Algorithmic Computation of Symmetries, Invariants and Recursion Operators for Systems of Nonlinear Evolution and Differential-Difference Equations

by
Ünal GÖKTAŞ

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Ph.D. (Mathematical and Computer Sciences).

Golden, Colorado

Date April 7, 1998

Signed: Ü. Göktaş

Ünal GÖKTAŞ

Approved: [signature]

Willy HEREMAN
Professor
Thesis Advisor

Golden, Colorado

Date 4/7/98

Graeme Fairweather

Graeme FAIRWEATHER
Professor and Department Head
Mathematical and Computer Sciences

ABSTRACT

New straightforward algorithms for the symbolic computation of higher-order symmetries, conservation laws and recursion operators of nonlinear evolution equations and lattice equations are presented. The scaling properties of the evolution or lattice equations are used to determine the polynomial form of the symmetries, invariants, and recursion operators. The coefficients of these polynomials can be found by solving linear systems. The methods apply to polynomial systems of partial differential equations of first-order in time and arbitrary order in one space variable. Likewise, lattices must be of first order in time but may involve arbitrary shifts in the discretized space variable.

The recursion operator of an equation is the link between the symmetries of the equation. Therefore, the existence of a recursion operator is important to prove that the equation has infinitely many symmetries, which is an indicator of complete integrability. Our algorithm for finding the recursion operators of partial differential equations uses the knowledge of symmetries and conservation laws in connection with the scaling properties.

The algorithms for symmetries and conservation laws are implemented in *Mathematica* and can be used to test the integrability of both nonlinear evolution equations and semi-discrete lattice equations. With our *Integrability Package*, higher-order symmetries and invariants are obtained for several well-known systems of evolution and lattice equations. For partial differential equations and lattices with parameters, the code allows one to determine the conditions on these parameters so that a sequence of higher-order symmetries or invariants exists. The existence of a sequence of such symmetries and invariants is a predictor for integrability.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my thanks to Dr. Willy Hereman, my thesis advisor, for his helpful suggestions and great assistance in making this project possible.

Many thanks go to my thesis committee members, Dr. John Berger, Dr. Martin deHoop, Dr. John DeSanto, and Dr. Steven Pruess, for their comments and suggestions.

I greatly appreciate the generous financial aid from the Department of Mathematical and Computer Sciences which supported my studies. My research was also supported in part by the National Science Foundation under Grant CCR-9625421, and by Wolfram Research, Inc.

I acknowledge helpful discussions with Profs. J. Sanders and F. Verheest. I also thank Profs. B. Herbst, C. Nucci and Pavel Winternitz for their helpful comments.

Finally, my deepest and greatest gratitude goes to my family for their encouragement and support.

DEDICATION

To my wife, Esra

Chapter 1

INTRODUCTION

A large number of physically important nonlinear models are completely integrable, which means that they are solvable in terms of elementary functions or linearizable via an explicit transformation, or solvable via the Inverse Scattering Technique (IST) [1]. Integrable continuous or discrete models arise in key branches of physics including classical, quantum, particle, statistical, and plasma physics. Integrable equations also model wave phenomena in nonlinear optics and the bio-sciences. Mathematically, nonlinear models involve ordinary or partial differential equations (ODEs or PDEs), differential-difference equations (DDEs), integral equations, etc. [11].

Whichever form they come in, completely integrable equations exhibit analytic properties reflecting their rich mathematical structure. For instance, most completely integrable PDEs and DDEs possess infinitely many symmetries and conserved quantities (if the model is conservative). Perhaps after a suitable change of variables, the equations have the Painlevé property, admit Bäcklund and Darboux transformations, prolongation structures, or can be written in bi-Hamiltonian form [11].

An indication that certain evolution equations might have remarkable mathematical properties came with the discovery of an infinite number of conservation laws for the Korteweg-de Vries (KdV) equation, $u_t = 6uu_x + u_{3x}$. The conserved quantities u and u^2, corresponding to conservation of momentum and energy, respectively, were long known, and Whitham [70] had found a third one, $u^3 - \frac{1}{2}u_x^2$, which corresponds to Boussinesq's moment of instability. Zabusky and Kruskal found a fourth and fifth.

However, the search for additional conservation laws was halted due to a mistake in their computations. Miura eventually continued the search and, beyond the missing sixth, found additional three conservation laws [49]. It became clear that the KdV equation had an infinite sequence of conservation laws, later proven to be true.

The existence of an infinity of conservation laws was an important link in the discovery of other special properties of the KdV equation [74]. It led, for example, to the construction of the Miura transformation, which connects the solutions of the KdV and modified KdV equations. Consequently, the famous Lax pair was found, which associates a couple of linear equations to the KdV equation. From that, the IST for linearization of integrable equations was developed, and it was then shown that the KdV equation, and many other integrable equations, admit bi-Hamiltonian structures.

There are several motives to find symmetries and conservation laws explicitly. The first few conservation laws may have a physical meaning, such as conserved momentum and energy. Additional ones may facilitate the study of both quantitative and qualitative properties of solutions [32]. Furthermore, the existence of a sequence of symmetries and conservation laws predicts integrability. Yet, the nonexistence of conserved quantities does not preclude integrability. Indeed, integrable equations could be disguised with a coordinate transformation so that they no longer admit conservation laws of polynomial type. The same care should be taken in drawing conclusions about non-integrability based on the lack of higher-order symmetries. Another compelling argument relates to the numerical solution of PDEs. In numerical schemes the discrete conserved quantities should remain constant. In particular, the conservation of a positive definite quadratic quantity may prevent the occurrence of nonlinear instabilities in the numerical scheme. The use of conservation laws in PDE solvers has been discussed in [28, 43, 59].

The existence of an infinite hierarchy of symmetries for integrable equations can be established by explicitly constructing the recursion operator that connects such symmetries. Finding symmetries and recursion operators for nonlinear models is a nontrivial task, in particular if attempted with pen and paper. Computer algebra systems can greatly assist in the search for higher-order symmetries and recursion operators.

For PDEs, there are fairly complicated algorithms for the computation of symmetries and conservation laws. These include the algorithms designed by Bocharov and co-workers [5], Gerdt and Zharkov [16, 17], Fuchssteiner *et al.* [14], Sanders and Wang [55, 56, 57] and Wolf *et al.* [71]. Several methods to test the integrability of DDEs and for solving them are also available. Solution methods include symmetry reduction [45] and solving the spectral problem [44] on the lattice. Adaptations of the singularity confinement approach [53], the Wahlquist-Estabrook method [7], and symmetry techniques [6, 46, 63] also allow one to investigate integrability. The most comprehensive integrability study of nonlinear DDEs was done by Yamilov and co-workers (see e.g. [62, 73]). Their papers provide a classification of semi-discrete equations possessing infinitely many local conservation laws. Using the formal symmetry approach, they derive the necessary and sufficient conditions for the existence of local conservation laws, and provide an algorithm to construct them.

In contrast to these algorithms, in this thesis we present *new* direct algorithms that allow one to compute polynomial higher-order symmetries, conservation laws and recursion operators for polynomial PDEs in 1+1 dimension and polynomial DDEs (semi-discrete lattice equations). Our algorithms are fairly straightforward, and can be implemented in computer algebra languages.

The systems of PDEs or DDEs that our methods cover must be of evolution type, i.e. first order in (continuous) time. The number of equations and the order of differentiation (or shift level) in the spatial variable are arbitrary.

We use the dilation invariance of the given system of PDEs or DDEs to determine the form of the polynomial symmetry, invariant, or the recursion operator. Upon substitution of this form into the defining equation, one has to solve a linear system for the unknown constant coefficients of this form. In case the original system contains free parameters, the eliminant of that linear system will determine the necessary conditions for the parameters, so that the system admits the postulated generalized symmetry or invariant. Our algorithms can thus be used as an integrability test for classes of PDEs and DDEs involving parameters.

For the PDE case, a slight extension of our algorithms allows one to compute higher-order symmetries and invariants that *explicitly* depend on the independent variables. However, in such cases, it is necessary to specify the highest degree of the independent variables in the generalized symmetry and in the invariant.

Once the generalized symmetries and invariants are explicitly known, it is quite often possible to find the recursion operator by inspection [10]. If the recursion operator is *hereditary* then the equation will possess infinitely many symmetries. If the operator is hereditary and *factorizable* then the equation has infinitely many conserved quantities [11, 15].

For Lagrangian systems the set of higher-order symmetries can be shown to lead to the set of conservation laws [51, 54]. For equations without Lagrangian structure there is no universal correspondence between symmetries and conservation laws. The relationship between symmetries and conservation laws, as expressed through Noether's theorem, is beyond the scope of this thesis (see [54] for details).

The thesis is organized as follows. In Chapter 2, we give the definitions of a conservation law, a symmetry, and a recursion operator for a system of PDEs. Also we describe the dilation invariance, which is the key concept behind our algorithms. We also state a theorem from calculus of variations about the Euler-Lagrange equations, which plays a role in our algorithm for conservation laws. Chapter 3 is devoted to the description of our algorithm for the computation of polynomial type symmetries of evolution equations. Also we address how to handle symmetries that explicitly depend on the independent variables, wave equations, and nonuniform systems. In Chapter 4, we give a brief description of the algorithm to compute conservation laws of evolution equations. We do not give details of the algorithm, since it was presented in [18, 19]. A major application of our algorithm deals with systems with parameters. Our algorithms can be used to find the conditions on the parameters for the existence of symmetries and invariants. If for some choices of parameters, the system admits a sequence of symmetries and/or conservation laws, we may have detected an integrable system. In Chapter 5, we present such analyses and also we list the results for several other examples. In Chapter 6, our algorithm for the computation of recursion operators of evolution equations is described, followed by several examples. The algorithm uses the knowledge of symmetries and invariants of the evolution equations.

In Chapter 7, we turn to the DDE case. The definitions of a symmetry, a conservation law for a system of DDEs are given. Also, we describe the dilation invariance for DDEs, which is again the key concept behind our algorithms. Further in Chapter 7, we define an equivalence relationship, which is an essential tool for the computation of conservation laws. In Chapters 8 and 9, we describe our algorithms for the computation of symmetries and conservation laws of lattice equations, respectively.

Further in these chapters, we show how to handle nonuniform systems. In Chapter 10, we list our results for some lattice systems.

In Chapter 11, we briefly review the software packages related to the symbolic computation of higher-order symmetries, conservation laws and recursion operators. Apart from ours, all the other software works only for PDEs. We are not aware of any software for DDEs to compute higher-order symmetries and conservation laws.

Conclusions and an overview of future research are given in Chapter 12.

In summary, the original contribution of this thesis is twofold: new algorithms to test integrability of PDEs and DDEs, and their implementation in *Mathematica*.

The algorithms are based on a unifying concept which is applicable to both continuous and semi-discrete equations. Indeed, using the scaling invariance of the given equations, it is possible to explicitly construct symmetries, conservation laws, and recursion operators. The knowledge of all three is key for the study of the integrability of PDEs and DDEs.

In contrast to some of the existing algorithms and programs, ours can handle the computation of symmetries and conservation laws of parameterized equations, and systems that lack uniformity of rank. Furthermore, we can compute space and time dependent symmetries and conservation laws. In particular for systems with parameters, the necessary conditions for the existence of conservation laws and symmetries are automatically generated by our program.

The extension of the algorithms towards symmetries and conservation laws of DDEs is *new*. There are no other programs available for DDEs that automate these computations.

