TableForm
```

$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{1}{6}$ $\Gamma 2 \to \frac{8}{3}$	$-\frac{q_1^4}{12} - q_1^2\, q_2^2 - \frac{4\, q_2^4}{3} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{1}{3}$ $\Gamma 2 \to \frac{1}{3}$	$-\frac{q_1^4}{6} - q_1^2\, q_2^2 - \frac{q_2^4}{6} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{1}{3}$ $\Gamma 2 \to \frac{8}{3}$	$-\frac{q_1^4}{6} - q_1^2\, q_2^2 - \frac{4\, q_2^4}{3} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$-\frac{q_1^4}{2} - q_1^2\, q_2^2 - \frac{q_2^4}{2} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{8}{3}$ $\Gamma 2 \to \frac{1}{6}$	$-\frac{4\, q_1^4}{3} - q_1^2\, q_2^2 - \frac{q_2^4}{12} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{8}{3}$ $\Gamma 2 \to \frac{1}{3}$	$-\frac{4\, q_1^4}{3} - q_1^2\, q_2^2 - \frac{q_2^4}{6} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \frac{\alpha 1}{4}$ $\Gamma 1 \to \frac{1}{3}$ $\Gamma 2 \to \frac{1}{3}$	$-\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + \frac{1}{16}(-\alpha 1 - 16\, q_1^2)\, q_2^2 - \frac{q_2^4}{6} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$
$(q_1)_t$ $(q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \frac{\alpha 1}{4}$ $\Gamma 1 \to \frac{1}{3}$ $\Gamma 2 \to \frac{8}{3}$	$-\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + \frac{1}{16}(-\alpha 1 - 16\, q_1^2)\, q_2^2 - \frac{4\, q_2^4}{3} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2$

$$
\begin{array}{lll}
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \frac{\alpha 1}{4} & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{2} + \frac{1}{16}(-\alpha 1 - 16\, q_1^2)\, q_2^2 - \frac{q_2^4}{2} \\
(q_2)_t & \Gamma 1 \to 1 & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to 1 & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \frac{\alpha 1}{4} & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{4\, q_1^4}{3} + \frac{1}{16}(-\alpha 1 - 16\, q_1^2)\, q_2^2 - \\
(q_2)_t & \Gamma 1 \to \frac{8}{3} & \frac{q_2^4}{12} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{6} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \frac{\alpha 1}{4} & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{4\, q_1^4}{3} + \frac{1}{16}(-\alpha 1 - 16\, q_1^2)\, q_2^2 - \\
(q_2)_t & \Gamma 1 \to \frac{8}{3} & \frac{q_2^4}{6} - \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + \frac{1}{4}(-\alpha 1 - 4\, q_1^2)\, q_2^2 - \frac{q_2^4}{6} - \\
(q_2)_t & \Gamma 1 \to \frac{1}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + \frac{1}{4}(-\alpha 1 - 4\, q_1^2)\, q_2^2 - \frac{4\, q_2^4}{3} - \\
(q_2)_t & \Gamma 1 \to \frac{1}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{8}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{2} + \frac{1}{4}(-\alpha 1 - 4\, q_1^2)\, q_2^2 - \frac{q_2^4}{2} - \\
(q_2)_t & \Gamma 1 \to 1 & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to 1 & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to \alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{4\, q_1^4}{3} + \frac{1}{4}(-\alpha 1 - 4\, q_1^2)\, q_2^2 - \frac{q_2^4}{6} - \\
(q_2)_t & \Gamma 1 \to \frac{8}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to 4\,\alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{12} + (-\alpha 1 - q_1^2)\, q_2^2 - \frac{4\, q_2^4}{3} - \\
(q_2)_t & \Gamma 1 \to \frac{1}{6} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{8}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to 4\,\alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + (-\alpha 1 - q_1^2)\, q_2^2 - \frac{q_2^4}{6} - \\
(q_2)_t & \Gamma 1 \to \frac{1}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to 4\,\alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{6} + (-\alpha 1 - q_1^2)\, q_2^2 - \frac{4\, q_2^4}{3} - \\
(q_2)_t & \Gamma 1 \to \frac{1}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{8}{3} & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to 4\,\alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{q_1^4}{2} + (-\alpha 1 - q_1^2)\, q_2^2 - \frac{q_2^4}{2} - \\
(q_2)_t & \Gamma 1 \to 1 & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to 1 & \\
 & \alpha 1 \to \alpha 1 & \\
(q_1)_t & \alpha 2 \to 4\,\alpha 1 & -\frac{1}{4}\alpha 1\, q_1^2 - \frac{4\, q_1^4}{3} + (-\alpha 1 - q_1^2)\, q_2^2 - \frac{q_2^4}{6} - \\
(q_2)_t & \Gamma 1 \to \frac{8}{3} & \frac{1}{2}(q_1)_t^2 - \frac{1}{2}(q_2)_t^2 \\
 & \Gamma 2 \to \frac{1}{3} &
\end{array}
$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{1}{6}$
$\Gamma 2 \to \Gamma 2$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{12}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{\Gamma 2\,q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{1}{3}$
$\Gamma 2 \to \frac{1}{3}$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{6}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{q_2^4}{6}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{1}{3}$
$\Gamma 2 \to \frac{8}{3}$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{6}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{4\,q_2^4}{3}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{1}{3}$
$\Gamma 2 \to \Gamma 2$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{6}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{\Gamma 2\,q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to 1$
$\Gamma 2 \to 1$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{2}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to 1$
$\Gamma 2 \to \Gamma 2$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{q_1^4}{2}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{\Gamma 2\,q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{8}{3}$
$\Gamma 2 \to \frac{1}{3}$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{4\,q_1^4}{3}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{q_2^4}{6}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \frac{8}{3}$
$\Gamma 2 \to \Gamma 2$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{4\,q_1^4}{3}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{\Gamma 2\,q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \Gamma 1$
$\Gamma 2 \to \frac{1}{6}$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{\Gamma 1\,q_1^4}{2}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{q_2^4}{12}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_1)_t$
$(q_2)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 2$
$\Gamma 1 \to \Gamma 1$
$\Gamma 2 \to \Gamma 2$

$$-\frac{1}{4}\,\alpha 1\,q_1^2-\frac{\Gamma 1\,q_1^4}{2}+\frac{1}{4}\,(-\alpha 2-4\,q_1^2)\,q_2^2-\frac{\Gamma 2\,q_2^4}{2}-\frac{1}{2}\,(q_1)_t^2-\frac{1}{2}\,(q_2)_t^2$$

$(q_2)_t$
$(q_1)_t$

$\alpha 1 \to 0$
$\alpha 2 \to 0$
$\Gamma 1 \to \frac{1}{3}$
$\Gamma 2 \to \frac{1}{3}$

$$-\frac{2}{3}\,q_1^3\,q_2-\frac{2}{3}\,q_1\,q_2^3-(q_1)_t\,(q_2)_t$$

$(q_2)_t$
$(q_1)_t$

$\alpha 1 \to \alpha 1$
$\alpha 2 \to \alpha 1$
$\Gamma 1 \to \frac{1}{3}$
$\Gamma 2 \to \frac{1}{3}$

$$-\frac{1}{6}\,q_1\,(3\,\alpha 1+4\,q_1^2)\,q_2-\frac{2}{3}\,q_1\,q_2^3-(q_1)_t\,(q_2)_t$$

$q_2\ (q_2)_t$ $q_2\ (q_1)_t -$ $2\ q_1\ (q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{8}{3}$ $\Gamma 2 \to \frac{1}{6}$	$-\frac{2}{3}\ q_1^3\ q_2^2 - \frac{1}{3}\ q_1\ q_2^4 - q_2\ (q_1)_t\ (q_2)_t + q_1\ (q_2)_t^2$
$q_2\ (q_2)_t$ $q_2\ (q_1)_t -$ $2\ q_1\ (q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \frac{\alpha 1}{4}$ $\Gamma 1 \to \frac{8}{3}$ $\Gamma 2 \to \frac{1}{6}$	$-\frac{1}{24}\ q_1\ (3\ \alpha 1 + 16\ q_1^2)\ q_2^2 - \frac{1}{3}\ q_1\ q_2^4 -$ $q_2\ (q_1)_t\ (q_2)_t + q_1\ (q_2)_t^2$
$-2\ q_2\ (q_1)_t +$ $q_1\ (q_2)_t$ $q_1\ (q_1)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to \frac{1}{6}$ $\Gamma 2 \to \frac{8}{3}$	$-\frac{2}{3}\ q_1^2\ q_2^3 + \frac{1}{3}\ q_2\ (-q_1^4 + 3\ (q_1)_t^2) -$ $q_1\ (q_1)_t\ (q_2)_t$
$-2\ q_2\ (q_1)_t +$ $q_1\ (q_2)_t$ $q_1\ (q_1)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to 4\ \alpha 1$ $\Gamma 1 \to \frac{1}{6}$ $\Gamma 2 \to \frac{8}{3}$	$-\frac{2}{3}\ q_1^2\ q_2^3 + \frac{1}{6}\ q_2\ (-3\ \alpha 1\ q_1^2 - 2\ q_1^4 + 6\ (q_1)_t^2) -$ $q_1\ (q_1)_t\ (q_2)_t$
$-q_2^2\ (q_1)_t +$ $q_1\ q_2\ (q_2)_t$ $q_1\ q_2\ (q_1)_t -$ $q_1^2\ (q_2)_t$	$\alpha 1 \to 0$ $\alpha 2 \to 0$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$\frac{1}{2}\ q_2^2\ (q_1)_t^2 - q_1\ q_2\ (q_1)_t\ (q_2)_t + \frac{1}{2}\ q_1^2\ (q_2)_t^2$
$-q_2^2\ (q_1)_t +$ $q_1\ q_2\ (q_2)_t$ $q_1\ q_2\ (q_1)_t -$ $q_1^2\ (q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \alpha 1$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$\frac{1}{2}\ q_2^2\ (q_1)_t^2 - q_1\ q_2\ (q_1)_t\ (q_2)_t + \frac{1}{2}\ q_1^2\ (q_2)_t^2$
$(-\frac{3\ \alpha 1}{2} - q_2^2)$ $(q_1)_t +$ $q_1\ q_2\ (q_2)_t$ $q_1\ q_2\ (q_1)_t -$ $q_1^2\ (q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to 4\ \alpha 1$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$\frac{3}{8}\ \alpha 1^2\ q_1^2 + \frac{3\ \alpha 1\ q_1^4}{4} + \frac{3}{4}\ \alpha 1\ (q_1)_t^2 +$ $\frac{1}{4}\ q_2^2\ (3\ \alpha 1\ q_1^2 + 2\ (q_1)_t^2) - q_1\ q_2\ (q_1)_t\ (q_2)_t +$ $\frac{1}{2}\ q_1^2\ (q_2)_t^2$
$(\frac{3\ \alpha 1}{8} - q_2^2)$ $(q_1)_t +$ $q_1\ q_2\ (q_2)_t$ $q_1\ q_2\ (q_1)_t -$ $q_1^2\ (q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \frac{\alpha 1}{4}$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$-\frac{3}{32}\ \alpha 1^2\ q_1^2 - \frac{3\ \alpha 1\ q_1^4}{16} -$ $\frac{3}{16}\ \alpha 1\ (q_1)_t^2 + \frac{1}{16}\ q_2^2\ (-3\ \alpha 1\ q_1^2 + 8\ (q_1)_t^2) -$ $q_1\ q_2\ (q_1)_t\ (q_2)_t + \frac{1}{2}\ q_1^2\ (q_2)_t^2$
$(\frac{1}{2}\ (\alpha 1 - \alpha 2) -$ $q_2^2)$ $(q_1)_t +$ $q_1\ q_2\ (q_2)_t$ $q_1\ q_2\ (q_1)_t -$ $q_1^2\ (q_2)_t$	$\alpha 1 \to \alpha 1$ $\alpha 2 \to \alpha 2$ $\Gamma 1 \to 1$ $\Gamma 2 \to 1$	$-\frac{1}{8}\ \alpha 1\ (\alpha 1 - \alpha 2)\ q_1^2 +$ $\frac{1}{4}\ (-\alpha 1 + \alpha 2)\ q_1^4 + \frac{1}{4}\ (-\alpha 1 + \alpha 2)\ (q_1)_t^2 +$ $\frac{1}{4}\ q_2^2\ (-\alpha 1\ q_1^2 + \alpha 2\ q_1^2 + 2\ (q_1)_t^2) -$ $q_1\ q_2\ (q_1)_t\ (q_2)_t + \frac{1}{2}\ q_1^2\ (q_2)_t^2$