With respect to the implementation, we designed a comprehensive *Mathematica* program that automates the symbolic computation of symmetries and conservation laws of PDEs and DDEs. The algorithm for recursion operators of PDEs is still to be implemented.

Some of the results of this thesis have already been published [19, 20, 22, 24] in research journals, or under review [23].

Chapter 2

PDE CASE: DEFINITIONS AND THE KEY CONCEPT

In this chapter, we give the definitions of a conservation law, a symmetry, and a recursion operator for a system of PDEs. Also we describe dilation invariance, the key concept behind our algorithms. At the end of the chapter, we introduce a tool from the calculus of variations, the Euler operator, which is useful for the computation of conservation laws.

Consider a system of PDEs in the (single) space variable x and time variable t,

$$\mathbf{u}_t = \mathbf{F}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, ..., \mathbf{u}_{mx}), \tag{2.1}$$

where $\mathbf{u}$ and $\mathbf{F}$ are vector dynamical variables with the same number of components: $\mathbf{u} = (u_1, u_2, ..., u_n), \mathbf{F} = (F_1, F_2, ..., F_n)$ and $\mathbf{u}_{mx} = \mathbf{u}^{(m)} = \frac{\partial^m \mathbf{u}}{\partial x^m}$. The vector function $\mathbf{F}$ is assumed to be polynomial in $\mathbf{u}, \mathbf{u}_x, ..., \mathbf{u}_{mx}$. There are no restrictions on the order of the system or its degree of nonlinearity. If PDEs are of second or higher order in t, we assume that they can be recast in the form (2.1).

For simplicity of notation, unless we deal with a general case, the components of $\mathbf{u}$ will be denoted by $u, v, ...$ (instead of u_1, u_2, etc.).

2.1 Conservation Law

A *conservation law* is of the form

$$\mathrm{D}_t \rho + \mathrm{D}_x J = 0, \tag{2.2}$$

which is satisfied for all solutions of (2.1). The functional ρ is the *invariant* (conserved density), J is the associated *flux*. In general, both are, functions of $x, t, \mathbf{u}$, and its partial derivatives with respect to x [1]. Furthermore, D_t denotes the total derivative with respect to t; D_x the total derivative with respect to x. Specifically, ρ is a *local* invariant if ρ is a local functional of $\mathbf{u}$ and its derivatives, i.e. if the value of ρ at any x depends only on the values of $\mathbf{u}$ in an arbitrary small neighborhood of x. If J is also local, then (2.2) is a local conservation law. If ρ is a polynomial in $\mathbf{u}$, its x derivatives, and in x and t, then ρ is called a *polynomial* invariant. If J is also a polynomial, then (2.2) is called a polynomial conservation law. There is a close relationship between constants of motion and conservation laws. Indeed, for polynomial-type ρ and J, integration of (2.2) yields

$$P = \int_{-\infty}^{+\infty} \rho \, dx = \text{constant}, \tag{2.3}$$

provided that J vanishes at infinity. For ODEs, the P's are called constants of motion.

2.2 Symmetry

A vector function $\mathbf{G}(x, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, \ldots)$, with $\mathbf{G} = (G_1, G_2, \ldots, G_n)$, is called a *symmetry* of (2.1) if and only if it leaves (2.1) invariant for the replacement $\mathbf{u} \rightarrow \mathbf{u} + \epsilon\mathbf{G}$ within order ϵ. Hence,

$$\mathrm{D}_t(\mathbf{u} + \epsilon\mathbf{G}) = \mathbf{F}(\mathbf{u} + \epsilon\mathbf{G}) \tag{2.4}$$

must hold up to order ϵ on the solutions of (2.1). Consequently, $\mathbf{G}$ must satisfy the linearized equation [9, 11]

$$\mathrm{D}_t\mathbf{G} = \mathbf{F}'(\mathbf{u})[\mathbf{G}], \tag{2.5}$$

where $\mathbf{F}'$ is the Fréchet derivative of $\mathbf{F}$, i.e.,

$$\mathbf{F}'(\mathbf{u})[\mathbf{G}] = \frac{\partial}{\partial \epsilon}\mathbf{F}(\mathbf{u} + \epsilon\mathbf{G})_{|\epsilon=0}. \tag{2.6}$$

In (2.4) and (2.6) we infer that $\mathbf{u}$ is replaced by $\mathbf{u} + \epsilon\mathbf{G}$, and $\mathbf{u}_{nx}$ by $\mathbf{u}_{nx} + \epsilon \mathrm{D}_x^n \mathbf{G}$.

Symmetries of the form $\mathbf{G}(x, t, \mathbf{u})$ are called *point* symmetries. Symmetries of the form $\mathbf{G}(x, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_t)$ are called *classical* or Lie-Bäcklund symmetries, and all other symmetries involving higher derivatives than the first are called generalized or *higher-order* symmetries [47].

Example 2.1 The most famous evolution equation from soliton theory, the KdV equation [48],

$$u_t = 6uu_x + u_{3x}, \tag{2.7}$$

is known to have infinitely many polynomial symmetries and conservation laws [10].

The KdV equation is a member of the well-known Lax hierarchy. Each member of this infinite Lax family corresponds to a symmetry of the KdV equation, and the first four symmetries are:

$$G^{(1)} = u_x, \quad G^{(2)} = 6uu_x + u_{3x}, \quad G^{(3)} = 30u^2u_x + 20u_xu_{2x} + 10uu_{3x} + u_{5x}, \tag{2.8}$$
$$G^{(4)} = 140u^3u_x + 70u_x^3 + 280uu_xu_{2x} + 70u^2u_{3x} + 70u_{2x}u_{3x} + 42u_xu_{4x} + 14uu_{5x} + u_{7x}.$$

The evolution equations $u_t = G^{(i)}, i = 1, 2, 3, ...$, which are completely integrable, constitute the Lax hierarchy. Also, the first three conservation laws of (2.7) are

$$\mathrm{D}_t(u) - \mathrm{D}_x\left(3u^2 + u_{2x}\right) = 0,$$
$$\mathrm{D}_t\left(u^2\right) - \mathrm{D}_x\left(4u^3 - u_x^2 + 2uu_{2x}\right) = 0,$$

$$\mathrm{D}_t \left(u^3 - \frac{1}{2} u_x^2 \right) - \mathrm{D}_x \left(\frac{9}{2} u^4 - 6 u u_x^2 + 3 u^2 u_{2x} + \frac{1}{2} u_{2x}^2 - u_x u_{3x} \right) = 0. \quad (2.9)$$

The first two express conservation of momentum and energy, respectively. They are easy to compute by hand. The third one, less obvious and requiring more work, corresponds to Boussinesq's moment of instability [49].

2.3 Key Concept: Dilation Invariance

Observe that (2.7), its symmetries (2.8), and its conservation laws (2.9) are all invariant under the dilation (scaling) symmetry

$$(t, x, u) \rightarrow (\lambda^{-3} t, \lambda^{-1} x, \lambda^2 u), \quad (2.10)$$

where λ is an arbitrary parameter. The result of this dimensional analysis of (2.7) can be stated as follows: u corresponds to two derivatives with respect to x, for short, $u \sim \mathrm{D}_x^2$. Similarly, $\mathrm{D}_t \sim \mathrm{D}_x^3$. Scaling invariance, which is a special Lie-point symmetry, is an intrinsic property of many integrable nonlinear PDEs and DDEs.

As we will show in the coming chapters, our algorithms exploit this scaling invariance to find symmetries, invariants and recursion operators.

To describe an algorithmic way for determining the scaling invariance, we require additional definitions. The *weight*, w, of a variable is by definition equal to the number of derivatives with respect to x that variable carries. Weights are rational, and weights of dependent variables are nonnegative. We set $w(\mathrm{D}_x) = 1$. In view of (2.10), we have $w(u) = 2$ and $w(\mathrm{D}_t) = 3$. Consequently, $w(x) = -1$ and $w(t) = -3$.

The *rank* of a monomial is defined as the total weight of the monomial, again in terms of derivatives with respect to x. Observe that (2.7) is an equation of rank

5, since all the terms (monomials) have the same rank, namely 5. This property is called *uniformity in rank.*

Conversely, requiring uniformity in rank for (2.7) allows one to compute the weights of the dependent variables by solving a linear system. Indeed, with $w(\mathrm{D}_x) = 1$ we have

$$w(u) + w(\mathrm{D}_t) = 2w(u) + 1 = w(u) + 3,$$

which yields $w(u) = 2$, $w(\mathrm{D}_t) = 3$. Hence, $w(t) = -3$, which is consistent with (2.10).

2.4 Recursion Operator

A *recursion operator* [52] for (2.1) is a linear operator $\mathbf{\Phi}$ in the space of differential functions with the property that whenever $\mathbf{G}$ is a symmetry of (2.1), so is $\hat{\mathbf{G}}$ with $\hat{\mathbf{G}} = \mathbf{\Phi}\mathbf{G}$. For n-component systems, $\mathbf{\Phi}$ is an $n \times n$ matrix.

Example 2.2 The recursion operator [52] for the KdV equation is given by

$$\Phi = \mathrm{D}_x^2 + 2u + 2\mathrm{D}_x u \mathrm{D}_x^{-1} = \mathrm{D}_x^2 + 4u + 2u_x \mathrm{D}_x^{-1}, \qquad (2.11)$$

where D_x^{-1} is the integration operator. This operator is hereditary [15] and connects the symmetries (2.8) of the KdV equation (2.7).

For example,

$$\begin{aligned}
\Phi u_x &= (\mathrm{D}_x^2 + 2u + 2\mathrm{D}_x u \mathrm{D}_x^{-1}) u_x = 6uu_x + u_{3x}, \\
\Phi(6uu_x + u_{3x}) &= (\mathrm{D}_x^2 + 2u + 2\mathrm{D}_x u \mathrm{D}_x^{-1})(6uu_x + u_{3x}) \\
&= 30u^2 u_x + 20u_x u_{2x} + 10uu_{3x} + u_{5x},
\end{aligned}$$

and so forth.

2.5 A Tool: Euler Operator

We introduce a tool from calculus of variations, the Euler operator, which is very useful for testing if an expression is a total derivative [52], without having to carry out any integrations by parts. In Chapter 4, we will use this tool for the computation of conservation laws.

Theorem 2.1 *If* $f = f(x, u_1, ..., u_1^{(m)}, \ldots, u_n, ..., u_n^{(m)})$, *then* $\mathcal{E}_{\mathbf{u}}(f) \equiv \mathbf{0}$, *if and only if* $f = \mathrm{D}_x g$, *where* $g = g(x, u_1, ..., u_1^{(m-1)}, \ldots, u_n, ..., u_n^{(m-1)})$.

In this theorem, for which a proof can be found in [52],

$$\mathbf{u} = (u_1, ..., u_n), \;\; \mathcal{E}_{\mathbf{u}}(f) = (\mathcal{E}_{u_1}(f), ..., \mathcal{E}_{u_n}(f)), \;\; \mathbf{0} = (0, ..., 0),$$

and

$$\mathcal{E}_{u_i} = \frac{\partial}{\partial u_i} - \frac{d}{dx}(\frac{\partial}{\partial u_i{}'}) + \frac{d^2}{dx^2}(\frac{\partial}{\partial u_i{}''}) + \cdots + (-1)^k \frac{d^k}{dx^k}(\frac{\partial}{\partial u_i{}^{(k)}}), \tag{2.12}$$

is the *Euler operator* (or variational derivative).

To show the usefulness of this operator in verifying if a given function is the derivative of another function, let us consider an example.

Example 2.3 Suppose that

$$f(u, u', u''', u^{(4)}) = u'(x)u''(x)^2 + 2u(x)u''(x)u^{(3)}(x) - 3u'(x)^2 u''(x)u^{(3)}(x) - u'(x)^3 u^{(4)}(x)$$

is given. We have chosen f so that it is the derivative with respect to x of the function

$$g(u, u', u'', u^{(3)}) = u(x)u''(x)^2 - u'(x)^3 u^{(3)}(x).$$

Indeed, one can easily verify that $f = \frac{d}{dx}g$. Now, in order to see that Theorem 2.1 is indeed true, we explicitly compute all the relevant terms. For this example $n = 4$; therefore, we need:

$$\begin{aligned}
\frac{\partial f}{\partial u} &= 2u''(x)u^{(3)}(x), \\
\frac{d}{dx}(\frac{\partial f}{\partial u'}) &= 2u''(x)u^{(3)}(x) - 6u''(x)^2u^{(3)}(x) - 6u'(x)u^{(3)}(x)^2 \\
&\quad -12u'(x)u''(x)u^{(4)}(x) - 3u'(x)^2u^{(5)}(x), \\
\frac{d^2}{dx^2}(\frac{\partial f}{\partial u''}) &= 8u''(x)u^{(3)}(x) - 6u''(x)^2u^{(3)}(x) - 6u'(x)u^{(3)}(x)^2 + 6u'(x)u^{(4)}(x) \\
&\quad -12u'(x)u''(x)u^{(4)}(x) + 2u(x)u^{(5)}(x) - 3u'(x)^2u^{(5)}(x), \\
\frac{d^3}{dx^3}(\frac{\partial f}{\partial u'''}) &= 8u''(x)u^{(3)}(x) - 36u''(x)^2u^{(3)}(x) - 18u'(x)u^{(3)}(x)^2 + 6u'(x)u^{(4)}(x) \\
&\quad -24u'(x)u''(x)u^{(4)}(x) + 2u(x)u^{(5)}(x) - 3u'(x)^2u^{(5)}(x), \\
\frac{d^4}{dx^4}(\frac{\partial f}{\partial u^{(4)}}) &= -36u''(x)^2u^{(3)}(x) - 18u'(x)u^{(3)}(x)^2 - 24u'(x)u''(x)u^{(4)}(x) - 3u'(x)^2u^{(5)}(x).
\end{aligned}$$

Substitution of the right hand sides into

$$\frac{\partial f}{\partial u} - \frac{d}{dx}(\frac{\partial f}{\partial u'}) + \frac{d^2}{dx^2}(\frac{\partial f}{\partial u''}) - \frac{d^3}{dx^3}(\frac{\partial f}{\partial u^{(3)}}) + \frac{d^4}{dx^4}(\frac{\partial f}{\partial u^{(4)}})$$

and simplification indeed gives zero.

Chapter 3

SYMMETRIES OF EVOLUTION EQUATIONS

In this chapter, we describe our algorithm for the computation of polynomial symmetries of evolution equations. Also we address how to compute $x - t$ dependent symmetries, and how to handle wave equations, and nonuniform systems.

Recall that for a system of evolutionary PDEs,

$$\mathbf{u}_t = \mathbf{F}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, ..., \mathbf{u}_{mx}), \tag{3.1}$$

a vector function $\mathbf{G}(x, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, ...)$, with $\mathbf{G} = (G_1, G_2, ..., G_n)$, is called a *symmetry* of (3.1) if and only if $\mathbf{G}$ satisfies the linearized equation [9, 11]

$$\mathrm{D}_t\mathbf{G} = \mathbf{F}'(\mathbf{u})[\mathrm{G}], \tag{3.2}$$

where $\mathbf{F}'$ is the Fréchet derivative of $\mathbf{F}$, as defined in (2.6).

3.1 Algorithm

To illustrate our algorithm, we consider the KdV equation (2.7) with scaling properties such that $w(u) = 2$.

Our algorithm exploits this scaling invariance to find higher-order symmetries. The algorithm has two steps.

3.1.1 Step 1: Construct the form of the symmetry

This step involves finding the building blocks (monomials) of a polynomial symmetry with prescribed rank. All terms in the symmetry must have the same rank. Since we may introduce parameters with weights (see Section 3.3), the fact that the symmetry will be a sum of monomials of uniform rank does not necessarily imply that the symmetry must be uniform in rank with respect to the dependent variables.

As an example, let us compute the form of the symmetry of (2.7) with rank 7. Start by listing all powers in u with rank 7 or less: $\mathcal{L}=\{1, u, u^2, u^3\}$. Next, for each monomial in $\mathcal{L}$, introduce enough x-derivatives, so that each term has exactly rank 7. Thus,

$$\mathrm{D}_x(u^3) = 3u^2 u_x, \quad \mathrm{D}_x^3(u^2) = 6u_x u_{2x} + 2uu_{3x}, \quad \mathrm{D}_x^5(u) = u_{5x}, \quad \mathrm{D}_x^7(1) = 0.$$

Then, gather the resulting (non-zero) terms in a set $\mathcal{R} = \{u^2 u_x, u_x u_{2x}, uu_{3x}, u_{5x}\}$, which contains the building blocks of the symmetry. Linear combination of the monomials in $\mathcal{R}$ with constant coefficients c_i gives the form of the symmetry:

$$G = c_1\, u^2 u_x + c_2\, u_x u_{2x} + c_3\, uu_{3x} + c_4\, u_{5x}. \tag{3.3}$$

3.1.2 Step 2: Determine the unknown coefficients in the symmetry

We determine the coefficients c_i by requiring that (3.2) holds on the solutions of (3.1). Compute $\mathrm{D}_t \mathrm{G}$ and use (3.1) to remove $\mathbf{u}_t, \mathbf{u}_{tx}, \mathbf{u}_{txx}$, etc. For given $\mathbf{F}$, compute the Fréchet derivative, and in view of (3.2), equate the resulting expressions. Treating the different monomial terms in $\mathbf{u}$ and its x-derivatives as independent, the linear system for the coefficients c_i is readily obtained.

For (2.7), we perform this computation with $F = 6uu_x + u_{3x}$ and G in (3.3). Considering as independent all products and powers of $u, u_x, u_{xx}, \ldots$, in

$$(12c_1 - 18c_2)u_x^2 u_{2x} + (6c_1 - 18c_3)uu_{2x}^2 + (6c_1 - 18c_3)uu_x u_{3x} + (3c_2 - 60c_4)u_{3x}^2$$
$$+(3c_2 + 3c_3 - 90c_4)u_{2x}u_{4x} + (3c_3 - 30c_4)u_x u_{5x} \equiv 0, \tag{3.4}$$

we obtain the linear system for the coefficients c_i :

$\mathcal{S} = \{12c_1 - 18c_2 = 0, 6c_1 - 18c_3 = 0, 3c_2 - 60c_4 = 0, 3c_2 + 3c_3 - 90c_4 = 0, 3c_3 - 30c_4 = 0\}$. The solution is $\frac{c_1}{30} = \frac{c_2}{20} = \frac{c_3}{10} = c_4$. Since symmetries can only be determined up to a multiplicative constant, we choose $c_4 = 1$, then $c_1 = 30, c_2 = 20, c_3 = 10$, and substitute this into (3.3). Hence,

$$G = 30u^2 u_x + 20u_x u_{2x} + 10uu_{3x} + u_{5x}. \tag{3.5}$$

Note that $u_t = G$ is known as the Lax equation, which is the fifth-order PDE in the completely integrable KdV hierarchy [42].

Analogously, for (2.7) we computed the ($x-t$ independent) symmetries of rank ≤ 11. They are:

$$G^{(1)} = u_x, \quad G^{(2)} = 6uu_x + u_{3x}, \quad G^{(3)} = 30u^2 u_x + 20u_x u_{2x} + 10uu_{3x} + u_{5x},$$
$$G^{(4)} = 140u^3 u_x + 70u_x^3 + 280uu_x u_{2x} + 70u^2 u_{3x} + 70u_{2x}u_{3x} + 42u_x u_{4x} + 14uu_{5x} + u_{7x},$$
$$G^{(5)} = 630u^4 u_x + 1260uu_x^3 + 2520u^2 u_x u_{2x} + 1302u_x u_{2x}^2 + 420u^3 u_{3x} + 966u_x^2 u_{3x}$$
$$+1260uu_{2x}u_{3x} + 756uu_x u_{4x} + 252u_{3x}u_{4x} + 126u^2 u_{5x} + 168u_{2x}u_{5x}$$
$$+72u_x u_{6x} + 18uu_{7x} + u_{9x}. \tag{3.6}$$

These results agree with those listed in the literature (see e.g. [10, 47, 52]).

Remark 3.1 Instead of working with the definition (3.2) of the symmetry, one could introduce an auxiliary evolution equation,

$$\mathbf{u}_\tau = \mathbf{G}(x, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, \ldots), \tag{3.7}$$

which defines the flow generated by $\mathbf{G}$ and parameterized by the auxiliary time variable τ. The symmetry can then be computed from the compatibility condition of (3.1) and (3.7):

$$\mathrm{D}_\tau \mathbf{F}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, \ldots, \mathbf{u}_{mx}) = \mathrm{D}_t \mathbf{G}(x, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, \ldots). \tag{3.8}$$

One then proceeds as follows: as above, determine the form of the symmetry $\mathbf{G}$ involving the constant coefficients c_i. Then, compute $\mathrm{D}_t\mathbf{G}$ and use (3.1) to remove $\mathbf{u}_t, \mathbf{u}_{tx}$, etc. Subsequently, compute $\mathrm{D}_\tau\mathbf{F}$ and use (3.7) to remove $\mathbf{u}_\tau, \mathbf{u}_{\tau x}$, etc. Finally, use (3.8) to determine the linear system for the unknown c_i. Solve the system and substitute the result into the form of $\mathbf{G}$.

Applied to (2.7), $\mathrm{D}_t G$ is computed with G in (3.3). Next, (2.7) is used to eliminate all t-derivatives of u from the expression of $\mathrm{D}_t G$. Then, compute $\mathrm{D}_\tau F$ with F in the right hand side of (2.7), and eliminate all τ-derivatives through (3.7) after substitution of (3.3). Finally, expressing that $\mathrm{D}_\tau F - D_t G \equiv 0$ leads to (3.4).

Although this procedure [35] circumvents the evaluation of the Fréchet derivative, it seems more involved than our algorithm which uses the definition (3.2).

3.2 Symmetries Explicitly Dependent on x and t

The KdV equation (2.7) has also polynomial symmetries that *explicitly* depend on x and t. Our algorithm can be used to find these symmetries provided that we specify the maximum degree in x and t.

As an example, we will compute the symmetry of rank 2 for (2.7) that linearly depends on x and/or t. In other words, the highest degree in x or t in the symmetry is 1.