$$\left(\frac{q_1^4}{6} + q_1^2\, q_2^2\right)$$
$$(q_1)_t +$$
$$\frac{1}{2}\,(q_1)_t^3 -$$
$$\frac{1}{3}\,q_1^3\, q_2\,(q_2)_t$$
$$-\frac{1}{3}\,q_1^3\, q_2\,(q_1)_t +$$
$$\frac{1}{6}\,q_1^4\,(q_2)_t$$

$\alpha 1 \to 0$, $\alpha 2 \to 0$, $\Gamma 1 \to \frac{1}{3}$, $\Gamma 2 \to \frac{8}{3}$

$$-\frac{q_1^8}{72} - \frac{1}{18}\,q_1^4\,q_2^4 - \frac{1}{12}\,q_1^4\,(q_1)_t^2 - \frac{1}{8}\,(q_1)_t^4 - \frac{1}{18}\,q_1^2\,q_2^2\,(q_1^4 + 9\,(q_1)_t^2) + \frac{1}{3}\,q_1^3\,q_2\,(q_1)_t\,(q_2)_t - \frac{1}{12}\,q_1^4\,(q_2)_t^2$$

$$\left(\frac{\alpha 1\, q_1^2}{4} + \frac{q_1^4}{6} + q_1^2\, q_2^2\right)$$
$$(q_1)_t +$$
$$\frac{1}{2}\,(q_1)_t^3 -$$
$$\frac{1}{3}\,q_1^3\, q_2\,(q_2)_t$$
$$-\frac{1}{3}\,q_1^3\, q_2\,(q_1)_t +$$
$$\frac{1}{6}\,q_1^4\,(q_2)_t$$

$\alpha 1 \to \alpha 1$, $\alpha 2 \to 4\,\alpha 1$, $\Gamma 1 \to \frac{1}{3}$, $\Gamma 2 \to \frac{8}{3}$

$$-\frac{1}{32}\,\alpha 1^2\,q_1^4 - \frac{\alpha 1\, q_1^6}{24} - \frac{q_1^8}{72} - \frac{1}{18}\,q_1^4\,q_2^4 - \frac{1}{24}\,q_1^2\,(3\,\alpha 1 + 2\,q_1^2)\,(q_1)_t^2 - \frac{1}{8}\,(q_1)_t^4 - \frac{1}{36}\,q_1^2\,q_2^2\,(3\,\alpha 1\,q_1^2 + 2\,q_1^4 + 18\,(q_1)_t^2) + \frac{1}{3}\,q_1^3\,q_2\,(q_1)_t\,(q_2)_t - \frac{1}{12}\,q_1^4\,(q_2)_t^2$$

$$\frac{1}{6}\,q_2^4\,(q_1)_t -$$
$$\frac{1}{3}\,q_1\,q_2^3\,(q_2)_t$$
$$-\frac{1}{3}\,q_1\,q_2^3\,(q_1)_t +$$
$$\left(q_1^2\,q_2^2 + \frac{q_2^4}{6}\right)$$
$$(q_2)_t +$$
$$\frac{1}{2}\,(q_2)_t^3$$

$\alpha 1 \to 0$, $\alpha 2 \to 0$, $\Gamma 1 \to \frac{8}{3}$, $\Gamma 2 \to \frac{1}{3}$

$$-\frac{1}{18}\,q_1^2\,q_2^6 - \frac{q_2^8}{72} + \frac{1}{36}\,q_2^4\,(-2\,q_1^4 - 3\,(q_1)_t^2) + \frac{1}{3}\,q_1\,q_2^3\,(q_1)_t\,(q_2)_t - \frac{1}{12}\,q_2^2\,(6\,q_1^2 + q_2^2)\,(q_2)_t^2 - \frac{1}{8}\,(q_2)_t^4$$

$$\frac{1}{6}\,q_2^4\,(q_1)_t -$$
$$\frac{1}{3}\,q_1\,q_2^3\,(q_2)_t$$
$$-\frac{1}{3}\,q_1\,q_2^3\,(q_1)_t +$$
$$\left(\frac{\alpha 1\, q_2^2}{16} + q_1^2\,q_2^2 + \frac{q_2^4}{6}\right)$$
$$(q_2)_t +$$
$$\frac{1}{2}\,(q_2)_t^3$$

$\alpha 1 \to \alpha 1$, $\alpha 2 \to \frac{\alpha 1}{4}$, $\Gamma 1 \to \frac{8}{3}$, $\Gamma 2 \to \frac{1}{3}$

$$\frac{1}{288}\,(-3\,\alpha 1 - 16\,q_1^2)\,q_2^6 - \frac{q_2^8}{72} + \frac{1}{4608}\,(q_2^4\,(-9\,\alpha 1^2 - 96\,\alpha 1\,q_1^2 - 256\,q_1^4 - 384\,(q_1)_t^2)) + \frac{1}{3}\,q_1\,q_2^3\,(q_1)_t\,(q_2)_t - \frac{1}{96}\,q_2^2\,(3\,\alpha 1 + 48\,q_1^2 + 8\,q_2^2)\,(q_2)_t^2 - \frac{1}{8}\,(q_2)_t^4$$

$$\left(\frac{q_1^4}{6} + q_1^2\,q_2^2 + \frac{q_2^4}{6}\right)$$
$$(q_1)_t +$$
$$\frac{1}{2}\,(q_1)_t^3 +$$
$$\left(-\frac{1}{3}\,q_1^3\,q_2 - \frac{1}{3}\,q_1\,q_2^3\right)$$
$$(q_2)_t$$
$$\left(-\frac{1}{3}\,q_1^3\,q_2 - \frac{1}{3}\,q_1\,q_2^3\right)$$
$$(q_1)_t + \left(\frac{q_1^4}{6} + q_1^2\,q_2^2 + \frac{q_2^4}{6}\right)$$
$$(q_2)_t +$$
$$\frac{1}{2}\,(q_2)_t^3$$

$\alpha 1 \to 0$, $\alpha 2 \to 0$, $\Gamma 1 \to \frac{1}{3}$, $\Gamma 2 \to \frac{1}{3}$

$$-\frac{q_1^8}{72} - \frac{1}{18}\,q_1^2\,q_2^6 - \frac{q_2^8}{72} - \frac{1}{12}\,q_1^4\,(q_1)_t^2 - \frac{1}{8}\,(q_1)_t^4 + \frac{1}{36}\,q_2^4\,(-11\,q_1^4 - 3\,(q_1)_t^2) - \frac{1}{18}\,q_1^2\,q_2^2\,(q_1^4 + 9\,(q_1)_t^2) + \frac{1}{3}\,q_2\,(q_1^3 + q_1\,q_2^2)\,(q_1)_t\,(q_2)_t + \frac{1}{12}\,(-q_1^4 - 6\,q_1^2\,q_2^2 - q_2^4)\,(q_2)_t^2 - \frac{1}{8}\,(q_2)_t^4$$

$$\left(\frac{\alpha 1\, q_1^2}{4} + \frac{q_1^4}{6} + \frac{\alpha 1\, q_2^2}{4} + q_1^2\, q_2^2 + \frac{q_2^4}{6}\right)(q_1)_t + \frac{1}{2}(q_1)_t^3 + \left(-\frac{1}{4}\alpha 1\, q_1\, q_2 - \frac{1}{3} q_1^3\, q_2 - \frac{1}{3} q_1\, q_2^3\right)(q_2)_t$$

$$\alpha 1 \to \alpha 1$$
$$\alpha 2 \to \alpha 1$$
$$\Gamma 1 \to \frac{1}{3}$$
$$\Gamma 2 \to \frac{1}{3}$$

$$\left(-\frac{1}{4}\alpha 1\, q_1\, q_2 - \frac{1}{3} q_1^3\, q_2 - \frac{1}{3} q_1\, q_2^3\right)(q_1)_t + \left(\frac{\alpha 1\, q_1^2}{4} + \frac{q_1^4}{6} + \frac{\alpha 1\, q_2^2}{4} + q_1^2\, q_2^2 + \frac{q_2^4}{6}\right)(q_2)_t + \frac{1}{2}(q_2)_t^3$$

$$-\frac{1}{32}\alpha 1^2\, q_1^4 - \frac{\alpha 1\, q_1^6}{24} - \frac{q_1^8}{72} + \frac{1}{72}(-3\,\alpha 1 - 4\, q_1^2)\, q_2^6 - \frac{q_2^8}{72} - \frac{1}{24} q_1^2 (3\,\alpha 1 + 2\, q_1^2)(q_1)_t^2 - \frac{1}{8}(q_1)_t^4 + \frac{1}{288} q_2^4 (-9\,\alpha 1^2 - 36\,\alpha 1\, q_1^2 - 88\, q_1^4 - 24\,(q_1)_t^2) + \frac{1}{72} q_2^2 (-9\,\alpha 1\, q_1^4 - 4\, q_1^6 - 9\,\alpha 1\,(q_1)_t^2 - 36\, q_1^2\,(q_1)_t^2) + \frac{1}{12} q_1\, q_2\,(3\,\alpha 1 + 4\, q_1^2 + 4\, q_2^2)(q_1)_t (q_2)_t + \frac{1}{24}(-3\,\alpha 1\, q_1^2 - 2\, q_1^4 - 3\,\alpha 1\, q_2^2 - 12\, q_1^2\, q_2^2 - 2\, q_2^4)(q_2)_t^2 - \frac{1}{8}(q_2)_t^4$$

The result is a list containing 47 possible cases for finding integrals of motion. Scanning through the sub-lists, we realize that not only is the coupled system considered by Baecklund[] but also a decoupling of the equations is taken into account.

9.7.3 Two Ions in a Trap

Today, ion trapping is a fundamental experimental tool. Basic properties and elementary constants of physics are derivable from such experiments (cf. Wineland et al. [1983]). A standard trap for such experiments is, for example, a Paul trap (Paul and Steinwedel [1953]) in which ions are confined by a high-frequency *rf* field in connection with a static electric field. A Paul trap is completely free of any magnetic fields. To uncover the physical properties of a single ion, it is necessary to trap one or at most a small number of ions; otherwise, one gets a statistical mean of the observed property. In a typical trap experiment, two ions or a single ion are confined. Here, we shall consider the case in which two ions are stored in the trap. For two particles in the trap, we can describe the motion with Newton's equation. A schematic illustration of a trap follows:

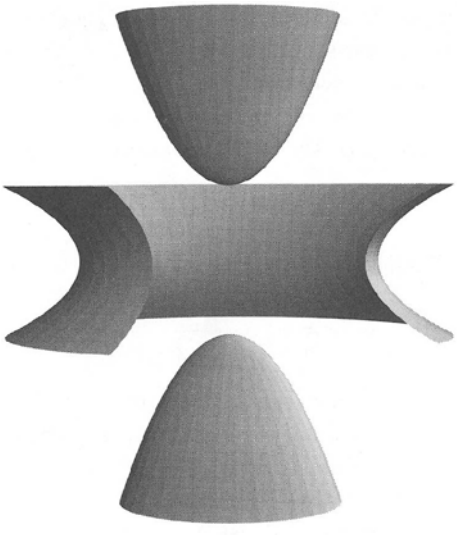


Figure 9.1

A Paul trap is designed by three hyperbolas of revolution, which are the electrodes. The inside diameter of the ring is $2\,r_0$ and the axial separation of the end caps is $2\,z_0$. The electrostatic potential

$$\phi = \frac{U_0}{r_0^2 + 2\,z_0^2}\,(r^2 - 2\,z^2), \tag{9.39}$$

where $r^2 = x^2 + y^2$ is created by the shape of the trap if the ring electrode is held at the dc potential U_0 with respect to the two end caps. This static part of the total potential confines ions along the z-axis. In the x and y direction, the motion is unbounded. The confinement in both of these directions is achieved by superimposing the static potential (9.39) by an *rf* potential

$$\phi^{\mathrm{rf}} = \frac{V_0 \cos(\Omega t)}{r_0^2 + 2\,z_0^2}\,(r^2 - 2\,z^2)\,. \tag{9.40}$$

In the case of two ions in the trap, an additional force influences the motion of the particles. This interacting force is due to the Coulomb repulsion of the ions. In the case of two identical particles, the Coulomb force is repulsive between particle 1 and particle 2. The force is given by

$$\vec{F}_{12}^{\,C} = \frac{q^2}{4\,\pi\epsilon_0}\,\frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3}, \tag{9.41}$$

where $\vec{x}_{1,2}$ denotes the position of particle 1 and 2, respectively, both carrying the charge q. In addition to these main forces, there are two other kinds of forces coming from the emission of photons and from the cooling process of the ions. These two forces, compared with the other three forces, are of minor importance in connection with the classical motion of the ions. So we neglect them in our further considerations. The classical equations of motion for two ions in a Paul trap now read

$$m\,\ddot{\vec{x}}_i = \vec{F}_i^{\,T} + \vec{F}_{ij}^{\,C} \quad \text{with } i,\ j = 1,\ 2 \text{ and } j \neq i. \tag{9.42}$$

In equation (9.42), $\vec{F}_i^{\,T}$ denotes the linear trap force

$$\vec{F}_i^{\,T} = -2\,q\,\frac{U_0 + V_0\cos(\Omega t)}{r_0^2 + 2\,z_0^2}\,(\vec{x}_i - 3\,z_i\,\vec{e}_z), \quad i = 1,\ 2, \tag{9.43}$$

where Ω is the external driving frequency. It is obvious that the equations of motion depend on time t. We can eliminate this time dependency by the averaging method of Landau and Lifshitz [1981]. Before we carry out this procedure, we introduce relative and center-of-mass coordinates by $\vec{x} = \vec{x}_1 - \vec{x}_2$ and $\vec{X} = \vec{x}_1 + \vec{x}_2$, respectively. The introduction of cylindrical coordinates (ρ, ζ) simplifies the equations of motion in the relative coordinates to

$$\ddot{\rho} + \omega_\rho^2\,\rho = \frac{\omega_\rho^2\,\rho}{\left(\sqrt{\rho^2 + \zeta^2}\right)^3} + \frac{l_z^2}{\rho^3}, \tag{9.44}$$

$$\ddot{\zeta} + \omega_\zeta^2\,\zeta = \frac{\omega_\rho^2\,\zeta}{\left(\sqrt{\rho^2 + \zeta^2}\right)^3}, \tag{9.45}$$

where the dots denote differentiation with respect to time t. The constants ω_ρ, ω_ζ, and l_z denote the secular frequencies in radial and axial directions and the angular momentum directed along the z-axis. The secular frequencies ω_ρ and ω_ζ are dimensionless quantities containing the ac and dc voltage, the mass m of the ions, the external driving frequency Ω, and the geometrical properties of the trap. For a detailed description of the derivation of these two equations, compare Baumann and Nonnenmacher [1992]. A further scaling of the dimensionless time τ by one of the secular frequencies ($\tau' = \omega_\rho\,\tau$) and introducing the ratios $\nu = l_z/\omega_\rho$ and $\lambda = \omega_\zeta/\omega_\rho$ gives us

$$\ddot{\rho} + \rho = \frac{\rho}{\left(\sqrt{\rho^2 + \zeta^2}\right)^3} + \frac{\nu^2}{\rho^3}, \tag{9.46}$$

$$\ddot{\zeta} + \lambda^2\,\zeta = \frac{\zeta}{\left(\sqrt{\rho^2 + \zeta^2}\right)^3}. \tag{9.47}$$

These two equations describe the relative ion motion in a Paul trap in the secular approximation. The two equations are derivable from the Lagrangian