We start with the list of monomials in u, x and t of rank 2 or less:

$$\mathcal{L} = \{1, u, x, xu, t, tu, tu^2\}.$$

Then, for each monomial in $\mathcal{L}$, introduce enough x-derivatives so that each term exactly has weight 2. Thus,

$$\mathrm{D}_x(xu) = u + xu_x,\ \mathrm{D}_x(tu^2) = 2tuu_x,\ \mathrm{D}_x^3(tu) = tu_{3x},\ \mathrm{D}_x^2(1) = \mathrm{D}_x^3(x) = \mathrm{D}_x^5(t) = 0.$$

Gather the non-zero resulting terms in a set $\mathcal{R} = \{u, xu_x, tuu_x, tu_{3x}\}$, which contains the building blocks of the symmetry. A linear combination of the monomials in $\mathcal{R}$ with constant coefficients c_i gives the form of the symmetry:

$$G = c_1\, u + c_2\, xu_x + c_3\, tuu_x + c_4\, tu_{3x}. \tag{3.9}$$

Now, determine the coefficients c_1 through c_4 by requiring that (3.2) holds on the solutions of (2.7). After grouping the terms, one gets

$$(6c_1 + 6c_2 - c_3)uu_x + (3c_3 - 18c_4)tu_{2x}^2 + (3c_2 - c_4)u_{3x} + (3c_3 - 18c_4)tu_xu_{3x} \equiv 0,$$

which yields

$$\mathcal{S} = \{6c_1 + 6c_2 - c_3 = 0, 3c_3 - 18c_4 = 0, 3c_2 - c_4 = 0\}. \tag{3.10}$$

The solution is $\frac{3c_1}{2} = 3c_2 = \frac{c_3}{6} = c_4$. We choose $c_4 = 1$, consequently $c_1 = \frac{2}{3}, c_2 = \frac{1}{3}, c_3 = 6$, and substitute this into (3.9). Hence,

$$G = \frac{2}{3}u + \frac{1}{3}xu_x + 6tuu_x + tu_{3x}. \tag{3.11}$$

We computed two symmetries of (2.7) that linearly depend on x and t. They are of rank 0 and 2 :

$$G = 1 + 6tu_x, \quad \text{and} \quad G = \frac{1}{3}(2u + xu_x) + tu_t = \frac{1}{3}(2u + xu_x) + t(6uu_x + u_{3x}). \tag{3.12}$$

Our results agree with those in the literature [47].

3.3 Nonuniform Systems

For scaling invariant equations such as (2.7), it suffices to consider the dilation symmetry on the space of independent and dependent variables. For PDE systems that are inhomogeneous under a suitable scaling symmetry, such as the example given below, we use the following trick: we introduce one (or more) auxiliary parameter(s) with an appropriate scaling. These extra parameters can be viewed as additional dependent variables, however, their derivatives are zero. By extending the action of the dilation symmetry to the space of independent and dependent variables, *including* the parameters, we are able to apply our algorithm to a larger class of polynomial PDE systems.

3.3.1 Example: Boussinesq Equation

Consider the wave equation,

$$u_{tt} - u_{2x} + 3uu_{2x} + 3u_x^2 + \alpha u_{4x} = 0, \tag{3.13}$$

(α constant) which was proposed by Boussinesq to describe surface water waves whose horizontal scale is much larger than the depth of the water [1].

To apply our algorithm, we must first rewrite (3.13) as a first-order system,

$$u_t = v_x, \quad v_t = u_x - 3uu_x - \alpha u_{3x}, \tag{3.14}$$

where v is an auxiliary dependent variable. It is easy to verify that the terms u_x and αu_{3x} in the second equation obstruct uniformity in rank. To circumvent the problem we introduce an auxiliary parameter β with (unknown) weight, and replace (3.14) by

$$u_t = v_x, \quad v_t = \beta u_x - 3uu_x - \alpha u_{3x}. \tag{3.15}$$

As described in Chapter 2, we compute the weights from

$$\begin{aligned} w(u) + w(\mathrm{D}_t) &= w(v) + 1, \\ w(v) + w(\mathrm{D}_t) &= w(\beta) + w(u) + 1 = 2w(u) + 1 = w(u) + 3. \end{aligned}$$

This yields

$$w(u) = 2,\ w(v) = 3,\ w(\beta) = 2, \text{ and } w(\mathrm{D}_t) = -w(t) = 2, \tag{3.16}$$

and the scaling properties of (3.15) are $u \sim \beta \sim \mathrm{D}_t \sim \mathrm{D}_x^2$, $v \sim \mathrm{D}_x^3$. Indeed, (3.15) is invariant under the dilation symmetry

$$(x, t, u, v, \beta) \rightarrow (\lambda^{-1}x, \lambda^{-2}t, \lambda^2 u, \lambda^3 v, \lambda^2 \beta). \tag{3.17}$$

Observe that all the monomials in the equations in (3.15) have rank 4 and 5. Therefore, for any symmetry **G** of (3.15),

$$\text{rank}(G_2) = \text{rank}(G_1) + 1 = \text{rank}(G_1) + w(v) - w(u). \tag{3.18}$$

Let us construct the form of the symmetry $\mathbf{G} = (G_1, G_2)$ with $\text{rank}(G_1) = 6$ and $\text{rank}(G_2) = 7$. First, list all monomials in u, v and β of rank 6 (respectively rank 7) or less:

$$\begin{aligned}
\mathcal{L}_1 &= \{1, \beta, \beta^2, \beta^3, u, \beta u, \beta^2 u, u^2, \beta u^2, u^3, v, \beta v, uv, v^2\}, \\
\mathcal{L}_2 &= \{1, \beta, \beta^2, \beta^3, u, \beta u, \beta^2 u, u^2, \beta u^2, u^3, v, \beta v, \beta^2 v, uv, \beta uv, u^2 v, v^2\}.
\end{aligned}$$

Next, for each monomial in $\mathcal{L}_1$ and $\mathcal{L}_2$, introduce the necessary x-derivatives, so that each term in $\mathcal{L}_1$ exactly has rank 6, and each term in $\mathcal{L}_2$ has rank 7. Keeping in mind that β is constant, and proceeding with the rest of the algorithm, we obtain:

$$\begin{aligned}
G_1^{(1)} &= u_x v + u v_x + \frac{2}{3}\alpha v_{3x}, \\
G_2^{(1)} &= \beta u u_x - 3u^2 u_x + v v_x - 6\alpha u_x u_{2x} + \frac{2}{3}\alpha\beta u_{3x} - 3\alpha u u_{3x} - \frac{2}{3}\alpha^2 u_{5x}.
\end{aligned} \tag{3.19}$$

Finally, setting $\beta = 1$ in (3.19), one obtains a symmetry of (3.14) although initially this system was not uniform in rank. We list one more higher-order symmetry of

(3.14):

$$\begin{aligned}
G_1^{(2)} &= uu_x - \frac{3}{2}u^2u_x + vv_x - 5\alpha u_x u_{2x} + \frac{2}{3}\alpha u_{3x} - 2\alpha u u_{3x} - \frac{8}{15}\alpha^2 u_{5x}, \\
G_2^{(2)} &= uv_x + vu_x - 3uu_x v - \frac{3}{2}u^2 v_x - 2\alpha u_{2x} v_x - 3\alpha u_x v_{2x} - \alpha u_{3x} v + \frac{2}{3}\alpha v_{3x} \\
&\quad -2\alpha u v_{3x} - \frac{8}{15}\alpha^2 v_{5x}.
\end{aligned} \tag{3.20}$$

Chapter 4

INVARIANTS OF EVOLUTION EQUATIONS

In this chapter, we describe the algorithm to compute conservation laws of evolution equations. We will use our tool from calculus of variations, the Euler operator. We do not give details of the algorithm here, since it was presented in [18, 19].

Recall that for a system of evolutionary PDEs

$$\mathbf{u}_t = \mathbf{F}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, ..., \mathbf{u}_{mx}), \tag{4.1}$$

a *conservation law* is of the form

$$\mathrm{D}_t\rho + \mathrm{D}_x J = 0, \tag{4.2}$$

which is satisfied for all solutions of (4.1). The functional ρ is the *invariant*, J is the associated *flux*.

4.1 Algorithm

To illustrate our algorithm, once again we consider the KdV equation (2.7). Keeping in mind the scaling properties of (2.7) with $w(u) = 2$, we now start the description of our algorithm in two steps.

4.1.1 Step 1: Construct the form of the invariant

As an example, let us compute the form of the invariant of rank 6. Start by listing all powers in u with rank 6 or less: $\mathcal{L}=\{1, u, u^2, u^3\}$. Next, for each monomial in $\mathcal{L}$, introduce the necessary x-derivatives, so that each term has exactly rank 6. Thus,

$$\mathrm{D}_x^0(u^3) = u^3, \quad \mathrm{D}_x^2(u^2) = 2u_x^2 + 2uu_{2x}, \quad \mathrm{D}_x^4(u) = u_{4x}, \quad \mathrm{D}_x^6(1) = 0.$$

Then, gather the resulting (non-zero) monomials in $\mathcal{M} = \{u^3, u_x^2, uu_{2x}\}$. Removing the terms that are total derivatives with respect to x or total derivative up to terms kept earlier in $\mathcal{M}$, results in $\mathcal{R} = \{u^3, u_x^2\}$, which has the building blocks of the invariant. Notice that, since $uu_{2x} = (uu_x)_x - u_x^2$, we have removed uu_{2x} from $\mathcal{M}$. Linear combination of the monomials in $\mathcal{R}$ with constant coefficients c_i gives the form of the invariant:

$$\rho = c_1\, u^3 + c_2\, u_x^2. \tag{4.3}$$

4.1.2 Step 2: Determine the unknown coefficients in the invariant

We determine the coefficients c_i by requiring that (4.2) holds on the solutions of (4.1). Compute $\mathrm{D}_t\rho$ and use (4.1) to remove $\mathbf{u}_t, \mathbf{u}_{tx}, \mathbf{u}_{txx}$, etc. Then apply the Euler operator (2.12) and require that the resulting expression vanishes. For our example, this means

$$\mathcal{E}_u(\mathrm{D}_t\rho) = -18(c_1 + 2c_2)u_x u_{2x} \equiv 0,$$

and we obtain the linear system for the coefficients c_i : $\mathcal{S} = \{c_1 + 2c_2 = 0\}$. The solution is $c_1 = -2c_2$. Since invariants can only be determined up to a multiplicative constant, we choose $c_2 = 1$ and so $c_1 = -2$. Substituting this into (4.3) gives

$$\rho = -2u^3 + u_x^2. \tag{4.4}$$

Analogously, we computed more invariants of (2.7). Here, we list the first seven:

$$\begin{aligned}
\rho_1 &= u, \qquad \rho_2 = u^2, \qquad \rho_3 = -2u^3 + u_x^2, \\
\rho_4 &= 5u^4 - 10uu_x^2 + u_{2x}^2, \qquad \rho_5 = -14u^5 + 70u^2u_x^2 - 14uu_{2x}^2 + u_{3x}^2, \\
\rho_6 &= 42u^6 - 420u^3u_x^2 - 35u_x^4 + 126u^2u_{2x}^2 + 20u_{2x}^3 - 18uu_{3x}^2 + u_{4x}^2, \\
\rho_7 &= -132u^7 + 2310u^4u_x^2 + 770uu_x^4 - 924u^3u_{2x}^2 - 462u_x^2u_{2x}^2 - 440uu_{2x}^3 \\
&\quad +198u^2u_{3x}^2 + 110u_{2x}u_{3x}^2 - 22uu_{4x}^2 + u_{5x}^2.
\end{aligned} \tag{4.5}$$

Note that the forms of invariants are not unique. Some of the terms in invariants can always be integrated by parts to obtain *equivalent* forms, modulo total derivatives. For example, ρ_3 can be replaced by $\tilde{\rho}_3 = -2u^3 - uu_{2x}$, since $\rho_3 - \tilde{\rho}_3 = \mathrm{D}_x(uu_x)$.