```
LPaul = 1/2 ((∂τ ρ[τ])^2 + (∂τ ζ[τ])^2) -
  (1/2 (ρ[τ]^2 + λ^2 ζ[τ]^2) + 1/Sqrt[ρ[τ]^2 + ζ[τ]^2] + ν^2/(2 ρ[τ]^2));
LPaul // LieTraditionalForm
```

$$-\frac{\nu^2}{2\rho^2}+\frac{1}{2}\left(-\zeta^2\lambda^2-\rho^2\right)-\frac{1}{\sqrt{\zeta^2+\rho^2}}+\frac{1}{2}\left(\zeta_\tau^2+\rho_\tau^2\right)$$

describing in classical terms a particle with two degrees of freedom in an anharmonic potential. The equations of motion follow by applying the Euler derivative to the Lagrangian:

```
PaulTrapEquations = Thread[δ^τ_{ζ,ρ} [LPaul] == {0, 0}];
PaulTrapEquations // LTF
```

$$-\zeta\lambda^2+\frac{\zeta}{\left(\zeta^2+\rho^2\right)^{3/2}}-\zeta_{\tau,\tau}==0$$

$$\frac{\nu^2}{\rho^3}-\rho+\frac{\rho}{\left(\zeta^2+\rho^2\right)^{3/2}}-\rho_{\tau,\tau}==0$$

This system of equations is the starting point of our examination in connection with generalized symmetries. To detect the cases under which this second-order system of equation is analytically solvable, we have to reveal the parameter combinations ν and λ for which Noether's theorem is satisfied. We determine the integrals of motion by applying the above theory to the equations of motion. The function Baecklund[] serves as the main tool to derive the integrals of motion:

```
PaulIntegrals = Baecklund[PaulTrapEquations, {ζ, ρ}, {τ},
    {∂τ,τ ρ[τ], ∂τ,τ ζ[τ]}, 1, {ν, λ}, AnsatzPoly → {2, 1}];
  PaulIntegrals //
   TableForm[LieTraditionalForm[#]], TableHeadings ->
   {Automatic, {"Q1,2", "Parameters", "Integral"}},
   TableSpacing -> {1, 1}]&
```

	$Q_{1,2}$	Parameters	Integral
1	ζ_τ ρ_τ	$\nu \to \nu$ $\lambda \to -2$	$-2\zeta^2-\frac{\nu^2}{2\rho^2}-\frac{\rho^2}{2}-\frac{1}{\sqrt{\zeta^2+\rho^2}}-\frac{\zeta_\tau^2}{2}-\frac{\rho_\tau^2}{2}$
2	ζ_τ ρ_τ	$\nu \to \nu$ $\lambda \to 0$	$-\frac{\nu^2}{2\rho^2}-\frac{\rho^2}{2}-\frac{1}{\sqrt{\zeta^2+\rho^2}}-\frac{\zeta_\tau^2}{2}-\frac{\rho_\tau^2}{2}$
3	ζ_τ ρ_τ	$\nu \to \nu$ $\lambda \to 1$	$-\frac{\zeta^2}{2}-\frac{\nu^2}{2\rho^2}-\frac{\rho^2}{2}-\frac{1}{\sqrt{\zeta^2+\rho^2}}-\frac{\zeta_\tau^2}{2}-\frac{\rho_\tau^2}{2}$

4	ζ_τ ρ_τ	$\nu \to \nu$ $\lambda \to 2$	$-2\,\zeta^2 - \frac{\nu^2}{2\,\rho^2} - \frac{\rho^2}{2} - \frac{1}{\sqrt{\zeta^2+\rho^2}} - \frac{\zeta_\tau^2}{2} - \frac{\rho_\tau^2}{2}$
5	ζ_τ ρ_τ	$\nu \to \nu$ $\lambda \to \lambda$	$-\frac{1}{2}\,\zeta^2\,\lambda^2 - \frac{\nu^2}{2\,\rho^2} - \frac{\rho^2}{2} - \frac{1}{\sqrt{\zeta^2+\rho^2}} - \frac{\zeta_\tau^2}{2} - \frac{\rho_\tau^2}{2}$
6	$\rho\,\rho_\tau$ $\rho\,\zeta_\tau - 2\,\zeta\,\rho_\tau$	$\nu \to \nu$ $\lambda \to -2$	$\frac{\zeta\,\nu^2}{\rho^2} - \zeta\,\rho^2 + \frac{\zeta}{\sqrt{\zeta^2+\rho^2}} - \rho\,\zeta_\tau\,\rho_\tau + \zeta\,\rho_\tau^2$
7	$\rho\,\rho_\tau$ $\rho\,\zeta_\tau - 2\,\zeta\,\rho_\tau$	$\nu \to \nu$ $\lambda \to 2$	$\frac{\zeta\,\nu^2}{\rho^2} - \zeta\,\rho^2 + \frac{\zeta}{\sqrt{\zeta^2+\rho^2}} - \rho\,\zeta_\tau\,\rho_\tau + \zeta\,\rho_\tau^2$
8	$-\rho^2\,\zeta_\tau + \zeta\,\rho\,\rho_\tau$ $\zeta\,\rho\,\zeta_\tau - \zeta^2\,\rho_\tau$	$\nu \to \nu$ $\lambda \to 1$	$\frac{\zeta^2\,\nu^2}{2\,\rho^2} + \frac{1}{2}\,\rho^2\,\zeta_\tau^2 - \zeta\,\rho\,\zeta_\tau\,\rho_\tau + \frac{1}{2}\,\zeta^2\,\rho_\tau^2$

The result contains eight cases for which Baecklund[] found integrals. The assumption of the calculation was that the characteristics are polynomials in the dependent variables and their derivatives.

Among the obtained integrals are the total energy of the Paul trap, the angular momentum and its generalization, and integrals which are related to the Runge-Lenz vector. In all cases for which generalized symmetries are known, we find that the two ions are confined to a two-dimensional surface in phase space. These two integrals are the total energy plus a second constant of motion. The existence of two integrals in a phase space with two dimensions of freedom is sufficient to determine a regular motion and are thus integrable (cf. Tabor [1989]).

The integrability of the equation of motion is, in fact, closely related to the ratios of the two secular frequencies ω_ρ and ω_z. We find that integrals exist for equal frequencies ($\lambda = \pm 1$) and that the axial frequency is twice as large as the radial frequency ($\lambda = \pm 2$), independent of the value of ν. We note that the case with $\lambda = \pm 2$ is related to the Runge-Lenz vector known from Kepler's problem in classical mechanics.

So far, we illustrated the application of Baecklund[] for different types of differential equations. We demonstrated that the function is capable of finding the generalized symmetries of PDEs. In the case of second-order ODEs the generalized symmetries are beneficial, in connection with Noether's theorem, for constructing solutions at least in implicit form. The existence of a sufficient number of integrals of motion excludes the occurrence of chaotic motion.

10

Solution of Coupled Linear Partial Differential Equations

10.1. Introduction

In this chapter, we discuss the main steps for solving systems of coupled linear partial differential equations (PDEs). Such linear PDEs are the result of the invariance conditions discussed in Chapter 5 on point symmetries, in Chapter 7 on potential symmetries, in Chapter 8 on approximate symmetries, and in Chapter 9 on generalized symmetries. Especially for these types of symmetries, the following procedures are very successful. The main topic here is the automatic derivation of solutions. This self-governed method is the basis for an efficient calculation of symmetries by computer algebra programs.

The ideas behind this automatic procedure goes back to works of Riquier [1910] and Janet [1920]. Their basic ideas for solving differential equations are extended by a few heuristics. The incorporation of heuristics into solution procedures simplifies the task, as Kamke [1977] and Schwarz [1992] remark. However, the use of heuristic solution steps by itself are not sufficient to succeed. The combination of a regular procedure and heuristics establish a method which is successful for a very large number of linear PDEs. Our experience is that the combination of a differential Groebner technique with heuristic steps is very successful in solving linear PDEs.

The central steps for finding solutions of a linear overdetermined system of PDEs consists of the following:

1. The decoupling of the equations by a completion algorithm.
2. The integration of simple equations.
3. The simplification of the equations.

These three steps are essential in the solution procedure. Since the PDEs are, in general, coupled in a non-trivial form, it is recommended to disconnect the equations. This disjoining of differential equations results into a *general canonical form*. The term general canonical form was introduced by Janet and Riquier at the beginning of the 20th century. These authors developed a procedure to create an equivalent and simplified representation of a given system of PDEs. Thus, our discussion is not only restricted to heuristic solution steps for PDEs but also includes a step of standardization.

To keep the integration procedure for PDEs as simple as possible, we first calculate the general canonical form to find a simpler representation of the equations. The first section of this chapter discusses the theory of the general canonical algorithm due to Janet and Riquier. The second section describes the heuristic integration steps to solve the simplified equations.

10.2. General Canonical Form of PDEs

In this section, we will describe a method originally given by Riquier and Janet which they called *forme canonique généralé* (general canonical form) and later discussed by Thomas [1929, 1934] in the investigation of coupled differential equations. The present implementation in *Mathematica* uses a Groebner basis algorithm originally designed to solve polynomial equations of higher order (Buchberger [1985]) adapted and generalized for differential equations. The aim of our Groebner algorithm for differential equations is to transform a given system of equations to an equivalent decoupled representation. Before we describe our procedure to derive a general canonical form, let us define what we understand by a canonical form.

Definition: Canonical form

A general canonical form of a system of partial differential equations is a simplified form of a system in such a representation that all Schwarzian integrability conditions are satisfied. It is essential that the solution manifold under such a transformation is conserved. ○

This verbal definition states the essential property of a general canonical form simplifying the original equations but does not clarify how such a representation of equations can be calculated.

In practical applications, it is necessary to have a constructive definition available. However, the above definition does not explain how a general canonical form may be calculated and thus gives room for several algorithms. In fact, there exist several more or less different procedures in the literature approaching the definition of a general canonical form (cf. Schwarz [1985], Reid [1993], Mansfield [1992], Carrà-Ferro [1993]).

In the following, we will discuss five steps which are sufficient to calculate a general canonical form. One of these basic steps uses the knowledge that all equations involved can be solved for a certain derivative which is called the leading derivative, meaning that a leading derivative can be identified which may depend on other independent derivatives. We further assume that the derivatives are unique. Uniqueness of the derivatives suggests that independent derivatives cannot be represented by other derivatives in the system. Again, we realize that derivatives are the basic tools of our calculations.

After these preliminaries, let us state the five sufficient properties of a general canonical form:

1. All equations are solved in such a way that the leading derivatives occur only on the left-hand side of the equations.

2. In a canonical representation of equations, it is impossible that the same derivatives occur on the left-hand side and on the right-hand side of the equation.

3. All derivatives of the left-hand side of the equations are disjunct.

4. There exists no derivative which is a non-trivial derivative of any derivative occurring on the left-hand side of the equations.

5. All the Schwarzian integrability conditions are satisfied.

The condition that a leading derivative exists will require that we have an appropriate measure for detecting it as a leading derivative. An appropriate tool for this decision is the order of a differential expression. All of the remaining four steps are based on the ordering idea of derivatives. There are many ordering schemes suited; all have various advantages for different purposes. For example, a lexicographic ordering yields elimination ideals, while a total degree ordering is used to obtain the integrability conditions. The total degree ordering is useful in getting the initial data

for convergent formal power series solutions. In general, the ordering scheme used, denoted here by ord(), can be chosen arbitrarily. However, the essential point is that we are consistent in its application. Following Janet, this means that an arbitrary differential operator ∂ has to satisfy the following:

(i) $\mathrm{ord}(\partial u) > \mathrm{ord}(u)$,

(ii) $\mathrm{ord}(u^1) > \mathrm{ord}(u^2) \longrightarrow \mathrm{ord}(\partial u^1) > \mathrm{ord}(\partial u^2)$.

These two rules summarize the idea that the order of a differential equation is always greater than the order of an ordinary function and that the lexicographic ordering has a direct consequence on the ordering of derivatives. In our algorithm, we realized these concept by an ordering scheme which satisfies the following properties:

(a) Total ordering of the derivatives

$$\mathrm{ord}\,(u_J^\alpha) > \mathrm{ord}\,(u_K^\alpha) \quad \text{if } \#J > \#K$$

where $\#J$ denotes the number of J.

(b) Lexicographic ordering of the derivatives

$$\mathrm{ord}\,(\partial_{x_i} u^\alpha) > \mathrm{ord}\left(\partial_{x_j} u^\alpha\right) \text{ if } i > j.$$

(c) Lexicographic ordering of function names

$$\mathrm{ord}\,(u^\alpha) > \mathrm{ord}\,(u^\beta) \text{ if } \alpha > \beta.$$

We note that the ordering scheme used has an influence on the representation of the equations. However, it does not change the solution manifold of the equations. Calculating the general canonical form of a given system of equations, we only operate on equivalence relations. This guarantees that the solution manifold is not changed.

The first step in calculating a general canonical form is thus the solution of the equations with respect to the highest derivative. The solution is used to eliminate all occurrences of the highest derivatives in the remaining equations. Thus, the procedure is based on identifying an equation containing the highest derivative. The remaining equations can then be treated by the above ordering scheme. It is also crucial to insert the solution for the highest derivatives into the remaining equations. If we carry out the steps accurately, we can satisfy conditions 1, 2, and 3 of the canonical algorithm.

If we have solved all equations for the highest derivatives, we next implicitly substitute the derivatives; i.e., we replace all non-trivial derivatives by the derivatives of the corresponding terms. At this point, it may happen that not all derivatives of the left-hand side are different from each other. If this happens, we have to go back to the first step of our algorithm and repeat the calculation again. The whole procedure is repeated as long a conditions 1 to 4 are satisfied.

If we can satisfy the first four steps of the general canonical form algorithm, we can proceed to calculate the integrability conditions for the complete system. The integrability conditions serve to terminate the algorithm. If we cannot find new integrability conditions, we say that the original system of PDEs is, in general, a canonical form.

The explicit calculation of the integrability conditions works in the following way. We differentiate two of the equations containing different derivatives of the same function on the left-hand side. If we do the differentiation on both sides in an appropriate way, we create two identical expressions on the left-hand side. Subtraction of the two equations results in a new integrability condition. If we find such a relation, we start again with step 1 of the general canonical algorithm. The last step to find the integrability condition can be mathematically formulated as

$$u_J^\alpha = G(x_i, u^\beta, u_L^\beta), \tag{10.1}$$

$$u_K^\alpha = H(x_i, u^\beta, u_L^\beta), \tag{10.2}$$

where $L = \max(J, K)$ and

$$0 = u_L^\alpha - u_K^\alpha = G_{L-J} - H_{L-K} . \tag{10.3}$$

Using this procedure, we are able to simplify all types of partial differential equations, both linear and non-linear. In practical situations, especially for non-linear PDEs, there is a little restriction for the termination of the algorithm. This restriction is connected with the unique solution of the non-linear equation. For example, we all know, as does *Mathematica*, that the general solution of a quadratic polynomial consists of two solutions. Thus, the program has to know what branch of solution it should take next. In a general case, we have to deal with more complicated solution branches than just a bifurcation. So, if we are unable to uniquely handle the set of solutions for non-linear equations in the highest derivative, we cannot derive a general canonical form of the original equations.Thus, the algorithm for calculating the general canonical form bifurcates to an interactive version if a set of solutions for the highest derivatives exists. So, the occurrence of more than one solution will force us to examine different representations of the canonical form.

10.2.1 Application of the General Canonical Form Algorithm

The following examples demonstrate the application of the general canonical form algorithm. As a first step, we recall how the original Groebner basis algorithm for polynomials works and how this algorithm can be used to solve PDEs. Groebner bases appear in many modern algebraic algorithms and applications. In *Mathematica*, the function GroebnerBasis[] takes a set of polynomials and reduces this set to a canonical form from which many properties can be deduced. An important feature is that the set of polynomials obtained always has exactly the same collection of common roots as the original set. Thus, a Groebner basis is useful if one is interested in the solution of a system of polynomials. The solutions, if they exist, are the invariants of the representations. Groebner bases were first introduced in the mid-1960s by Hironaka (who called them "standard bases") and, independently, later by Buchberger in his Ph.D. thesis. The name *Groebner bases* was coined by Buchberger to honor his thesis adviser W. Groebner. The Groebner basis representation has the merit that it is much easier to solve than the original set of polynomials. We can demonstrate this behavior by the following example:

```
poly1 = x^2 y + y + 4
```

$$4 + y + x^2\, y$$

```
poly2 = x y^2 + 1
```

$$1 + x\, y^2$$

The Groebner basis of this set of polynomials is determined in *Mathematica* by

```
gpoly = GroebnerBasis[{poly1, poly2}, {x, y}]
```

$$\{1 + 4\, y^3 + y^4,\ x - 4\, y - y^2\}$$

The result shows that the second-order set of polynomials can be represented by a fourth-order polynomial in y and a linear relation in x. The solution of these two polynomials are

```
Solve[Thread[gpoly == {0, 0}], {x, y}] // N
```

```
{{x → -2.23012, y → -0.669632}, {x → -0.0629971, y → -3.98419},
 {x → 1.14656 - 2.409 I, y → 0.32691 - 0.51764 I},
 {x → 1.14656 + 2.409 I, y → 0.32691 + 0.51764 I}}
```

According to the fourth order of the first polynomial, we find two real distinct solutions and two solutions which are complex conjugate to each other. The direct solution of the coupled system delivers the same result:

```
Solve[{poly1 == 0, poly2 == 0}, {x, y}] // N
```

```
{{x → -2.23012, y → -0.669632}, {x → -0.0629971, y → -3.98419},
 {x → 1.14656 - 2.409 I, y → 0.32691 - 0.51764 I},
 {x → 1.14656 + 2.409 I, y → 0.32691 + 0.51764 I}}
```

Another example, which at first glance, is very similar to the example examined above starts from the two polynomials

```
poly1 = x^2 y + y + 4
```

$$4 + y + x^2\, y$$

```
poly2 = x y^2
```

$$x\, y^2$$

The Groebner basis of the polynomials is given by

```
gbasis = GroebnerBasis[{poly1, poly2}, {x, y}]
```

```
{4 + y, x}
```

The solution of these polynomials follow by solving the basis

```
Solve[Thread[gbasis == {0, 0}], {x, y}]
```

```
{{x → 0, y → -4}}
```

The direct solution of the set of equations yields

```
Solve[{poly1 == 0, poly2 == 0}, {x, y}]
```

```
{{y → -4, x → 0}}
```

the same result. To understand what goes on behind the function GroebnerBasis[], we repeat the calculation for the last example a second time by hand.

The Groebner basis technique to solve sets of polynomials works manually as follows. The present equation *poly*1 and *poly*2 are decoupled by multiplying the polynomials with appropriate variables and adding the result to another polynomial. The starting equations are

$$2x^2y + y + 4 = 0, \tag{10.4}$$

$$xy^2 = 0. \tag{10.5}$$

Our aim is to separate both equations. We first multiply the first equation by the factor $-y$ and the second by the factor $2x$. After the multiplication, we add the results and find

$$2x^2y + y + 4 = 0, \tag{10.6}$$

$$xy^2 = 0, \tag{10.7}$$

$$y^2 + 4y = 0 \longrightarrow y^2 = -4y; \tag{10.8}$$

substituting the result of (10.8) into equation (10.7) and eliminating the factor 4 by division, we get

$$2x^2y + y + 4 = 0, \tag{10.9}$$

$$xy = 0, \tag{10.10}$$

$$y^2 + 4y = 0. \tag{10.11}$$

If we use equation (10.10) in equation (10.9), we obtain a relation which allows us to determine the solution for y:

$$y + 4 = 0 \longrightarrow y = -4, \tag{10.12}$$

$$xy = 0, \tag{10.13}$$

$$y^2 + 4y = 0. \tag{10.14}$$

The result from equation (10.12) can be used in the remaining two equations, (10.13) and (10.14), satisfying the third equation identically and delivering the second solution for $x = 0$. Thus, we get the complete solution of the original set of polynomials to be

$$y = -4, \tag{10.15}$$

$$x = 0. \tag{10.16}$$

We easily can check that this is the complete solution. Using equation (10.13), we observe that either x or y has to vanish. Setting, in a first attempt, $y = 0$, we observe a contradiction in the first equation, i.e., $4 = 0$. The other choice, $x = 0$, will result in the same result derived by the manipulations of the equations. The method of manipulating the set of polynomials is identical to the algorithm used in the calculation of the Groebner basis. A detailed description of the Groebner algorithmic procedure is given by Buchberger [1985].

Our aim is to use the Groebner basis technique to solve partial differential equations transformed to a canonical form. The first step for handling this problem is to decouple the system of partial differential equations. Using the Buchberger algorithm to solve this problem, we need a scheme to translate the operations from a differential representation to a polynomial representation. The rule to convert differential terms to polynomials goes as follows: Use the differentiation order and the variable to represent the related polynomial. Reading this transformation in reverse, we are able to transform a polynomial into a differential equation. To demonstrate this transformation, let us consider the example of polynomials examined in the previous example. We also have to introduce a function F depending on the two independent variables x and y. The differential analogue of equations (10.4) and (10.5) read

$$2\,\partial_{xxy}\,F(x,y) + \partial_y\,F(x,y) + 4\,F(x,y) = 0, \tag{10.17}$$

$$\partial_{xyy}\,F(x,y) = 0. \tag{10.18}$$

The question now is: How can we benefit from the Buchberger method of solving polynomials in simplifying this formally equivalent system of PDEs?

The answer of this question is that we have to use the same algorithm as for polynomials but now with the interpretation of our transformation rule that the variables x and y are replaced by partial derivatives. Using this interpretation, we are able to decouple the above system of PDEs. The following calculation will demonstrate this. First we apply to equation (10.17) the partial derivative with respect to y (∂_y). In a second step, we apply the operator $-2\,\partial_x$ to equation (10.18). Adding the results, we find three relations

$$2\,\partial_{xxy}\,F(x,y) + \partial_y\,F(x,y) + 4\,F(x,y) = 0, \tag{10.19}$$

$$\partial_{xyy}\,F(x,y) = 0, \tag{10.20}$$

$$\partial_{yy}\,F(x,y) + 4\,\partial_y\,F(x,y) = 0. \tag{10.21}$$

The last equation provides

$$\longrightarrow \partial_{yy}\,F(x,y) = -4\,\partial_y\,F(x,y). \tag{10.22}$$

Substituting the result (10.22) into equation (10.20) the PDEs become

$$2\,\partial_{xxy}\,F(x,y) + \partial_y\,F(x,y) + 4\,F(x,y) = 0, \tag{10.23}$$

$$\partial_{xy}\,F(x,y) = 0, \tag{10.24}$$

$$\partial_{yy}\,F(x,y) + 4\,\partial_y\,F(x,y) = 0. \tag{10.25}$$

Using relation (10.24) of this system, we are able to write (10.23) as

$$\partial_y F(x, y) + 4 F(x, y) = 0 \longrightarrow \partial_y F(x, y) = -4 F(x, y), \tag{10.26}$$

$$\partial_{xy} F(x, y) = 0, \tag{10.27}$$

$$\partial_{yy} F(x, y) + 4 \partial_y F(x, y) = 0. \tag{10.28}$$

The result from relation (10.26) can be used to satisfy equation (10.28). Substituting expression (10.26) into equation (10.27), we automatically decouple the differentiations. At the end, we get a decoupled system of PDEs which is equivalent to the original system (10.17) and (10.18):

$$\partial_y F(x, y) = -4 F(x, y), \tag{10.29}$$

$$\partial_x F(x, y) = 0. \tag{10.30}$$

The solution of this simplified system of equations is

$$F(y) = F_0 e^{-4y}. \tag{10.31}$$

It is a simple task to verify that the original equations are also satisfied by this solution. Thus, we conclude that the canonical representation of the PDEs allows us to derive the solution by a simple integration. In fact, the solution also satisfies the original equation. All these steps carried out so far by hand can be handled by *Mathematica*. To show how this calculation works, we will first define the transformation of polynomials in two variables to a differential representation. The transformation rule given is a very simple version of transformations for the products of two variables and for the variables itself. If the polynomial contains numeric factors, they will be transformed to the dependent function. We note that the given rules are only usable for handling our special problem but are not designed to allow general transformations. The simple transformation to a differential representation has the form

```
toDifferentials = {x^n_. y^m_. -> Derivative[n, m][F][x, y],
        y -> Derivative[0, 1][F][x, y],
        x -> Derivative[1, 0][F][x, y],
        a_ /; NumberQ[a] -> a F[x, y]}
```

```
{x^n_. y^m_. → F^(n,m)[x, y], y → F^(0,1)[x, y], x → F^(1,0)[x, y],
 a_ /; NumberQ[a] → a F[x, y]}
```

The application of this transformation rule to the two polynomials poly1 and poly2 gives us the representation of the linear second-order PDEs:

```
pde1 = poly1 /. toDifferentials
```

$$4\,F[x, y] + F^{(0,1)}[x, y] + F^{(2,1)}[x, y]$$

```
pde2 = poly2 /. toDifferentials
```

$$F^{(1,2)}[x, y]$$

The more interesting case in connection with our aim to solve PDEs is the case considering the inverse transformation from a differential representation to a polynomial one. A simple version for this reduction to a polynomial can be defined as follows:

```
backToPolynomials = {b_. Derivative[m___][F][x___] :>
          b Apply[Times, {x}^{m}],
        F[___] -> 1}
```

$$\{b\_.\ F^{(m\_\_\_)}[x\_\_\_] :\to b\,\text{Times}@@\{x\}^{\{m\}},\ F[\_\_\_] \to 1\}$$

We note that the given transformation rule is sufficient to handle one dependent variable only. The application of our transformation gives us

```
pol1 = pde1 /. backToPolynomials
```

$$4 + y + x^2\,y$$

```
pol2 = pde2 /. backToPolynomials
```

$$x\,y^2$$

These two polynomials are just the polynomials from which we started. We know that the Groebner basis algorithm allows us a simplified representation of this set by

```
gbas = GroebnerBasis[{pol1, pol2}, {x, y}]
```

$$\{4 + y,\ x\}$$

Applying the transformation rule to a differential representation, we find a linear first-order system for the function F:

```
pdegbasis = gbas /. toDifferentials
```

$$\{4\,F[x, y] + F^{(0,1)}[x, y],\ F^{(1,0)}[x, y]\}$$

Now, using the capabilities of *Mathematica* to solve this system of first-order partial differential equations, we end up with the solution. Rewriting the PDEs in a standard form, we can use them in the function DSolve[]:

```
pdes = Thread[pdegbasis == {0, 0}];
pdes // LieTraditionalForm // TableForm
```

$$4\,F + F_y == 0$$
$$F_x == 0$$

The solution of the first equation shows that the y coordinate decays exponentially, while the x coordinate is included in an arbitrary function denoted by $C[1][x]$:

```
s1 = DSolve[pdes[[1]], F[x, y], {x, y}]
```

```
{{F[x, y] → E^(-4 y) C[1][x]}}
```

The solution of the second equation shows us that F is just a function of y:

```
s2 = DSolve[pdes[[2]], F[x, y], {x, y}]
```

```
{{F[x, y] → C[1][y]}}
```

Thus, the complete solution of this problem is given by *s1* if we set the arbitrary function $C[1][x]$ equal to a constant $C[1]$:

```
sol = s1 /. C[1][x] -> C[1]
```

```
{{F[x, y] → E^(-4 y) C[1]}}
```

Knowing the solution, we are able to check whether the original equations *pde1* and *pde2* are satisfied. To make this check as simple as possible, we convert the solution of F to a pure function:

```
solution = F -> Function[{x, y}, F[x, y]] /. sol[[1, 1]]
```

```
F → Function[{x, y}, E^(-4 y) C[1]]
```

Having this representation of the solution available, it is easy to verify that the original PDEs are satisfied. The only thing that remains is to replace the dependent variable F by the solution:

```
{pde1, pde2} /. solution
```

```
{0, 0}
```

Both checks show that the derived solution, in fact, solves the original equations.

As discussed above in the interactive calculation, a general canonical form follows from the application of a differential Groebner basis algorithm. The package *MathLie* contains a function GeneralCanonicalForm[] supporting this kind of calculation. The

function contains a generalization of the simple steps presented in the above discussion. To demonstrate the automatic calculation of the general canonical form as an example, let us recall the manipulations done by hand. All the following steps are implemented in the function GeneralCanonicalForm[], allowing the derivation of a canonical representation.

First, let us repeat the ordering for two functions $\xi^1 = \xi^1(x, y)$ and $\xi^2 = \xi^2(x, y)$. The ordering is defined by

$$\text{ord}(\partial_x \xi^1) > \text{ord}(\partial_x \xi^2) > \text{ord}(\partial_y \xi^1) > \text{ord}(\partial_y \xi^2) > \text{ord}(\xi^1) > \text{ord}(\xi^2). \quad (10.32)$$

The ordering used gives the highest derivative the highest priority in the calculation. To see the consequences of the ordering, let us consider the simple system

$$0 = \xi^1_x + \xi^1_y - \xi^2, \quad (10.33)$$

$$0 = (\xi^1)^2_y + 2\xi^2 \xi^1_x - (\xi^2)^2, \quad (10.34)$$

where $\xi^1 = \xi^1(x, y)$ and $\xi^2 = \xi^2(x, y)$ are functions of the independent variables x and y, respectively. Applying the general canonical form algorithm, we first have to solve the equations with respect to the leading derivative which results to

$$\xi^1_x = -\xi^1_y + \xi^2. \quad (10.35)$$

Substituting this result into the second equation gives us

$$0 = (\xi^1_y)^2 + 2\xi^2 \left(-\xi^1_y + \xi^2\right) - (\xi^2)^2, \quad (10.36)$$

$$0 = (\xi^1_y - \xi^2)^2. \quad (10.37)$$

The result of these manipulations is that $\xi^1_y = \xi^2$. Going back to our second equation of the original system and solving this equation with respect to the leading derivative ξ^1_x, we find

$$\xi^1_x = \frac{1}{2}\xi^2 - \frac{(\xi^1)^2_y}{2\xi^2} = \frac{\xi^2}{2} - \frac{\xi^2}{2} = 0. \quad (10.38)$$

To derive this result, we used the relation $\xi^1_y = \xi^2$. At this stage of our calculation, we know two relations connecting ξ^1 and ξ^2 by

$$\xi^1_x = 0, \quad (10.39)$$

$$\xi^1_y = \xi^2. \quad (10.40)$$

Since these two equations are represented by two differential equations with unique derivatives on the left-hand side, we do not need to use an implicit substitution step. Thus, we can skip step 3 and go to step 4 of the algorithm and calculate the integrability condition by

$$0 = \xi^1_{xy} - \xi^1_{xy} = \xi^2_x \,. \tag{10.41}$$

Since the Schwarzian integrability conditions are necessary conditions for deriving the general canonical form, we have to assume that the resulting relation for ξ^2 is an additional equation determining the solutions for ξ^1 and ξ^2. Thus, we have to add relation (10.41) to equations (10.39) and (10.40). The complete general canonical form thus reads

$$\xi^1_x = 0, \tag{10.42}$$

$$\xi^1_y - \xi^2 = 0, \tag{10.43}$$

and

$$\xi^2_x = 0. \tag{10.44}$$

To check the function GeneralCanonicalForm[], we examine the same equations as above. The function GeneralCanonicalForm[] calculates the related general canonical form for equations expressed in the dependent variables *xi*[*i*] and *phi*[*i*]. The field functions *xi*[*i*] and *phi*[*i*] can depend on any number of independent variables. The function GeneralCanonicalForm[] possesses two options allowing us to control the printing on the screen. With the option *TraceStep→True*, gives information on the steps carried out by the function. The option *WarningSS→True* we get information on factors eliminated during the calculation. The on-line help text informs us about the capabilities of the function:

```
? GeneralCanonicalForm

GeneralCanonicalForm[equations_List] calculates the
  general canonical form for a given list of coupled
  partial differential equations.
```

Our example above needs the following input:

```
GeneralCanonicalForm[{∂x xi[1][x, y] + ∂y xi[1][x, y] -
      xi[2][x, y],
(∂y xi[1][x, y])^2 + 2 xi[2][x, y] ∂x xi[1][x, y] -
xi[2][x, y]^2}, TraceStep -> True] //
 TableForm[LieTraditionalForm[Map[# == 0&, #]]]&
```

```
C-1 : explicit substitutions of leading derivatives

C-2 : implicit substitutions of leading derivatives

C-3 : determining minimal integrability conditions

C-1 : explicit substitutions of leading derivatives

C-2 : implicit substitutions of leading derivatives
```

$$-\xi_2 + (\xi_1)_y == 0$$
$$(\xi_1)_x == 0$$
$$(\xi_2)_x == 0$$

The derived canonical system is the same as above and contains all the solutions of the original equations. We can show the equivalence of the solution manifold by solving the general canonical form representation. The use of the solution in the original equations shows us the equivalence of both representations. The first and last equations of the canonical form imply that

$$\xi^1 = \xi^1(y) \tag{10.45}$$

and

$$\xi^2 = \xi^2(y). \tag{10.46}$$

This result shows that the dependent functions are only functions of y. Knowing this fact, we are able to satisfy both original equations if we take into account the second relation of our canonical representation of the equations.

10.3. Solution of Linear PDEs

In this section, we discuss heuristic procedures to integrate simple differential equations. The term *simple equations* means that we are interested in equations containing, for example, only one element so-called monomials, or ordinary differential equations, and exact PDEs. Applying one of the integration steps successfully to such equations, we will find an explicit solution or at least some functional dependencies of the solution. Consequently, we can use this information to transform or simplify some or all of the equations. We will discuss heuristic procedures like the integration of monomials, the solution of ODEs and pseudo-ODEs, the determination of integrating factors for exact PDEs, and the derivation of a potential representation for some types of PDE. Solving equations is intriguingly connected with the simplification of the intermediate results. Thus, we will also discuss different steps of simplification.

10.3.1 Integration of Monomials

The simplest type of equation which may appear in a system of PDEs is a monomial with the general representation

$$\frac{\partial^{i_1 \ldots i_n} f(x_1, \ldots x_n)}{\partial x_1^{i_1} \ldots \partial x_n^{i_n}} = 0, \tag{10.47}$$

where $i_1, \ldots i_n$ are integers denoting the order of differentiation with respect to the independent variables $x_1, \ldots, x_n$. Using the terminology of Riquier and Janet, we call this type of equation a monomial. Integrating the monomial with respect to the independent variable x_k, we obtain the general solution

$$f(x_1, \ldots, x_n) = \sum_{k=1}^{n} \sum_{j=0}^{i_k - 1} c_{jk}(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)\, x_k^j \tag{10.48}$$

with c_{jk} arbitrary functions of the independent variables x_i. Note that the functions c_{jk} do not depend on x_k. This result is useful if we try to simplify the remaining equations of our system.

The simple integration of a monomial with respect to a variable x_k is a common method used in pencil calculations and is widely discussed in the standard texts like Kamke [1977] or Ince [1956]. In the following sections, we will describe other procedures which are also effective in pencil as well as in computer-based calculations.

Let us consider a simple example to illustrate the integration step discussed. Assume $f = f(x, y)$ is a sufficiently smooth function which satisfies the PDE

$$\partial_x f(x, y) = 0. \tag{10.49}$$

The solution of this equation follows by a partial integration with respect to the independent variable x. In the integration step, we have to assume that y is independent of x, which is trivial if we consider x and y as independent variables. Assuming this, we can write down the solution for f as

$$f(x, y) = g(y)\, x + h(y), \tag{10.50}$$

where $g(y)$ and $h(y)$ are arbitrary functions in y. These two functions are especially independent of x. Integrating the PDE (10.49), we introduced two arbitrary functions depending only on the variables of the intersection of the independent variables and the variable with respect the differentiation is carried out. This set-theoretic interpretation is crucial when implementing this step in *Mathematica*.

10.3.2 Integrating ODEs and Pseudo-ODEs

Ordinary differential equations (ODEs) have a distinguished role in mathematics. We know that for certain types of ODEs, there exists at least particular solutions. Since *Mathematica* has implemented a large number of procedures solving ordinary differential equations, we will use them for our purposes. Knowing a solution of an ODE will help us to solve partial differential equations. Commonly, differential equations with one independent variable are called ODEs. However, dealing with PDEs, we can generalize the term ODE to pseudo-ODE. A pseudo-ODE is defined by the following type of relation:

$$\partial_{x_i} F(x_1, x_2, \ldots) + a_1(x_1, x_2, \ldots)\partial_{x_i} F(x_1, x_2, \ldots) + \cdots = 0. \tag{10.51}$$

This kind of equation is called a pseudo-ODE because there only occur derivatives of the field F with respect to one independent variable x_i. So the changes of the function F are restricted on the single coordinate x_i. The other independent variables, x_k, act as parameters or pseudo-constants. Under this condition, it is obvious that (10.51) behaves like an ODE. So we can treat equation (10.51) as an pseudo-ODE. With this interpretation, we are able to solve (10.51) by taking into account that the integration constants are functions of the independent variables except x_i. The notion of a pseudo-ODE allows us to solve special types of PDEs occurring frequently in the determining systems in symmetry analysis.

This procedure allows us to determine the explicit dependence of the function F on the single variable x_i. Knowing this kind of solution, we are able to simplify some of the remaining equations in the system of determining equations.

10.3.3 Integrating Exact PDEs

From the theory of ordinary differential equations, we know that a distinguished type of equation is an exact ODE. This type of ODE allows us to write down the solution if we know an integrating factor of the equation. In Chapter 4, we discussed the generalization of the term-integrating factor. However, this factor follows from a complete algorithmic procedure using only the ODE itself. The idea is to use the results for ordinary differential equations again to solve some types of partial differential equation.

In case of PDEs, we call a differential equation $\Delta = 0$ *exact* if we are able to write the differential expression Δ as a total derivative of an integral I with respect to a single variable x_i. Consequently, we have

$$\Delta = \frac{dI}{dx_i}. \tag{10.52}$$

Integrating this relation automatically, we need a function calculating the integral I of the right-hand side. If we know the integral I, we can replace the equation $\Delta = 0$ by the relation $I = \text{const}$. It is important to note that we do not lose any information by this replacement. However, using a procedure originally designed for ODEs, we have to take into account that the integration constants are again not constants in the original sense but depend on all independent variables occurring in Δ except x_i.

10.3.4 Potential Representation

Closely related to the integration of exact PDEs is the procedure to find a potential representation of the original equations, as Wolf [1991] remarks. Let us discuss the procedure by considering an equation of the form

$$\partial_{x_1} I_1 + \partial_{x_2} I_2 = 0, \tag{10.53}$$

where I_1 and I_2 are functions of x_1 and x_2. Relation (10.53) can be integrated by introducing a potential $V = V(x_1, x_2)$ satisfying the properties

$$I_1 = \partial_{x_2} V \quad \text{and} \quad I_2 = -\partial_{x_1} V. \tag{10.54}$$

The connection between the potential representation with V and the concept of an exact PDE can be shown if we take into account that the left-hand side of our first equation is exact with respect to x_1 and the right-hand side is exact with respect to x_2. This observation is the basis of the algorithmic calculation. Thus, we can use the following six steps to check the existence of a potential representation. As the input quantity, we use the differential expression Δ.

1. Integrate Δ with respect to x_1 and call the exact part of this result I_1.
2. Calculate the residue R by $R = \Delta - \partial_{x_1} I_1$.
3. If $R = 0$, then Δ is exact and the procedure terminates.

Otherwise:

4. Determine I_2 as the exact part of R with respect to x_2.
5. Again, calculate the residue $R = R - \partial_{x_2} I_2$.
6. If $R \neq 0$, then no potential exists. This also terminates the procedure.

The result of these steps are that $I_1 = \partial_{x_2} V$ and $I_2 = -\partial_{x_1} V$.

This algorithm performs two steps. First, it checks whether a potential can be introduced, and second, if it is possible to introduce a potential the necessary conditions to represent the potential are calculated.

10.4. Simplification of Equations

This section discusses three procedures allowing the simplification of the determining equations in connection with the integration steps discussed so far.

10.4.1 Direct Separation

We already know that the integration of monomials results into a pseudo-polynomial representation of the solution. This pseudo-polynomial has the special property that the expansion coefficients possess a dependence on a reduced set of independent variables. The explicit occurrence of some of the independent variables in the representation of the solution can be used to simplify the determining equations by a direct separation of variables. Inserting the solution into the determining equations allows us to separate some parts of the determining equations from each other. Comparing the coefficients of the resulting polynomials allows us to derive a new, simplified set of determining equations. These equations are actually simpler than those from which we originally started. Suppose some parts of the determining equations are of a polynomial type in the variable x_i. We can perform a separation by setting the coefficients of the various powers of x_i equal to zero. Let us demonstrate this procedure by the following example where $x_i = z$. Consider the equation:

$$\partial_y f(x, y) + z\{\partial_x f(x, y) + \partial_x g(x)\} + z^2\{\partial_x g(x) + y\partial_x^2 g(x)\} = 0. \tag{10.55}$$

This PDE can be considered as a polynomial in z with variable coefficients. The extraction of the different coefficients of powers of z results in a new system of determining equations given by

$$\partial_y f(x, y) = 0, \tag{10.56}$$

$$\partial_x f(x, y) + \partial_x g(x) = 0, \tag{10.57}$$

$$\partial_x g(x) + y\partial_x^2 g(x) = 0. \tag{10.58}$$

The last equation (10.58) of this system can again be considered as a polynomial in y. The application of the same rules result into an additional separation of the equation. Thus, we can maintain, in a second step, the last equation in two equations. The resulting equations are

$$\partial_y f(x, y) = 0, \tag{10.59}$$

$$\partial_x f(x, y) + \partial_x g(x) = 0, \tag{10.60}$$

$$\partial_x g(x) = 0, \tag{10.61}$$

$$\partial_x^2 g(x) = 0. \tag{10.62}$$

From this set of four equations it follows by applying the integration of monomials that the solution can be represented by

$$g(x) = k_1 \text{ and } f(x, y) = k_2, \tag{10.63}$$

where k_1 and k_2 are constants of integration. This solution satisfies not only the final four equations but also the original equation from which we started. At this point, it is obvious that we can reduce the integration process of a partial differential equation to a simple chain of steps involving the integration of monomials.

10.4.2 Indirect Separation

An essential requirement of a direct separation was the existence of terms containing one independent variable in polynomial form. Generally, we cannot assume that such case will occur. In the following we will discuss a more general situation in which the independent variable only occurs in the argument of a function. If the other involved functions do not depend on the same variable, it is possible to separate parts of the equation. The separation is possible if we cross-differentiate the expressions. This procedure allows an indirect separation of terms. Cross-differentiating splits up the equation into independent parts. The separated parts depend only on a reduced set of independent variables.

To demonstrate the method, let us examine the solutions of the first-order partial differential equation

$$f(x, y) + \partial_x g(x) = 0. \tag{10.64}$$

Differentiating this PDE with respect to the independent variable y, we find that f has to satisfy the equation

$$\partial_y f(x, y) = 0. \tag{10.65}$$

Since the term containing g does not depend on y, it simply vanishes. The differentiation of the equation with respect to the conjugate variable to x allows a separation. The original equation reduces to a single determining equation containing

only f. However, the equation for f can be integrated with respect to y. The result reads

$$f(x, y) = h(x). \tag{10.66}$$

This results shows that f does not depend on y but only on x. Inserting the result into the original equation (10.64), we get

$$h(x) + \partial_x g(x) = 0. \tag{10.67}$$

This equation is an inhomogeneous ordinary differential equation of first order in g. An integration with respect to x yields the solution

$$g(x) = -\int^x h(z)\, dz. \tag{10.68}$$

The result of the indirect separation is a relation connecting f with g by an arbitrary function $h = h(x)$. If we know the function h, we also know g and f.

Although we derived simpler equations from an indirect separation, this method will not contain as much information as the direct separation, meaning that we cannot replace the original equations by the separated ones. However, we are able to use parts of the equation or the separated equation itself in other equations. This again will simplify the original equations but does not replace them.

10.4.3 Reducing the Number of Dependent Variables

The fact is that if we can uniquely solve an equation of the determining equations with respect to an unknown function, this allows us to use this function as a substitute. The substitution of this function permits an elimination of all occurrences of this function in the rest of the system. If such a situation occurs, we can reduce the number of unknown functions in the system by one. The elimination of unknown functions by using information from the equations involved is an important method of simplifying systems of PDEs. Because the number of unknown functions increases by each integration of monomials, one can imagine that after a few steps, we have a huge number of unknown functions in the determining equations. Thus, it is necessary to counterbalance the increase of unknown functions by an elimination procedure. Since not all unknown functions are useful in the elimination of dependencies, the unknown function has to satisfy several conditions. The following list contains the main properties an unknown function has to satisfy for an elimination.

1. The function with respect to which the equation under consideration is solved must depend on all variables which occur in the remaining set of equations.

2. The equation must be uniquely solvable with respect to the unknown function.

3. The equation solved does not contain derivatives of the unknown function.

The last condition is very important, because if the unknown function occurs in a derivative, we cannot replace the unknown function itself in any expression of the whole system of determining equations.

To demonstrate the elimination of unknown functions, let us consider the example

$$f(x, y) + \partial_x g(x, y) + h(x) = 0. \tag{10.69}$$

This equation is only solvable with respect to $f(x, y)$. Neither g nor h satisfy the three conditions from above. If we accidentally solve this equation [e.g., with respect to $h(x)$], we lose the information that $f(x, y) + \partial_x g(x, y)$ depends on y. On the other hand, if we solve the equation with respect to $\partial_x g(x, y)$, we cannot replace all expressions containing $g(x, y)$ which occasionally may occur in other equations (not shown here).

All methods so far discussed for simplifying and integrating partial differential equations are implemented in our *MathLie* function PDESolve[]. To solve a given system of the determining equations completely, it is necessary to repeat the discussed methods several times in different orders as long as no further simplifications are possible. To show how the function PDESolve[] can be used, let us examine the determining equations of the Korteweg-de Vries equation. These equations follow by