4.2 Invariants Explicitly Dependent on x and t

The KdV equation (2.7) has also a single invariant which *explicitly* depends on x and t. Again, our algorithm can be used to find this invariant provided that we specify the maximum degree in x and t.

Doing so, the single invariant of KdV (2.7) with linear explicit dependency

$$\rho = tu^2 + \frac{1}{3}xu \tag{4.6}$$

readily follows. Observe that ρ is of rank 1.

4.3 Nonuniform Systems

For systems that are inhomogeneous under a suitable scaling symmetry, we can again use our trick, and introduce one (or more) auxiliary parameter(s) with an appropriate scaling.

Example 4.1 Working with the Boussinesq equation (3.13), rewritten as a first-order system (3.14), we were able to compute many invariants. The first four invariants are:

$$\begin{aligned} \rho_1 &= u, & \rho_2 &= v, \\ \rho_3 &= uv, & \rho_4 &= u^2 - u^3 + v^2 + \alpha u_x^2. \end{aligned} \tag{4.7}$$

Conservation laws play a key role in the study of this wave equation. They can be used to prove that solutions are bounded for certain sets of initial conditions [28], or, conversely, to show that solutions fail to exist after a finite time.

Chapter 5

PDE CASE: APPLICATIONS AND EXAMPLES

A major application of our algorithm deals with the investigation of integrability of systems with parameters. Our algorithms can be used to find the conditions on the parameters so that symmetries and invariants exist. If for some choices of parameters, the system admits a sequence of symmetries and/or conservation laws, we may have detected an integrable system. In this chapter, we present such analyses and also list our results for several other examples. For more examples we refer to [19].

5.1 Fifth-Order Korteweg-de Vries Equations

Consider the parameterized family of fifth-order equations,

$$u_t + \alpha u^2 u_x + \beta u_x u_{2x} + \gamma u u_{3x} + u_{5x} = 0, \tag{5.1}$$

where α, β, γ are nonzero constants. Integrable cases of (5.1) are well known in the literature [12, 30, 41, 60]. Indeed, for $\alpha = 30, \beta = 20, \gamma = 10$, equation (5.1) reduces to the Lax equation [42]. The SK equation, due to Sawada and Kotera [61], and Dodd and Gibbon [8] is obtained for $\alpha = 5, \beta = 5, \gamma = 5$. The KK equation, due to Kaup [39] and Kupershmidt [40], corresponds to $\alpha = 20, \beta = 25, \gamma = 10$. Although they only differ by the values of their parameters, the named equations have vastly different properties.

The scaling properties of (5.1) are such that $w(u) = 2, w(\mathrm{D}_t) = 5$. Using our algorithm, one easily computes the *compatibility conditions* for the parameters α, β

and γ, so that (5.1) admits a symmetry of fixed rank. The results are:

Rank 3: $G = u_x$ is a symmetry of (5.1) without any conditions on the parameters.

Rank 5: $G = uu_x + \frac{5}{3\gamma}u_{3x}$ is a symmetry of (5.1) provided that

$$\alpha = \frac{3}{10}\gamma^2, \text{ and } \beta = 2\gamma. \tag{5.2}$$

The Lax equation satisfies (5.2). Since the KdV equation is a member of Lax hierarchy, condition (5.2) comes as no surprise.

Rank 7: Equation (5.1) is of rank 7. The stationary part of (5.1) is the symmetry.

Rank 9: Three branches emerge:

(i) If condition (5.2) holds then

$$\begin{aligned} G &= u^3u_x + \frac{5}{\gamma}u_x^3 + \frac{20}{\gamma}uu_xu_{2x} + \frac{5}{\gamma}u^2u_{3x} + \frac{50}{\gamma^2}u_{2x}u_{3x} + \frac{30}{\gamma^2}u_xu_{4x} \\ &+ \frac{10}{\gamma^2}uu_{5x} + \frac{50}{7\gamma^3}u_{7x}. \end{aligned} \tag{5.3}$$

(ii) If

$$\alpha = \frac{1}{5}\gamma^2, \text{ and } \beta = \gamma \tag{5.4}$$

holds, one has the symmetry

$$\begin{aligned} G &= u^3u_x + \frac{15}{4\gamma}u_x^3 + \frac{45}{2\gamma}uu_xu_{2x} + \frac{15}{2\gamma}u^2u_{3x} + \frac{225}{4\gamma^2}u_{2x}u_{3x} + \frac{75}{2\gamma^2}u_xu_{4x} \\ &+ \frac{75}{4\gamma^2}uu_{5x} + \frac{375}{28\gamma^3}u_{7x}. \end{aligned} \tag{5.5}$$

The SK equation satisfies the condition (5.4).

(iii) One has the symmetry

$$\begin{aligned} G &= u^3u_x + \frac{75}{8\gamma}u_x^3 + \frac{135}{4\gamma}uu_xu_{2x} + \frac{15}{2\gamma}u^2u_{3x} + \frac{225}{2\gamma^2}u_{2x}u_{3x} + \frac{525}{8\gamma^2}u_xu_{4x} \\ &+ \frac{75}{4\gamma^2}uu_{5x} + \frac{375}{28\gamma^3}u_{7x} \end{aligned} \tag{5.6}$$

provided that

$$\alpha = \frac{1}{5}\gamma^2, \text{ and } \beta = \frac{5}{2}\gamma, \tag{5.7}$$

which holds for the KK case.

Rank 11: One obtains the symmetry

$$\begin{aligned} G &= u^4u_x + \frac{20}{\gamma}uu_x^3 + \frac{40}{\gamma}u^2u_xu_{2x} + \frac{620}{3\gamma^2}u_xu_{2x}^2 + \frac{20}{3\gamma}u^3u_{3x} + \frac{460}{3\gamma^2}u_x^2u_{3x} \\ &+ \frac{200}{\gamma^2}uu_{2x}u_{3x} + \frac{120}{\gamma^2}uu_xu_{4x} + \frac{400}{\gamma^3}u_{3x}u_{4x} + \frac{20}{\gamma^2}u^2u_{5x} + \frac{800}{3\gamma^3}u_{2x}u_{5x} \\ &+ \frac{800}{7\gamma^3}u_xu_{6x} + \frac{200}{7\gamma^3}uu_{7x} + \frac{1000}{63\gamma^4}u_{9x} \end{aligned} \tag{5.8}$$

provided that the condition (5.2) for the Lax hierarchy is satisfied.

In summary, our algorithm allows one to filter out all the integrable cases in the class (5.1). Alternatively, in [19] we investigated the conditions on the parameters α, β, γ such that (5.1) admits an infinite sequence (perhaps with gaps) of polynomial conservation laws. The conditions in [19] are exactly the same as the ones above.

5.2 Hirota-Satsuma System

Hirota and Satsuma [31] proposed a coupled system of KdV equations,

$$u_t = 6\alpha uu_x + 6vv_x + \alpha u_{3x}, \quad v_t = 3uv_x + v_{3x}, \tag{5.9}$$

where α is a nonzero parameter. System (5.9) describes the interaction of two long waves with different dispersion relations. It is known to be completely integrable provided $\alpha = -\frac{1}{2}$. The scaling properties of (5.9) are such that $w(u) = w(v) = 2$. System (5.9) is of rank 5. In a search for the symmetry of higher-order, we obtained the symmetry of rank 7:

$$
\begin{aligned}
G_1 &= u^2 u_x - \frac{2}{3} u_x v^2 - \frac{4}{3} u v v_x + \frac{2}{3} u_x u_{2x} - \frac{2}{3} v_x v_{2x} + \frac{1}{3} u u_{3x} - \frac{2}{3} v v_{3x} + \frac{1}{30} u_{5x}, \\
G_2 &= -\frac{1}{3} u^2 v_x - \frac{2}{3} v^2 v_x - \frac{1}{3} u_{2x} v_x - \frac{2}{3} u_x v_{2x} - \frac{2}{3} u v_{3x} - \frac{2}{15} v_{5x},
\end{aligned} \tag{5.10}
$$

provided that $\alpha = -\frac{1}{2}$, which is the condition for complete integrability of (5.9).

Similarly, as presented in [19], the search for conservation laws leads to the same condition on α.

5.3 Nonlinear Schrödinger Equation

The nonlinear Schrödinger (NLS) equation [1],

$$
i q_t - q_{2x} + 2|q|^2 q = 0, \tag{5.11}
$$

arises as an asymptotic limit of a slowly varying dispersive wave envelope in a nonlinear medium, and as such has significant applications in nonlinear optics and plasma physics. Together with the ubiquitous KdV equation, the completely integrable NLS equation is one of the most studied soliton equations.

In order to compute the symmetries of (5.11), we consider q and q^* as independent variables and add the complex conjugate equation to (5.11). Absorbing i in the scale of t, we get

$$
q_t - q_{2x} + 2q^2 q^* = 0, \quad q^*_t + q^*_{2x} - 2q^{*2} q = 0. \tag{5.12}
$$

Since $w(q) = w(q^*)$, we obtain

$$w(q) = w(q^*) = 1, \quad \text{and} \quad w(\mathrm{D}_t) = -w(t) = 2.$$

Hence, the symmetries of ranks $(4,4), (5,5)$, and $(6,6)$, as computed with our program **InvariantsSymmetries.m** [21], are:

$$\begin{aligned}
G_1^{(1)} &= -6qq_xq^* + q_{3x}, \qquad G_2^{(1)} = G_1^{*(1)} = -6qq^*q_x^* + q_{3x}^*, \\
G_1^{(2)} &= -6q^3q^{*2} + 6q_x^2q^* + 4qq_xq_x^* + 8qq_{2x}q^* + 2q^2q_{2x}^* - q_{4x}, \quad G_2^{(2)} = G_1^{*(2)} \\
G_1^{(3)} &= 30q^2q_xq^{*2} - 10{q_x}^2q_x^* - 20q_xq_{2x}q^* - 10qq_{2x}q_x^* - 10qq_xq_{2x}^* \\
&\quad -10qq_{3x}q^* + q_{5x}, \qquad G_2^{(3)} = G_1^{*(3)}.
\end{aligned} \tag{5.13}$$

For rank 2 through rank 6, we computed the following invariants:

$$\begin{aligned}
\rho_1 &= qq^*, & \rho_2 &= q^*q_x, \\
\rho_3 &= q^2q^{*2} + q_xq_x^*, & \rho_4 &= qq^{*2}q_x + \tfrac{1}{3}q_{2x}q_x^*, \\
\rho_5 &= q^3q^{*3} + \frac{1}{2}q^{*2}q_x^2 + 4qq^*q_xq_x^* + \frac{1}{2}q^2q_x^{*2} + \frac{1}{2}q_{2x}q_{2x}^*. & &
\end{aligned} \tag{5.14}$$

The NLS equation has infinitely many invariants and symmetries.

5.4 Vector Modified KdV Equation

In [68], Verheest investigates the integrability of a vector form of the modified KdV equation (vmKdV),

$$\mathbf{B}_t + (|\mathbf{B}|^2\mathbf{B})_x + \mathbf{B}_{3x} = 0, \tag{5.15}$$

or component-wise with $\mathbf{B} = (u, v)$,

$$\begin{aligned} u_t + 3u^2 u_x + v^2 u_x + 2uvv_x + u_{3x} &= 0, \\ v_t + 3v^2 v_x + u^2 v_x + 2uvu_x + v_{3x} &= 0. \end{aligned} \tag{5.16}$$

With our software we computed

$$\begin{aligned} \rho_1 &= u, \quad \rho_2 = v, \quad \rho_3 = u^2 + v^2, \\ \rho_4 &= \frac{1}{2}(u^2 + v^2)^2 - (u_x^2 + v_x^2), \\ \rho_5 &= \frac{1}{3}x(u^2 + v^2) - \frac{1}{2}t(u^2 + v^2)^2 + t(u_x^2 + v_x^2). \end{aligned} \tag{5.17}$$

Note that the latter invariant depends explicitly on x and t. Verheest [68] has shown that (5.16) is non-integrable for it lacks a bi-Hamiltonian structure and recursion operator. We were unable to find additional polynomial invariants. Polynomial higher-order symmetries for (5.16) do not appear to exist. Our results confirm Verheest's conclusion.

5.5 Heisenberg Spin Model

The continuous Heisenberg spin system [13] or Landau-Lifshitz equation,

$$\mathbf{S}_t = \mathbf{S} \times \Delta\mathbf{S} + \mathbf{S} \times D\mathbf{S}, \tag{5.18}$$

models a continuous anisotropic Heisenberg ferromagnet. It is considered a universal integrable system since various known integrable PDEs, such as the NLS and sine-Gordon equations, can be derived from it. In (5.18), $\mathbf{S} = [u, v, w]^T$ with real components, $\Delta = \nabla^2$ is the Laplacian, D is a diagonal matrix, and cross ($\times$) is the standard cross product of vectors.

Split into components, (5.18) reads

$$\begin{aligned} u_t &= v w_{2x} - w v_{2x} + (\beta - \alpha) v w, \\ v_t &= w u_{2x} - u w_{2x} + (1 - \beta) u w, \\ w_t &= u v_{2x} - v u_{2x} + (\alpha - 1) u v. \end{aligned} \tag{5.19}$$

In the following table, we list the invariants for three typical cases; other cases are similar.

$D = diag(1, \alpha, \beta)$ $\alpha \neq 0, \beta \neq 0$	$D = diag(1, \alpha, 0)$ $\alpha \neq 0$	$D = diag(0, 0, 0)$
$\rho = u$ if $\alpha = \beta$ $\rho = v$ if $\beta = 1$ $\rho = w$ if $\alpha = 1$ $\rho = u^2 + v^2 + w^2$ $\rho = (1-\alpha)v^2 + (1-\beta)w^2$ $+u_x^2 + v_x^2 + w_x^2$	$\rho = w$ if $\alpha = 1$ $\rho = u^2 + v^2 + w^2$ $\rho = (1-\alpha)v^2 + w^2$ $+u_x^2 + v_x^2 + w_x^2$	$\rho = u$ $\rho = v$ $\rho = w$ $\rho = u^2 + v^2 + w^2$ $\rho = u_x^2 + v_x^2 + w_x^2$

Invariants for the Heisenberg Spin Model

Note that for all the cases we considered,

$$\rho = u^2 + v^2 + w^2 = ||\mathbf{S}||^2 \tag{5.20}$$

is constant in time (since $J = 0$). Hence, all even powers of $||\mathbf{S}||$ are also invariants, but they are dependent on (5.20). Furthermore, the sum of two invariants is an invariant. Hence, after adding (5.20) to

$$\rho = (\alpha - 1)v^2 + (\beta - 1)w^2 - (u_x^2 + v_x^2 + w_x^2), \tag{5.21}$$

the latter can be replaced by

$$\rho = u^2 + \alpha v^2 + \beta w^2 - (u_x^2 + v_x^2 + w_x^2). \tag{5.22}$$

Note that $u_x^2 = D_x(uu_x) - uu_{2x}$ and recall that invariants are equivalent if they only differ by a total $x-$derivative. So, (5.22) is equivalent with

$$\rho = u^2 + \alpha v^2 + \beta w^2 + uu_{2x} + vv_{2x} + w_{2x}, \tag{5.23}$$

which can be compactly written as $\rho = \mathbf{S} \cdot \Delta \mathbf{S} + \mathbf{S} \cdot D\mathbf{S}$, where $D = diag(1, \alpha, \beta)$. Consequently, the Hamiltonian of (5.18)

$$\mathcal{H} = -\frac{1}{2} \int \mathbf{S} \cdot \Delta \mathbf{S} + \mathbf{S} \cdot D\mathbf{S} \;\; dx \tag{5.24}$$

is constant in time. The dot $(\cdot)$ refers to the standard inner product of vectors.

Chapter 6

RECURSION OPERATORS OF EVOLUTION EQUATIONS

In this chapter, we describe a new algorithm for the computation of recursion operators of evolution equations. Later, we list our results for several examples. The algorithm is based on scaling properties, and also uses knowledge of symmetries and invariants of the evolution equations.

Recall that for a system of evolutionary PDEs,

$$\mathbf{u}_t = \mathbf{F}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{2x}, ..., \mathbf{u}_{mx}), \tag{6.1}$$

a *recursion operator* is a linear operator $\mathbf{\Phi}$ in the space of differential functions with the property that whenever $\mathbf{G}$ is a symmetry of (6.1), so is $\hat{\mathbf{G}}$ with $\hat{\mathbf{G}} = \mathbf{\Phi}\mathbf{G}$ [52]. For n-component systems, $\mathbf{\Phi}$ is an $n \times n$ matrix.

6.1 Algorithm

For motivation, we start with two examples. For each, we list the recursion operators and make several observations. Later we return to these examples to show how the algorithm works.

Example 6.1 The first four symmetries of the KdV equation (2.7) are

$$G^{(1)} = u_x, \quad G^{(2)} = 6uu_x + u_{3x}, \quad G^{(3)} = 30u^2u_x + 20u_xu_{2x} + 10uu_{3x} + u_{5x},$$

$$G^{(4)} = 140u^3u_x + 70u_x^3 + 280uu_xu_{2x} + 70u^2u_{3x} + 70u_{2x}u_{3x} + 42u_xu_{4x} + 14uu_{5x} + u_{7x},$$

and the recursion operator which links these symmetries is

$$\Phi = \mathrm{D}_x^2 + 2u + 2\mathrm{D}_x u \mathrm{D}_x^{-1} = \mathrm{D}_x^2 + 4u + 2u_x \mathrm{D}_x^{-1}. \tag{6.2}$$

Recall that $w(u) = 2$ for (2.7), and $w(\mathrm{D}_x) = -w(\mathrm{D}_x^{-1}) = 1$ by definition. Therefore, Φ in (6.2) has rank 2, which is also the increment in rank, between the consecutive symmetries of (2.7), i.e.,

$$\mathrm{rank}(G^{(4)}) - \mathrm{rank}(G^{(3)}) = \mathrm{rank}(G^{(3)}) - \mathrm{rank}(G^{(2)}) = \mathrm{rank}(G^{(2)}) - \mathrm{rank}(G^{(1)}) = 2.$$

Since we are dealing with polynomial symmetries, applied to the right hand side of (6.1), for any piece of the recursion operator that involves D_x^{-1}, the integration has to be carried out completely. This is clearly true for our example (6.2), since D_x^{-1} applied to the right hand side of (2.7), i.e. $\mathrm{D}_x^{-1}(6uu_x + u_{3x}) = 3u^2 + u_{2x}$, leaves no integrations.

In order to obtain the terms involving D_x^{-1} in the recursion operator, we use the knowledge of the invariants of the PDE system. Recall that for (2.7), we have $\rho = u, \rho = u^2, \rho = u^3 - \frac{1}{2}u_x^2, \ldots$ Hence, by the conservation law (2.2), we have

$$\begin{aligned} &\mathrm{D}_t u = u_t = -\mathrm{D}_x J, \quad \mathrm{D}_t u^2 = 2uu_t = -\mathrm{D}_x J, \text{ and} \\ &\mathrm{D}_t(u^3 - \frac{1}{2}u_x^2) = 3u^2 u_t - u_x u_{xt} = (3u^2 - u_x \mathrm{D}_x)u_t = -\mathrm{D}_x J, \end{aligned} \tag{6.3}$$

for some polynomial J's. Therefore, $\mathrm{D}_x^{-1}, \mathrm{D}_x^{-1}u$, and $\mathrm{D}_x^{-1}(3u^2 - u_x\mathrm{D}_x)$ applied to the right hand sides of (2.7) will lead to a polynomial result after integration. However, as we will see in the description of the algorithm, the terms involving $\mathrm{D}_x^{-1}u$ and $\mathrm{D}_x^{-1}(3u^2 - u_x\mathrm{D}_x)$ will be of no use for this example.

Example 6.2 Consider the fifth-order Sawada-Kotera equation [8, 61]:

$$u_t = 5u^2u_x + 5u_xu_{2x} + 5uu_{3x} + u_{5x}, \tag{6.4}$$

with $w(u) = 2$. Equation (6.4) belongs to a family that has infinitely many polynomial symmetries. Moreover, these symmetries use one of the two distinct "seeds", namely:

$$G^{(1)} = u_x, \quad G^{(2)} = 5u^2u_x + 5u_xu_{2x} + 5uu_{3x} + u_{5x}, \tag{6.5}$$

i.e., $G^{(2i-1)}$ with rank $6i - 3$ uses $G^{(1)}$ as the seed, and $G^{(2i)}$ with rank $6i + 1$ uses $G^{(2)}$ as the seed, where i is a positive integer. Hence, the rank of the recursion operator should be 6, agreeing with the operator in [58]

$$\begin{aligned}\Phi &= \mathrm{D}_x^6 + 2u\mathrm{D}_x^4 + 2\mathrm{D}_xu\mathrm{D}_x^3 + \mathrm{D}_x^2u\mathrm{D}_x^2 + 3u\mathrm{D}_xu\mathrm{D}_x + 3u\mathrm{D}_x^2u - 2\mathrm{D}_xu\mathrm{D}_xu - 2u^3 \\ &\quad + \mathrm{D}_x^5u\mathrm{D}_x^{-1} + 5\mathrm{D}_xu\mathrm{D}_x^2u\mathrm{D}_x^{-1} + 5u^2\mathrm{D}_xu\mathrm{D}_x^{-1} + \mathrm{D}_xu\mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x). \end{aligned} \tag{6.6}$$

Again, let us look at the terms involving D_x^{-1} in the recursion operator. Applied to the right hand side of (6.4), they all lead to a polynomial, i.e.,

$$\begin{aligned}&\mathrm{D}_x^{-1}(5u^2u_x + 5u_xu_{2x} + 5uu_{3x} + u_{5x}) = \frac{5}{3}u^3 + 5uu_{2x} + u_{4x}, \qquad (6.7)\\ &\mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x)(5u^2u_x + 5u_xu_{2x} + 5uu_{3x} + u_{5x}) = u^5 - 10u^2u_x^2 + \ldots - 2u_xu_{5x}.\end{aligned}$$

Once again, we can use the invariants of (6.4) to find the pieces involving D_x^{-1}. The first two invariants of (6.4) [19] are $\rho = u, \rho = \frac{1}{3}u^3 - u_x^2$. Therefore, we have

$$\mathrm{D}_tu = u_t = -\mathrm{D}_xJ, \quad \text{and}$$

$$\mathrm{D}_t(\frac{1}{3}u^3 - u_x^2) = u^2 u_t - 2u_x u_{xt} = (u^2 - 2u_x\mathrm{D}_x)u_t = -\mathrm{D}_x J. \qquad (6.8)$$

Now that we showed where the terms involving D_x^{-1} and $\mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x)$ in (6.6) come from, we can start the description of our 2–step algorithm.