```
detEquationsKdV = DeterminingEquations[
  {∂t u[x, t] + u[x, t] ∂x u[x, t] + ∂x,x,x u[x, t]},
  {u}, {x, t}, {∂t u[x, t]}];
detEquationsKdV // LTF
```

$$(\xi_1)_u == 0$$

$$(\xi_2)_u == 0$$

$$(\phi_1)_{u,u} == 0$$

$$(\xi_2)_x == 0$$

$$\phi_1 - (\xi_1)_t - u\,(\xi_1)_x + u\,(\xi_2)_t - (\xi_1)_{x,x,x} + 3\,(\phi_1)_{x,x,u} == 0$$
$$(\phi_1)_t + u\,(\phi_1)_x + (\phi_1)_{x,x,x} == 0$$
$$-(\xi_1)_{x,x} + (\phi_1)_{x,u} == 0$$
$$-3\,(\xi_1)_x + (\xi_2)_t == 0$$

These eight equations are linear but coupled. Before we use the function PDESolve[], let us look at the shorthand description of the function:

```
? PDESolve

PDESolve[list of equations,dependent vars., indep.
  vars., Options]
```

The related option of the function PDESolve[] are

```
Options[PDESolve]

{Standard → False, WarningSS → False, TraceStep → True}
```

The application of this function to the determining equations of the Korteweg-de Vries equation gives us the result

```
infinitesimalsKdV = PDESolve[detEquationsKdV, {u}, {x, t}];
infinitesimalsKdV // LTF
```

$$\xi_1 == k1 + k2\,t + \frac{k4\,x}{3}$$
$$\xi_2 == k3 + k4\,t$$
$$\phi_1 == k2 - \frac{2\,k4\,u}{3}$$

remembering that the KdV equation allows a four-dimensional symmetry group.

10.5. Example

The algorithm to solve partial differential equations is designed to treat coupled systems of determining equations. Such determining equations occur frequently in symmetry analysis. The following example demonstrates the capabilities of the function PDESolve[] in connection with a non-standard equation. The example originates from quantum gravity theory and demonstrates the flexibility of the algorithm in connection with special functions.

10.5.1 Liouville-Type Equation of Quantum Gravity Theory

The first example is related to an equation occurring in quantum gravity theory describing the evolution of the Killing vector. Boyer and Finley [1989] showed that the Killing vector of a rotational symmetric space can be described by an equation of the form

$$\partial_{x,x}\, u(x, y, z) + \partial_{y,y}\, u(x, y, z) + \partial_{z,z}\, e^{u(x,y,z)} = 0. \tag{10.70}$$

This equation is of some importance in quantum gravity theory, dealing with a special form of general relativity theory including quantum effects. The solution of the Killing equation determines a Riemann metric which admits only vectors which are invariant under rotations. A first solution of equation (10.70) was given by Drew et al. [1989].

Applying Lie's symmetry method to this equation will give us a system of 16 determining equations:

```
detEqsLiouville = DeterminingEquations[
  {∂x,x u[x, y, z] + ∂y,y u[x, y, z] + ∂z,z Exp[u[x, y, z]]},
  {u}, {x, y, z}, {∂x,x u[x, y, z]}];
detEqsLiouville // LTF
```

$$(\xi_1)_u == 0$$

$$(\xi_2)_u == 0$$

$$(\xi_3)_u == 0$$

$$(\phi_1)_{u,u} == 0$$

$$(\xi_1)_z == 0$$

$$(\xi_2)_z == 0$$

$$(\xi_3)_y == 0$$

$$(\xi_3)_x == 0$$

$$-(\xi_2)_{x,x} - (\xi_2)_{y,y} + 2\,(\phi_1)_{y,u} == 0$$

$$-(\xi_1)_{x,x} - (\xi_1)_{y,y} + 2\,(\phi_1)_{x,u} == 0$$

$$(\phi_1)_{x,x} + (\phi_1)_{y,y} + E^u\,(\phi_1)_{z,z} == 0$$

$$(\xi_1)_y + (\xi_2)_x == 0$$

$$\phi_1 + 2\,(\xi_1)_x - 2\,(\xi_3)_z + (\phi_1)_u == 0$$

$$\phi_1 + 2\,(\xi_1)_x - 2\,(\xi_3)_z == 0$$

$$-(\xi_1)_x + (\xi_2)_y == 0$$

$$2\,(\phi_1)_z - (\xi_3)_{z,z} + 2\,(\phi_1)_{z,u} == 0$$

Using the notation of *Mathematica,* we denote the derivative with respect to the k*th* argument by a superscript at the k*th* position. The solution of these equations follow within a few seconds by applying the function PDESolve[] to these equations. The result of this calculation is

```
infiniLiouville =
 PDESolve[detEqsLiouville, {u}, {x, y, z}]; infiniLiouville //
 TableForm[LieTraditionalForm[#], TableHeadings →
    {{"infinitesimals", "det-equations"}, Automatic},
   TableDirections -> {Row, Column}]&
```

	infinitesimals	det-equations
1	$\xi_1 \to -(\mathcal{F}_2)_y$	$(\mathcal{F}_2)_{x,x} + (\mathcal{F}_2)_{y,y}$
2	$\xi_2 \to -(\mathcal{F}_2)_x$	$\mathcal{F}_3 + (\mathcal{F}_2)_{x,x} + (\mathcal{F}_2)_{y,y}$
3	$\xi_3 \to k1 + k2\ z$	$\mathcal{F}_4 + (\mathcal{F}_2)_{x,x} + (\mathcal{F}_2)_{y,y}$
4	$\phi_1 \to 2\ (k2 + (\mathcal{F}_2)_{x,y})$	$-\mathcal{F}_1 + (\mathcal{F}_2)_{x,x,y} + (\mathcal{F}_2)_{y,y,y}$

The result of the calculation is a combination of a discrete and a continuous group. The unknown function $\mathcal{F}_2 = free[2]$ of the continuous part has to satisfy the two-dimensional Laplace equation

$$(\mathcal{F}_2)_{x,x} + (\mathcal{F}_2)_{y,y} = 0. \tag{10.71}$$

The point here is that this result is different from the result given by Drew et al. The difference is not a marginal one. However, since a different result was derived by our program, we had to check the results of Drew et al. with another program. We substituted the result of Drew et al. into our determining equations and discovered that their result do not solve our determining equations. On the other hand, our result is a solution of the determining equations derived by the function Infinitesimals[]. We also examined the validity of our determining equations by using a different computer algebra program. Using the *Maple* program by Reid [1991], we could check that our derived determining equations are correct and that our solution solves the equations obtained from Reid's program.

The solution given by Drew et al. did not satisfy the determining equations derived by Reid's program in general. Only for the special case with

$$free[2][x, y] = \text{const}. \tag{10.72}$$

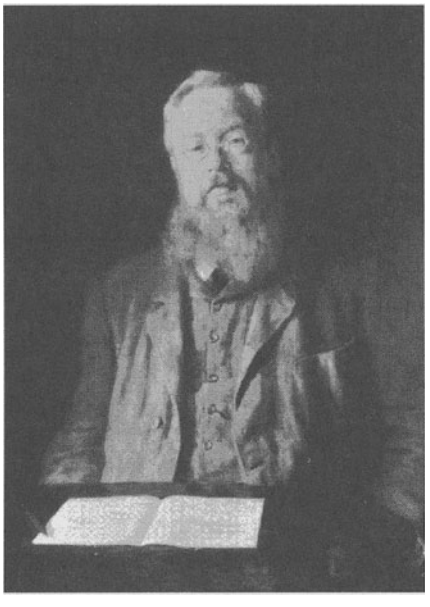
could the equations be satisfied. This choice of the unknown function was the main case discussed by Drew et al. All these various checks proved that our result is correct. Furthermore, we could show that there is no transformation which maps both solutions into each other. The result given by Drew et al. has a completely different structure of the infinitesimals ξ_1, ξ_2, and ϕ_1, but the expression for ξ_3 is identical in both results.

11

Appendix

A. Marius Sophus Lie: A Mathematician's Life

Born: Nordfjordeide, Norway, 17 December 1842
Died: Christiania (Oslo), Norway, 18 February 1899.



Sophus Lie: Derived from a painting by Erik Werenskiold (Engel and Heegaard Vol. 2 [1912]).

Marius Sophus Lie, a Norwegian mathematician who made significant contributions to the theories of algebraic invariants and differential equations, was the youngest of six children of a Lutheran pastor, Johann Herman Lie. His mother came from a well-known Trondheim family. Lie first attended school in Moss (Kristianiaford); then, from 1857 to 1859, he attended Nissen's Private Latin School in Christiania. Lie mastered the classes without any difficulties. He studied at Christiania University from 1859 to 1865, mainly mathematics and sciences. During his studies at Christiania, he had no preference for mathematics. Although mathematics was taught by Bjerknes and Sylow, Lie was not very impressed. After his examination in 1865, he gave private lessons, became somewhat interested in astronomy, and tried to learn mechanics; but he could not decide what to do. Lie himself said that the road to mathematics for him was long and difficult. The situation changed when, in 1868, he hit upon Poncelet's and Plücker's writings. Later, he called himself a student of Plücker, although he had never met him. Plücker's momentous idea to create new geometries by choosing figures other than points—in fact, straight lines—as elements of space pervaded all of Lie's work.

Lie's first publication brought him a scholarship for studying abroad. In 1869, Lie went to Berlin, where he met Felix Klein, with whom he later cooperated in publishing several papers. He spent the winter of 1869-1870 in Berlin where he met Kummer and Weierstraß. In the summer of 1870, Lie and, later, Klein traveled to Paris via Göttingen to meet Darboux and Jordan. Jordan acquainted Lie and Klein with the notion of a group introduced into algebra by Galois in 1832. In 1870, Lie discovered contact transformations. Using these transformations, a one-to-one correspondence could be established between lines and spheres in a way that tangent spheres correspond to intersecting lines. He also became familiar with Monge's theory of differential equations. At the outbreak of the Franco-Prussian war in July of 1870, Klein left Paris; Lie, a Norwegian, stayed. In August, he decided to hike to Italy, but on his way he was arrested as a German spy near Fontainebleau. His mathematical notes were suspected to be military secrets in code—a letter from Klein seemed suspicious. After being locked in prison for a month, he was freed through Darboux's intervention. Just before the Germans blockaded Paris, he escaped to Italy. From there, he returned to Germany, where he again met Klein in Düsseldorf.

In 1871, he became an assistant tutor at the University of Cristiania (Oslo). In the same year, he submitted for his doctor's degree a memoir in which he advanced the theory of tangential transformations. During the period 1871–1872, he developed the integration theory of partial differential equations, now found in many textbooks, although rarely under his name. Appointed extraordinary professor on 1 July 1872, he began his researches on continuous transformation groups in 1873. In 1874, Lie married Anna Birch from Tvedestrand. They had two daughters and a son. The marriage was very happy, and Lie was very fond of his family. In 1873, Lie turned from the invariants of contact transformations to the principles of the theory of

transformation groups. Together with Sylow, he assumed the editorship of Niels Abel's work.

Lie was quite isolated in Christiania at that time. He had no students who were interested in his research. He was disappointed that his works did not receive more attention abroad. Except for Klein, Mayer, and later, Picard, nobody paid attention to his work. Lie's results on the integration theory of partial differential equations were found by Adolph Mayer at that time, with whom he had conducted a lively correspondence. In a letter to Mayer he writes, "If I only knew how to get the mathematicians interested in transformation groups and their applications to differential equations. I am certain, absolutely certain in my case, that in the future these theories will be recognized as fundamental. I want to form thus an impression now, since for one thing, I could then achieve ten times as much." His main interest turned to transformation groups, his most celebrated creation; although in 1876, he returned to differential geometry. In the same year, he joined G.O. Sars and Worm Müller in founding the *Archiv för mathematik og naturvidenskab*.

In 1884, Klein and Mayer induced Engel, who had just received his Ph.D., to visit Lie in order to learn about transformation groups and to help him write a comprehensive book on the subject. Engel stayed 9 months with Lie. Thanks to Engel's activity, the work was accomplished, its three parts being published between 1888 and 1893. Engel and Lie developed a warm and lifelong friendship. Engel helped Lie by giving his rather intuitive geometrical ideas a more precise mathematical form. Lie often felt it a burden to prepare his ideas for publication. After 9 years of collaboration with Engel, Lie published *Theorie der Transformationsgruppen*, 3 vols. (1893). This work contains the results of his investigations of the general theory of finite continuous groups of transformations. It was followed by *Geometrie der Berührungstransformationen* (1896).

In 1886, he succeeded Klein on the chair of mathematics at the University of Leipzig, with Engel as his assistant. At Leipzig, he found interested students, among them Scheffers, Zorawski, and Kowalewski. With Scheffers, Lie published textbooks on transformation groups and on differential equations, and a fragmentary geometry of contact transformations. Afterward, his student Kowalewski, wrote many books about Lie's work. At this time, it was quite unusual for young French mathematicians to go to Germany for studying. But the Ecole Normale Supérieure in Paris sent some of their best students to Lie; and he was very proud of this. Lie did not plan to stay in Leipzig forever; he had in mind a period of 6 to 8 years. So he did not resign from his professorship in Christiania, but was granted an extraordinary leave of absence. Life in Leipzig was not that easy for Lie. His teaching duties were much heavier than at home, the language caused him some problems, and he became tired of supervising weak and dependent graduate students. As time passed, he also ran into trouble with

some of his colleagues. In the last years of his life, Lie turned to foundations of geometry, which at that time meant the Helmholtz space problem.

In 1898, he returned to Cristiania to accept a special chair of mathematics created for him, but his health was already broken. Lie, who was described as on open-hearted man of gigantic stature and excellent physical health, was struck by what was then called neurasthenia. This was the result of his rushing work and the overload of mental action. Treatment in a mental hospital led to his recovery, and in 1890, he could resume his work. His character, however, had changed greatly. He became increasingly sensitive, irascible, suspicious, and misanthropic, despite the many tokens of recognition that were heaped upon him. He died of pernicious anemia in February 1899. His papers were edited, with excellent annotations, by Engel and Heegaard.

An analysis of Lie's work is given in the *Bibliotheca Mathematica* (1900). His collected works are contained in *Gesammelte Abhandlungen*, 7 vols (1922–37). Two other standard works are his *Differentialgleichungen* (1891) and *Vorlesungen über continuierliche Gruppen* (1893).

In 1890, Lie himself wrote on his work in a letter to his friend Motzfeldt, "...my life' s work will stand through all times and, in the years to come, be more and more appreciated—no doubt about it." It seems he was absolutely right!

B. List of Key Symbols Used in *Mathematica*

This section summarizes some key symbols used in *Mathematica*. The compiled list contains the main shortcuts for symbols used in the text.

Abbreviation	Function description
=	assignment
==	used in equations
===	equality testing
:=	function definition lhs := rhs assigns rhs to be the delayed value of lhs.
/;	conditional (provided that)
→	rule assignment
:> or :→	delayed rule assignment
x_	pattern matching to free variable *x*
x__	sequence of symbols named *x*
x___	sequence of zero or more expressions named *x*
/.	under the rule ReplaceAll
# or #1	slot in a pure function
(#)&	pure function definition
**	non-commutative multiply

C. Installing *MathLie*

The following instructions serve to install *MathLie* a program based on the former program Lie by G. Baumann [1992].

C.1 Windows 95

C.1.1 Manual Installation

1. Create the directory "AddOns\Applications*MathLie*" at that location where your *Mathematica* files are located. You'll get the location by