6.1.1 Step 1: Construct the form of the recursion operator

By definition, the recursion operator is a linear operator in the space of differential functions. Based on the structure of the recursion operators in the previous examples, it makes sense to use $\mathbf{u}, \mathrm{D}_x$, and pieces involving D_x^{-1}, as the building blocks.

Example 6.3 We return to the KdV equation (2.7). Since the rank of the symmetries in the Lax hierarchy increases by two, the rank of the recursion operator must be two. From (6.3), we know that we can use

$$\mathcal{K} = \{\mathrm{D}_x^{-1}, \mathrm{D}_x^{-1}u, \mathrm{D}_x^{-1}(3u^2 - u_x\mathrm{D}_x)\}$$

as building blocks in addition to u and D_x.

Start by listing all the permutations of $\mathrm{D}_x^j u^k$ of exactly rank 2, for j, k nonnegative integers: $\mathcal{L} = \{\mathrm{D}_x^2, u\}$.

Next, for each element in $\mathcal{K}$, we introduce possible monomials of u and D_x on the left, so that we get a new expression that has exactly rank 2. For instance, for $\mathrm{D}_x^{-1}u$, we can introduce D_x on the left, but then it becomes u, which we have in $\mathcal{L}$. For D_x^{-1}, we can introduce $\mathrm{D}_x u$ and $u\mathrm{D}_x$ on the left, taking the different permutations. But, the latter one also results in u. There is no way of introducing powers of u and D_x on the left to $\mathrm{D}_x^{-1}(3u^2 - u_x\mathrm{D}_x)$, since it is already of rank 3, and we do not allow repetitions of the building blocks involving D_x^{-1} in the same term. Hence, we end up with $\mathcal{M} = \{\mathrm{D}_x u \mathrm{D}_x^{-1}\}$. The union of $\mathcal{L}$ and $\mathcal{M}$ is

$$\mathcal{R} = \{\mathrm{D}_x^2, u, \mathrm{D}_x u \mathrm{D}_x^{-1}\},$$

which has the building blocks of the recursion operator. Linear combination of the monomials in $\mathcal{R}$ with constant coefficients c_i gives the form of the recursion operator:

$$\Phi = c_1\,\mathrm{D}_x^2 + c_2\,u + c_3\,\mathrm{D}_x u \mathrm{D}_x^{-1}. \tag{6.9}$$

Example 6.4 Let us return to (6.4). We know that the recursion operator must be of rank 6. Start with all possible permutations of $\mathrm{D}_x^j u^k$ of exactly rank 6:

$$\begin{aligned}\mathcal{L} \;=\; & \{\mathrm{D}_x^6, u\mathrm{D}_x^4, \mathrm{D}_x u \mathrm{D}_x^3, \mathrm{D}_x^2 u \mathrm{D}_x^2, \mathrm{D}_x^3 u \mathrm{D}_x, \mathrm{D}_x^4 u, u^2\mathrm{D}_x^2, u\mathrm{D}_x u \mathrm{D}_x, u\mathrm{D}_x^2 u, \mathrm{D}_x u^2 \mathrm{D}_x,\\ & \mathrm{D}_x u \mathrm{D}_x u, \mathrm{D}_x^2 u^2, u^3\}.\end{aligned}$$

From (6.8), we have

$$\mathcal{K} = \{\mathrm{D}_x^{-1}, \mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x)\},$$

and

$$\begin{aligned}\mathcal{M} \;=\; & \{\mathrm{D}_x^5 u \mathrm{D}_x^{-1}, u\mathrm{D}_x^3 u \mathrm{D}_x^{-1}, \mathrm{D}_x u \mathrm{D}_x^2 u \mathrm{D}_x^{-1}, \mathrm{D}_x^2 u \mathrm{D}_x u \mathrm{D}_x^{-1}, \mathrm{D}_x^3 u^2 \mathrm{D}_x^{-1}, u^2\mathrm{D}_x u \mathrm{D}_x^{-1},\\ & u\mathrm{D}_x u^2 \mathrm{D}_x^{-1}, \mathrm{D}_x u^3 \mathrm{D}_x^{-1}, \mathrm{D}_x u \mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x)\}.\end{aligned}$$

Then based on all the elements in $\mathcal{R} = \mathcal{L} \bigcup \mathcal{M}$, we obtain

$$\begin{aligned}\Phi \;=\; & c_1\mathrm{D}_x^6 + c_2 u\mathrm{D}_x^4 + c_3\mathrm{D}_x u \mathrm{D}_x^3 + c_4\mathrm{D}_x^2 u \mathrm{D}_x^2 + c_5\mathrm{D}_x^3 u \mathrm{D}_x + c_6\mathrm{D}_x^4 u + c_7 u^2\mathrm{D}_x^2\\ & + c_8 u\mathrm{D}_x u \mathrm{D}_x + c_9 u\mathrm{D}_x^2 u + c_{10}\mathrm{D}_x u^2 \mathrm{D}_x + c_{11}\mathrm{D}_x u \mathrm{D}_x u + c_{12}\mathrm{D}_x^2 u^2 + c_{13}u^3\end{aligned}$$

$$+c_{14}\mathrm{D}_x^5 u\mathrm{D}_x^{-1} + c_{15}u\mathrm{D}_x^3 u\mathrm{D}_x^{-1} + c_{16}\mathrm{D}_x u\mathrm{D}_x^2 u\mathrm{D}_x^{-1} + c_{17}\mathrm{D}_x^2 u\mathrm{D}_x u\mathrm{D}_x^{-1}$$
$$+c_{18}\mathrm{D}_x^3 u^2\mathrm{D}_x^{-1} + c_{19}u^2\mathrm{D}_x u\mathrm{D}_x^{-1} + c_{20}u\mathrm{D}_x u^2\mathrm{D}_x^{-1} + c_{21}\mathrm{D}_x u^3\mathrm{D}_x^{-1}$$
$$+c_{22}\mathrm{D}_x u\mathrm{D}_x^{-1}(u^2 - 2u_x\mathrm{D}_x). \tag{6.10}$$

Now we can start with Step 2 of the algorithm.

6.1.2 Step 2: Determine the unknown coefficients in the operator

We determine the coefficients c_i by requiring that

$$\Phi G^{(i)} = G^{(i+s)}, \quad i = 1, 2, 3, ...,$$

where s is the number of seeds. Usually, the relations for the first few values of i already fix all the unknowns. Therefore, we go up in i as needed.

Example 6.5 For the KdV equation (2.7), requiring $\Phi G^{(2)} = G^{(3)}$, with Φ in (6.9), results in the linear system

$$\mathcal{S} = \{c_1 - 1 = 0, 18c_1 + c_3 - 20 = 0, 6c_1 + c_2 + c_3 - 10 = 0, 6c_2 + 9c_3 - 30 = 0\}.$$

The solution is $c_1 = 1, c_2 = c_3 = 2$. Substituting the solution into (6.9), we obtain (6.2).

Example 6.6 For (6.4), requiring that

$$G^{(3)} = \Phi G^{(1)}, \quad \text{and} \quad G^{(4)} = \Phi G^{(2)},$$

with Φ in (6.10) results in (6.6). Notice that after partial integration (6.6) can be also written as [58]

$$\begin{aligned}\Phi &= \mathrm{D}_x^6 + 6u\mathrm{D}_x^4 + 9u_x\mathrm{D}_x^3 + 9u^2\mathrm{D}_x^2 + 11u_{2x}\mathrm{D}_x^2 + 10u_{3x}\mathrm{D}_x + 21uu_x\mathrm{D}_x + 4u^3 \\ &\quad +16uu_{2x} + 6u_x^2 + 5u_{4x} + u_x\mathrm{D}_x^{-1}(2u_{2x} + u^2) + G^{(2)}\mathrm{D}_x^{-1}, \qquad (6.11)\end{aligned}$$

with $G^{(2)}$ in (6.5), and observe that

$$\mathrm{D}_x^{-1}(2u_{2x} + u^2)(5u^2u_x + 5u_xu_{2x} + 5uu_{3x} + u_{5x}) = u^5 + 5u^3u_{2x} + \ldots + 2u_{2x}u_{4x}. \quad (6.12)$$

Using an example, we now show how we can generalize the algorithm to systems of PDEs.

Example 6.7 Consider the Hirota-Satsuma system (5.9). The right hand side of (5.9) belong to an infinite family of symmetries if and only if $\alpha = -\frac{1}{2}$. The symmetries use one of the two seeds, namely:

$$\mathbf{G}^{(1)} = (G_1^{(1)}, G_2^{(1)}) \text{ with } G_1^{(1)} = u_x, \quad G_2^{(1)} = v_x, \qquad (6.13)$$

$$\mathbf{G}^{(2)} = (G_1^{(2)}, G_2^{(2)}) \text{ with } G_1^{(2)} = -3uu_x + 6vv_x - \frac{1}{2}u_{3x}, \quad G_2^{(2)} = 3uv_x + v_{3x}. \quad (6.14)$$

Furthermore, $\mathbf{G}^{(2i-1)}$ with rank $4i-1$ uses $\mathbf{G}^{(1)}$ as the seed, $\mathbf{G}^{(2i)}$ with rank $4i+1$ uses $\mathbf{G}^{(2)}$ as the seed, where i is a positive integer. Hence, the rank of the recursion operator $\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} \end{pmatrix}$ is 4, i.e., the ranks of $\boldsymbol{\Phi}_{ij}$ are all 4.