```
$TopDirectory
```

```
C:\PROGRAMME\WOLFRAM RESEARCH\MATHEMATICA\3.0\
```

2. Copy all the files and directories on CD in the directory *MathLie* into the directory "AddOns\Applications\ *MathLie*."

3. Go to the Help menu and select "Rebuild Help Index."

4. *MathLie* uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath = $TopDirectory<>"/AddOns/Applications/MathLie/"
```

```
C:\PROGRAMME\WOLFRAM  RESEARCH\MATHEMATICA\3.0/AddOns/
  Applications/MathLie/
```

5. Add to the variable $Path in the init.m file the location of *MathLie*.
AppendTo[$Path,$MathLiePath"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

C.1.2 Automatic Installation

The following lines install the package and the documentation of *MathLie*. The files are located in the *AddOns* application directories.

```
Clear[Install]
 Install[] := Block[{location1, source},
  location1 = $TopDirectory<>"/AddOns/Applications/MathLie";
  source = Input[
```

```
    "Where is the CD located? Use a string as input value;
eg. d:/MathLie ."];
  CreateDirectory[location1];
  CopyDirectory[source, location1];
  Print["Installation successfully
terminated"]];
Install[]
```

Installation successfully terminated

Go to the Help menu and select "Rebuild Help Index."

MathLie uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath =
 $TopDirectory<>"/AddOns/Applications/MathLie/";
```

Add to the variable $Path in the file init.m the location of *MathLie*.
AppendTo[$Path,$TopDirectory<>"/AddOns/Applications/*MathLie*"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

C.2 Mac

C.2.1 Manual Installation

1. Create the directory "AddOns\Applications*MathLie*" at that location where your *Mathematica* files are located. You'll get the location by

```
$TopDirectory
```

```
C:\PROGRAMME\WOLFRAM RESEARCH\MATHEMATICA\3.0\
```

2. Copy all the files and directories on disk/CD in the directory *MathLie* into the directory "AddOns\Applications\ *MathLie.*"

3. Go to the Help menu and select "Rebuild Help Index."

4. *MathLie* uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath = $TopDirectory<>"/AddOns/Applications/MathLie/"
```

```
C:\PROGRAMME\WOLFRAM  RESEARCH\MATHEMATICA\3.0/AddOns/
  Applications/MathLie/
```

5. Add to the variable $Path in the init.m file the location of *MathLie*.
AppendTo[$Path,$MathLiePath"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

C.2.2 Automatic Installation

The following lines install the package and the documentation of *MathLie*. The files are located in the *AddOns* application directories.

```
Clear[Install]
 Install[] := Block[{location1, source},
  location1 = $TopDirectory<>"/AddOns/Applications/MathLie";
  source = Input[
    "Where is the CD located? Use a string as input value;
eg. d:/MathLie ."];
  CreateDirectory[location1];
  CopyDirectory[source, lication1];
  Print["Installation successfully
terminated"]];
Install[]
```

```
Installation successfully terminated
```

Go to the Help menu and select "Rebuild Help Index."

MathLie uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath =
 $TopDirectory<>"/AddOns/Applications/MathLie/";
```

Add to the variable $Path in the file init.m the location of *MathLie*.
AppendTo[$Path,$TopDirectory<>"/AddOns/Applications/*MathLie*"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

C.3 UNIX

C.3.1 Manual Installation

1. Create the directory "*MathLie*" in your $HOME/.Mathematica/3.0/AddOns/ Applications directory. You'll get the location of your home directory by

```
$HomeDirectory

/users / depart1 / gbaumann
```

2. Copy all the files and directories on CD in the directory *MathLie* into the directory "*MathLie.*"

3. Go to the Help menu and select "Rebuild Help Index."

4. *MathLie* uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath = $HomeDirectory<>"/MatLie/"

/users/depart1/gbaumann/MathLie/
```

5. Add to the variable $Path in the init.m file the location of *MathLie*. AppendTo[$Path,$MathLiePath"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

C.3.2 Automatic Installation

The following lines install the package and the documentation of *MathLie*. The files are located in the *AddOns* application directories.

```
Clear[Install]
 Install[] := Block[{location1, source},
  location1 = $HomeDirectory<>
    "/.Mathematica/3.0/AddOns/Applications/MathLie";
  source = Input[
    "Where is the CD located? Use a string as input value;
eg. d:/MathLie ."];
  CreateDirectory[location1];
  CopyDirectory[source, location1];
  Print["Installation successfully
terminated"]];
Install[]
```

```
Installation successfully terminated
```

Go to the Help menu and select "Rebuild Help Index."

MathLie uses the global variable $MathLiePath to locate different files. Set this variable to the path where you have located the files.

```
$MathLiePath = $HomeDirectory<>
  "/.Mathematica/3.0/AddOns/Applications/MathLie";
```

Add to the variable $Path in the file init.m the location of *MathLie*.
AppendTo[$Path,$HomeDirectory<>"/.Mathematica/3.0/AddOns/Applications/*MathLie*"]

```
AppendTo[$Path, $MathLiePath]
```

You are now ready to use *MathLie* and view the Help information.

References

A

B. Abraham-Schrauner and A. Guo, Hidden and nonlocal symmetries of nonlinear differential equations, pp. 1-5, In: *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, Eds: N.H. Ibragimov, M. Torrisi, and A. Valenti, Kluwer, Dordrecht, 1993.

A.A. Alexeyev, Approximating pseudopotentials, scattering problems, and Bäcklund transformations for the Korteweg-de Vries and modified Korteweg-de Vries equations with perturbations, *J. Phys. A: Math. Gen.* **27**, 865–881 (1994).

W.F. Ames, *Nonlinear Ordinary Differential Equations in Transport Processes*, Academic Press, New York, 1968.

B

V.A. Baikov, R.K. Gazizov, and N.H. Ibragimov, Approximate symmetries, *Math. USSR Sbornik*, **46**, 427–441 (1989).

V.A. Baikov, R.K. Gazizov, and N.H. Ibragimov, Perturbation methods in group analysis, *J. Sov. Math.* **55**, 1450 (1991).

G. Baumann, Integrability and chaos for two copropagating pulses in optical fibers, *Phys. Lett.* **A156**, 298–302 (1991).

G. Baumann, Lie Symmetries of Differential Equations, A *Mathematica* program to determine Lie symmetries, Wolfram Research, Inc., Champaign, IL, Math-Source 0202–622 (1992).

G. Baumann, *Mathematica in Theoretical Physics*, Springer-Verlag/TELOS, New York, 1996.

G. Baumann, Symmetry analysis of differential equations with *Mathematica*, *Math. Comput. Modelling* **25**, 25–37 (1997).

G. Baumann and T.F. Nonnenmacher, Regular and chaotic motions in ion traps: A nonlinear analysis of trap equations, *Phys. Rev.* **A 46**, 2682–2692 (1992).

G. Baumann and T.F. Nonnenmacher, Lie transformations, similarity reduction, and solutions for the nonlinear Madelung fluid equations with external potential, *J. Math. Phys.* **28**, 1250–1260 (1987).

G. Birkhoff, *Hydrodynamics*, Princeton University Press, Princeton 1950.

Ph. Blanchard and E. Brüning, *Variational Methods in Mathematical Physics*, Springer-Verlag, Berlin, 1992.

H. Blasius, Grenzschichten in Flüssigkeiten mit kleiner Reibung, *Z. Angew. Math. Phys.* **56**, 1–37 (1908).

G.W. Bluman, Potential symmetries and equivalent conservation laws, p. 71; In: *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, Eds: N.H. Ibragimov, Kluwer Academic Publishers, Dordrecht, 1993.

G.W. Bluman, The Use of Factors to Discover Potential Systems or Linearizations, preprint (1993).

G.W. Bluman and J.D. Cole, The general simlarity solution of the heat equation, *J. Math. Mech.* **18**, 1025–1042 (1969).

G.W. Bluman and J.D. Cole, *Similarity Methods for Differential Equations*, Springer-Verlag, New York, 1974.

G.W. Bluman and S. Kumei, *Symmetries and Differential Equations*, Springer-Verlag, New York, 1989.

G.W. Bluman and S. Kumei, Symmetry-based algorithms to relate partial differential equations: I. Local symmetries, *Eur. J. Appl. Math.* **1**, 189–216 (1990).

T. Bountis, H. Segur, and F. Vivaldi, Integrable Hamiltonian systems and the Painlevé property, *Phys. Rev.* **A25**, 1257–1264 (1982).

J. Boussinesq, Théorie des ondes te des remous qui se propagent le long d'un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement parielles de la surface au fond, *J. Math. Pures Appl.* **7**, 55 (1872).

C.P. Boyer and J.D. Finley, Killing vectors in self-dual, Euclidian Einstein spaces, *J. Math. Phys.* **23**, 1126–1130 (1989).

B. Buchberger, Groebner bases: An algorithmic method in polynomial ideal theory, In: *Multidimensional Systems Theory*, Ed: N.K. Bose, Reidel Publ., Dordrecht, pp. 184–232, 1985.

J.M. Burgers, A mathematical model illustrating the theory of turbulence, *Adv. Appl. Mech.* **1**, 171-199, 1948.

C

F. Calogero and A. Degasperis, Solution by the spectral transform method of a nonlinear evolution equation including as a special case the cylindrical KdV equation, *Lett. Nuovo Cim.* **23**, 150–154 (1978).

F. Calogero and A. Degasperis, Reduction techniques for matrix nonlinear evolution equations solvable by the spectral transform, *J. Math. Phys.* **22**, 23–31 (1981).

G. Carrà-Ferro, Differential-algebraic and differential-geometric approach to the study of involutive symbols, pp. 93–99, In: *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, Eds.: N.H. Ibragimov, M. Torrisi, and A. Valenti, Kluwer, Dordrecht, 1993.

B. Champagne, W. Hereman, and P. Winternitz, The computer calculation of Lie point symmetries of large systems of differential equations, *Comp. Phys. Comm.* **66**, 319-340 (1991).

W.L. Chan and K.S. Li, Non-propagating solitons of the non-isospectral and variable coefficient modified KdV equation, *J. Phys. A: Math. Gen.* **27**, 883–902 (1994).

G. Cicogna and D. Vitali, Classification of the extended symmetries of Fokker-Planck equations, *J. Phys. A: Math. Gen.* **23**, L85–L88 (1990).

P.A. Clarkson, New similarity solutions for the modified Boussinesq equation, *J. Phys. A: Math. Gen.* **22**, 2355–2367 (1989).

P.A. Clarkson and M.D. Kruskal, New similarity solutions of the Boussinesq equation, *J. Math. Phys.* **30**, 2201–2213 (1989).

P.A. Clarkson and E.L. Mansfield, Algorithms for the Non-classical Method of Symmetry Reduction, *SIAM J. Appl. Math.* **54**, 1693–1719. (1994).

R. Courant and K.O. Friederichs, *Supersonic Flow and Shock Waves*, Interscience Publisher, Inc., New York, 1948.

D.G. Crighton, Model equations of nonlinear acoustics. *Rev Fluid Mech.* **11**, 11–23 (1979).

D

S. Das Sarma and P. Tamborenea, A new universality class for kinetic growth: One-dimensional molecular-beam epitaxy, *Phys. Rev. Lett.* **66**, 325–328 (1991).

P.G. Drazin, *Solitons*, Cambridge University Press, Cambridge, 1983.

M.S. Drew, S.C. Kloster, and J.D. Gegenberg, Lie group analysis and similarity solution for the equation $\partial_{x,x}\, u(x, y, z) + \partial_{y,y}\, u(x, y, z) + \partial_{z,z}\, e^{u(x,y,z)} = 0$, *Nonlinear Anal. Theory Methods and Applic.*, **13**, 489–505 (1989).

E

J. Eggers, Universal pinching of 3D axisymmetric free-surface flow, *Phys. Rev. Lett.* **71**, 3458–3460 (1993).

J. Eggers, Tropfenbildung, *Phys. Bl.* **53**, 431–434 (1997).

F. Engel, Sophus Lie, *Bibliotheca Mathematica* **3**, 166–204 (1900).

F. Engel and P. Heegaard, *Sophus Lie Gesammelte Abhandlungen*, B.G. Teubner, Leipzig; H. Aschehoug & Co., Kristiania, 1912–1934.

F

V.M. Falkner and S.W. Skan, Solution of the boundary-layer equations, *Phil. Mag.* **12**, 865–896 (1931).

A.S. Fokas, A symmetry approach to exactly solvable evolution equations, *J. Math. Phys.* **21**, 1318–1325 (1980).

A.S. Fokas and B. Fuchssteiner, Bäcklund transformations for hereditary symmetries, *Nonlinear Theory Methods Applic.*, **5**, 423–432 (1981).

A.S. Fokas, Symmetries and integrability, *Studies Appl. Math.*, **77**, 253–299 (1987).

P.C.W. Fung and C. Au, Bridge between the solutions and vacuum states of the Korteweg-de Vries equation and that of the nonlinear equation $y_t + y_{xxx} - 6y^2 y_x + 6\lambda y_x = 0$, *Phys. Rev. B* **26**, 4035–4038 (1982).

G

F.J. Garcia and A. Castellanos, One-dimensional model for slender axisymmetric viscous liquid jets, *Phys. Fluids* **6**, 2676–2689 (1994).

A. Gray, Modern *Differential Geometry of Curves and Surfaces*, CRC Press, Boca Raton, FL, 1993.

H

M. Hénon and C. Heiles, The applicability of the third integral of motion: Some numerical experiments, *Astron.* J. **69**, 73–79 (1964).

W. Hereman, Review of symbolic software for the computation of Lie symmetries of differential equations, *Euromath. Bull.*, **2**, 45–82 (1994).

W. Hereman, Symbolic software for Lie symmetry analysis pp. 367–413, Ed.: N.H. Ibragimov, In: *CRC Handbookof Lie Group Analysis of Differential Equations, Vol. 3: New Trends in Theoretical Developments and Computational Methods*, CRC Press, Boca Raton, FL, 1996.

R. Hirota and J. Satsuma, Soliton solutions of a coupled Korteweg de Vries equation, *Phys. Lett.* **85**, 407–408 (1981).

P.E. Hydon, Conformal symmetries of first-order ordinary differential equations, *J. Phys. A: Math. Gen.* **27**, 385–392 (1994).

I

N.H. Ibragimov, *Transformation Groups Applied to Mathematical Physics*, Reidel Publ., Dortrecht, 1985.

N.H. Ibragimov, Sophus Lie and harmony in mathematical physics, on the 150th anniversary of his birth, *Math. Intel.* **16**, 20–28 (1994).

N.H. Ibragimov, *CRC Handbook of Lie Group Analysis of Differential Equations*, Vols. 1–3, CRC Press, Boca Raton, FL 1994, 1995, 1996.

E.L. Ince, *Ordinary Differential Equations*, Dover Publications, New York, 1956.

J

M. Janet, Sur les systèmes d'équation aux dériveés partielles, *J. Math. Pure Appl.* **3**, 65–151 (1920).

K

L.P. Kadanoff, Exact solutions of the Saffman-Taylor problem with surface tension, *Phys. Rev. Lett.* **65**, 2986–2988 (1990).

E. Kamke, *Differentialgleichungen*, G.B. Teubner, Stuttgart, 1977.

V.I. Karpman and V.Yu. Belashov, Dynamics of two-dimensional solutions in weakly dispersive media, *Phys. Lett.* **154**, 131–139 (1991).

F. Klein, Über die Differentialgesetze für die Erhaltung von Impuls und Energie in der Einsteinschen Gravitationstheorie, *Nachr. Ges. Wiss. Göttingen Math. Phys.* **2**, 171–189 (1918).

K. Ko, and H.H. Kuehl, Korteweg-de Vries soliton in a slowly varying medium, *Phys. Rev. Lett.* **40**, 233–236 (1978).

D.J. Korteweg, and G. de Vries, On the change of form of long waves advancing in a rectangular channel, and on a new type of long stationary waves, *Phil. Mag.* **39**, 442–443 (1895).

G. Kowalewski, *Einführung in die Theorie der Kontinuierlichen Gruppen*, Akademische Verlagsgesellschaft M.B.H, Leipzig, 1931.

L

H. Lamb, *Hydrodynamics*, Dover Publications, New York, 1945.

L.D. Landau and E.M. Lifshitz, *Mechanics*, Butterworth-Heinemann, New York, 1981.

L.D. Landau and E.M. Lifshitz, *Fluid Mechanics*, Pergamon, Oxford, 1987.

D. Levi and P. Winternitz, Non-classical symmetry reduction: Example of the Boussinesq equation, *J. Phys. A: Math. Gen.* **22**, 2915–2924 (1989).

S. Lie, *Gesammelte Werke*, Vols. 1–7, Eds.: F. Engel and P. Hergard, Teubner, Leipzig, 1899.

S. Lie, Zur Theorie des Integrabilitätsfaktors, *Christ. Forh., Aar* 1874, 242–254, 1875.

S. Lie and F. Engel, *Transformationsgruppen*, Vols. I–II, Leipzig, (1888, 1890, 1893); reprinted by Chelsea, New York, 1970.

M.J. Lighthill, Viscosity effects in sound waves of finite amplitude, pp. 250–351. In: *Surveys in Mechanics*, Eds.: G.K. Batchelor and R.M. Davis, Cambridge University Press, Cambridge, 1956.

M

W. Ma, An exact solution in two-dimensional Korteweg-de Vries-Burgers equation, *J. Phys. A: Math. Gen.* **26**, L17–L20 (1993).

J.E. Mack, Semi-popular motion picture record of the Trinity explosion. MDDDC221. U.S. Atomic Energy Commission, Washington, DC, 1947.

E.L. Mansfield, *Differential Gröbner Bases*, Ph.D. Thesis, University of Sydney, Australia, 1992.

A.A. Mohammad, and M. Can, Exact solutions of the complex modified Korteweg-de Vries equation, *J. Phys. A: Math. Gen.* **28**, 3223–3233 (1995).

N

T. Nishitani and M. Tajiri, On similarity solutions of the Boussinesq equation, *Phys. Lett.* **89A**, 379–380 (1982).

E. Noether, Invariante Variationsprobleme, *Nachr. König. Gesell. Wissen. Göttingen, Math.-Phys. Kl.* **2** 235–257, (1918); see *Transport Theory Stat. Phys.* **1**, 186–207 (1971) for an English translation.

M.C. Nucci and P.A. Clarkson, The nonclassical method is more general than the direct method for symmetry reductions. An example of the Fitzhugh-Nagumo equation, *Phys. Lett. A* **164**, 49–56 (1992).

O

P.J. Olver, *Applications of Lie Groups to Differential Equations*, Springer-Verlag, Berlin, 1986.

P.J. Olver and P. Rosenau, The construction of special solutions to partial differential equations, *Phys. Lett. A* **114**, 107–112 (1986).

P.J. Olver and P. Rosenau, Group-invariant solutions of differential equations, *SIAM J. Appl. Math.* **47**, 263–278 (1987).

L.V. Ovsiannikov, *Group Analysis of Differential Equations*, Academic Press, New York, 1982.

P

E.J. Parkes, Exact solutions to the two-dimensional Korteweg-de Vries-Burgers equation, *J. Phys. A: Math. Gen.* **27**, L497–L501 (1994).

W. Paul and H. Steinwedel, Ein neues Massenspektrometer ohne Magnetfeld, *Z. Naturforschg.* **8a**, 448–450 (1953).

M.D.K. Porsezian and M. Lakshmanan, On the integrable models of the higher-order water wave equation, *Phys. Lett. A* **174**, 237–240 (1993).

E. Pucci, Similarity reductions of partial differential equations, *J. Phys. A: Math. Gen.* **25**, 2631–2640 (1992).

R

G. Reid, Findng abstract Lie symmetry algebras of differential equations without integrating determining equations, *Eur. J. Appl. Math.* **2**, 318–340 (1991).

G.J. Reid, D.T. Weih, and A.D. Wittkopf, A point symmetry group of a differential equation which cannot be found using infinitesimal methods, pp. 311–316, In: *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, Eds.: N.H. Ibragimov, M. Torrisi, and A. Valenti, Kluwer, Dordrecht, 1993.

Ch. Riquier, *Les systèmes d'équations aux dérivées partielles*, Gauthier-Villars, Paris, 1910.

H. Risken, *The Fokker-Planck Equation*, Springer-Verlag, New York, 1984.

P. Rosenau and J.L. Schwarzmeier, On similarity solutions of Boussinesq-type equations, *Phys. Lett.* **115A**, 75–77 (1986).

J.S. Russell, Report on waves, Report of the 14th Meeting of the British Association for the Advancement of Science, pp. 319–320, John Murray, London, 1844.

S

P.G. Saffman and G. Taylor, The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid, *Proc. Roy. Soc.* **245**, 312–329 (1958) .

G. Scheffers and S. Lie, *Vorlesungen über Differentialgleichungen mit bekannten infinitesimalen Transformationen*, B.G. Teubner, Leipzig, 1891.

H. Schlichting, Laminare Strahlausbreitung, *Z. Ang. Math. Mech.* **8**, 260–263 (1933).

F. Schwarz, Automatically determining symmetries of partial differential equations, *Computing* **34**, 91–106 (1985).

F. Schwarz, Reduction and completition algorithms for partial differential equations, pp. 27.6–29.6, In: *International Symposium on Symbolic and Algebraic Computation*, Berkeley, Ed.: P.S. Wang, ACM Press, New York, 1992.

H. Stephani, *Differential Equations: Their Solution Using Symmetries*, Cambridge University Press, Cambridge, 1989.

G.G. Stokes, On the dynamical theory of diffraction, *Trans. Camb. Phil. Soc.* **9**, 1–62 (1851).

T

M. Tabor, *Chaos and Integrability in Nonlinear Dynamics*, John Wiley & Sons, New York, 1989.

C. Tang, S. Feng, and L. Golubovic, Dynamics and noise spectra of a driven single flux line in superconductors, *Phys. Rev. Lett.* **72**, 1264–1267 (1994).

G. Taylor, The formation of a blast wave by a very intense explosion, I. Theoretical discussion, *Proc. Roy. Soc.* **201**, 159–174 (1950); II. The atomic explosion of 1945, *Proc. Roy. Soc.* **201**, 175–186 (1950).

J.M. Thomas, Riquier's existence theorems, *Ann. Math.* **30**, 285–321 (1929).

J.M. Thomas, Riquier's existence theorems, *Ann. Math.* **35**, 306–311 (1934).

M. Tinkham, *Introduction to Superconductivity*, McGraw-Hill, New York, 1975.

V

N.G. Van Kampen, *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam, 1981.

A.M. Vinogradov and I.S. Krasil'shchik, On the theory of nonlocal symmetries of nonlinear partial differential equations, *Sov. Math. Dokl.* **29**, 337–341 (1984).

W

H. Weyl, *Gruppentheorie und Quantenmechanik*, S. Hirzel, Leipzig, 1928.

D.J. Wineland, W.M. Itano, and R.S. van Dyck Jr., High-resolution spectroscopy of stored ions, *Adv. Atom. Mol. Phys.* **19**, 135–186 (1983).

T. Wolf, *The Symbolic Integration of Exact PDEs*, preprint, 1991.

D.E. Wolf and J. Villain, Growth with surface diffusion, *Europhys. Lett.* **13**, 389–394 (1990).

S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley Publishing Company, Reading, MA, 1991.

Y

Z.J. Yang, Traveling wave solutions to nonlinear evolution and wave equations, *J. Phys. A: Math. Gen.* **27**, 2837–2855 (1994).

Z

N.J. Zabusky and M.D. Kruskal, Interaction of solitons in a collisionless plasma and the recurrence of initial states, *Phys. Rev. Lett.* **15**, 240–243 (1965).

N.S. Zakharov and V.P. Korobeinikov, Group analysis of the generalized Korteweg-de Vries-Burgers equation, *J. Appl. Math.* **44**, 668–671 (1980).

V.E. Zakharov and E.A. Kuznetsov, Three-dimensional solitons, *Sov. Phys.-JETP* **39**, 285–286 (1974).

Index for *MathLie* and *Mathematica* Functions

Subject Index