Taking $\alpha = -\frac{1}{2}$, the first two invariants of (5.9) are $\rho = u, \rho = \frac{1}{2}u^2 - v^2$. Hence, we have

$$\mathrm{D}_t u = u_t = (1,0)\cdot(u_t, v_t) = -\mathrm{D}_x J, \text{ and}$$

$$D_t(\frac{1}{2}u^2 - v^2) = uu_t - 2vv_t = (u, -2v)\cdot(u_t, v_t) = -D_x J, \tag{6.15}$$

where dot $(\cdot)$ refers to the standard inner product of vectors. Therefore, we can use

$$\mathcal{K} = \left\{D_x^{-1}\mathbf{\Lambda}, D_x^{-1}\mathbf{\Psi}\right\},$$

where $\mathbf{\Lambda} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ and $\mathbf{\Psi} = \begin{pmatrix} u & -2v \\ u & -2v \end{pmatrix}$, as building blocks, in addition to $\mathbf{u}$ and D_x^{-1}. With

$$\mathcal{L} = \left\{u^2\mathbf{\Gamma}, v^2\mathbf{\Gamma}, uv\mathbf{\Gamma}, D_x^2 u\mathbf{\Gamma}, D_x u D_x\mathbf{\Gamma}, uD_x^2\mathbf{\Gamma}, D_x^2 v\mathbf{\Gamma}, D_x v D_x\mathbf{\Gamma}, vD_x^2\mathbf{\Gamma}, D_x^4\mathbf{\Gamma}\right\},$$

where $\mathbf{\Gamma} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and

$$\begin{aligned} \mathcal{M} \;=\; & \{D_x^3 u D_x^{-1}\mathbf{\Lambda}, D_x^3 v D_x^{-1}\mathbf{\Lambda}, D_x u^2 D_x^{-1}\mathbf{\Lambda}, uD_x u D_x^{-1}\mathbf{\Lambda}, D_x v^2 D_x^{-1}\mathbf{\Lambda}, vD_x v D_x^{-1}\mathbf{\Lambda}, \\ & D_x uv D_x^{-1}\mathbf{\Lambda}, uD_x v D_x^{-1}\mathbf{\Lambda}, vD_x u D_x^{-1}\mathbf{\Lambda}, D_x u D_x^{-1}\mathbf{\Psi}, D_x v D_x^{-1}\mathbf{\Psi}\}, \end{aligned} \tag{6.16}$$

we can obtain the form of the recursion operator $\mathbf{\Phi}$. The linear combinations involve the entries of $\mathbf{\Lambda}, \mathbf{\Psi}$ and $\mathbf{\Gamma}$ multiplied with different sets of constants c_{ij} for each appearance. Requiring that $\mathbf{G}^{(3)} = \mathbf{\Phi}\mathbf{G}^{(1)}$ and $\mathbf{G}^{(4)} = \mathbf{\Phi}\mathbf{G}^{(2)}$ gives

$$\begin{aligned} \mathbf{\Phi}_{11} &= \frac{1}{2}(-3u^2 - D_x^2 u - D_x u D_x - uD_x^2 - \frac{1}{2}D_x^4 - 2D_x u D_x^{-1} u - D_x^3 u D_x^{-1} \\ &\quad -3D_x u^2 D_x^{-1} + 4D_x v^2 D_x^{-1} + 4vD_x v D_x^{-1}), \\ \mathbf{\Phi}_{12} &= 2uv + D_x^2 v + 2D_x v D_x + 2vD_x^2 + 2D_x u D_x^{-1} v, \\ \mathbf{\Phi}_{21} &= \frac{1}{2}(-D_x v D_x - vD_x^2 - 2D_x v D_x^{-1} u + 2D_x^3 v D_x^{-1} + 4D_x uv D_x^{-1} \end{aligned}$$

$$+2u\mathrm{D}_x v\mathrm{D}_x^{-1} - 4v\mathrm{D}_x u\mathrm{D}_x^{-1}),$$

$$\Phi_{22} = 2v^2 + 2\mathrm{D}_x u\mathrm{D}_x + 2u\mathrm{D}_x^2 + \mathrm{D}_x^4 + 2\mathrm{D}_x v\mathrm{D}_x^{-1}v. \tag{6.17}$$

6.2 Examples

In the examples below, we list the recursion operators obtained using our algorithm.

Example 6.8 Burgers Equation

The Burgers equation is given by

$$u_t = uu_x + u_{2x}, \tag{6.18}$$

and based on our algorithm the recursion operator for (6.18) is

$$\Phi = \mathrm{D}_x + \frac{1}{2}\mathrm{D}_x u\mathrm{D}_x^{-1},$$

which agrees with the result in [58].

Example 6.9 Modified KdV (MKdV) Equation

For the MKdV equation [1],

$$u_t = 6u^2 u_x + u_{3x}, \tag{6.19}$$

the recursion operator is

$$\Phi = \mathrm{D}_x^2 + 4\mathrm{D}_x u\mathrm{D}_x^{-1}u.$$

Example 6.10 Potential KdV (PKdV) Equation

The PKdV equation is

$$u_t = u_x^2 + u_{3x}, \tag{6.20}$$

and its recursion operator [58] is

$$\Phi = \mathrm{D}_x^2 - \frac{2}{3}u\mathrm{D}_x + \frac{2}{3}\mathrm{D}_x u + \frac{2}{3}\mathrm{D}_x^{-1}u_x\mathrm{D}_x.$$

Example 6.11 Kaup-Kupershmidt Equation

The Kaup-Kupershmidt [39] equation,

$$u_t = 20u^2u_x + 25u_xu_{2x} + 10uu_{3x} + u_{5x}, \tag{6.21}$$

has the recursion operator [58]:

$$\begin{aligned}\Phi \;=\; & \mathrm{D}_x^6 + u\mathrm{D}_x^4 + \mathrm{D}_x u\mathrm{D}_x^3 + 2\mathrm{D}_x^2 u\mathrm{D}_x^2 + 3\mathrm{D}_x^3 u\mathrm{D}_x + 3\mathrm{D}_x^4 u - 3u^2\mathrm{D}_x^2 - 3u\mathrm{D}_x u\mathrm{D}_x \\ & +51u\mathrm{D}_x^2 u - 29\mathrm{D}_x u\mathrm{D}_x u + 2\mathrm{D}_x^5 u\mathrm{D}_x^{-1} - 30u\mathrm{D}_x^3 u\mathrm{D}_x^{-1} + 50\mathrm{D}_x u\mathrm{D}_x^2 u\mathrm{D}_x^{-1} \\ & +8u^2\mathrm{D}_x u\mathrm{D}_x^{-1} + 16u\mathrm{D}_x u^2\mathrm{D}_x^{-1} + 2\mathrm{D}_x u\mathrm{D}_x^{-1}(4u^2 - u_x\mathrm{D}_x).\end{aligned}$$

Example 6.12 Potential Sawada-Kotera Equation

The potential Sawada-Kotera equation,

$$u_t = \frac{5}{3}u_x^3 + 5u_xu_{3x} + u_{5x}, \tag{6.22}$$

has the recursion operator [58]:

$$\begin{aligned}\Phi \;=\; & \mathrm{D}_x^6 - 6u\mathrm{D}_x^5 + 7\mathrm{D}_x u\mathrm{D}_x^4 - \mathrm{D}_x^2 u\mathrm{D}_x^3 - \mathrm{D}_x^4 u\mathrm{D}_x + \mathrm{D}_x^5 u + 5u^2\mathrm{D}_x^4 \\ & -7u\mathrm{D}_x u\mathrm{D}_x^3 + 4u\mathrm{D}_x^3 u\mathrm{D}_x - 5u\mathrm{D}_x^4 u + 2\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x^2 - 4\mathrm{D}_x u\mathrm{D}_x^2 u\mathrm{D}_x \\ & +5\mathrm{D}_x u\mathrm{D}_x^3 u - \frac{8}{3}u^3\mathrm{D}_x^3 + \frac{13}{3}u^2\mathrm{D}_x u\mathrm{D}_x^2 - \frac{7}{3}u^2\mathrm{D}_x^2 u\mathrm{D}_x + 5u^2\mathrm{D}_x^3 u \\ & +\frac{2}{3}u\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x - \frac{20}{3}u\mathrm{D}_x u\mathrm{D}_x^2 u + \frac{5}{3}\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x u\end{aligned}$$

$$+\frac{1}{3}\mathrm{D}_x^{-1}(4u_x^3\mathrm{D}_x - 9u_{2x}^2\mathrm{D}_x - 18u_xu_{2x}\mathrm{D}_x^2 + 6u_{3x}\mathrm{D}_x^3)$$
$$+\mathrm{D}_x u\mathrm{D}_x^{-1}(u_x^2\mathrm{D}_x - 2u_{2x}\mathrm{D}_x^2).$$

Example 6.13 Potential Kaup-Kupershmidt Equation

The potential Kaup-Kupershmidt equation,

$$u_t = \frac{20}{3}u_x^3 + \frac{15}{2}u_{2x}^2 + 10u_xu_{3x} + u_{5x}, \tag{6.23}$$

has the recursion operator [58]:

$$\begin{aligned}\Phi \;=\; & \mathrm{D}_x^6 - 3u\mathrm{D}_x^5 + 2\mathrm{D}_x u\mathrm{D}_x^4 - \mathrm{D}_x^2 u\mathrm{D}_x^3 + \mathrm{D}_x^4 u\mathrm{D}_x + 2\mathrm{D}_x^5 u + 11u^2\mathrm{D}_x^4\\ & +2u\mathrm{D}_x u\mathrm{D}_x^3 - 45u\mathrm{D}_x^2 u\mathrm{D}_x^2 + 28u\mathrm{D}_x^3 u\mathrm{D}_x - 5u\mathrm{D}_x^4 u + 32\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x^2\\ & -28\mathrm{D}_x u\mathrm{D}_x^2 u\mathrm{D}_x - 10\mathrm{D}_x u\mathrm{D}_x^3 u + 15\mathrm{D}_x^2 u\mathrm{D}_x^2 u - \frac{64}{3}u^3\mathrm{D}_x^3 + \frac{104}{3}u^2\mathrm{D}_x u\mathrm{D}_x^2\\ & -\frac{56}{3}u^2\mathrm{D}_x^2 u\mathrm{D}_x + 40u^2\mathrm{D}_x^3 u + \frac{16}{3}u\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x - \frac{160}{3}u\mathrm{D}_x u\mathrm{D}_x^2 u\\ & +\frac{40}{3}\mathrm{D}_x u\mathrm{D}_x u\mathrm{D}_x u + \frac{1}{3}\mathrm{D}_x^{-1}(32u_x^3\mathrm{D}_x - 18u_{2x}^2\mathrm{D}_x - 36u_xu_{2x}\mathrm{D}_x^2 + 3u_{3x}\mathrm{D}_x^3).\end{aligned}$$

Example 6.14 Boussinesq Equation

For the Boussinesq equation [52, p. 460]

$$u_t = v_x, \quad v_t = \frac{8}{3}uu_x + \frac{1}{3}u_{3x}, \tag{6.24}$$

the recursion operator is

$$\Phi = \begin{pmatrix} 3(v + 2\mathrm{D}_x v\mathrm{D}_x^{-1}) & 3(\mathrm{D}_x^2 + u + \mathrm{D}_x u\mathrm{D}_x^{-1}) \\ \mathrm{D}_x^4 + 3\mathrm{D}_x(\mathrm{D}_x u + u\mathrm{D}_x) & \\ \quad +2u(\mathrm{D}_x^2 + 4u) + 2(4\mathrm{D}_x u + \mathrm{D}_x^3)u\mathrm{D}_x^{-1} & 3(2v + \mathrm{D}_x v\mathrm{D}_x^{-1}) \end{pmatrix}. \tag{6.25}$$

Chapter 7

DDE CASE: DEFINITIONS AND THE KEY CONCEPT

In this chapter, we give the definitions of a symmetry and a conservation law for a system of DDEs. Also, we describe the dilation invariance for DDEs, which is the key concept behind our algorithms. Later in the chapter we define an equivalence relationship, which will be used for the computation of conservation laws.

Consider a system of DDEs,

$$\dot{\mathbf{u}}_n = \mathbf{F}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...), \tag{7.1}$$

where the equations are continuous in time, and discretized in the (single) space variable. As before, $\mathbf{u}_n$ and $\mathbf{F}$ are vector dynamical variables with any number of components, and $\mathbf{F}$ is assumed to be a polynomial with constant coefficients. There are no restrictions on the level of the shifts or the degree of nonlinearity. If DDEs are of second or higher order in t, they must be recast in the form (7.1).

For notational simplicity, the components of $\mathbf{u}_n$ will be denoted by u_n, v_n, etc., and we use $F_1, F_2, ...$ to denote the components of $\mathbf{F}$.

7.1 Symmetry

A vector function $\mathbf{G}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...)$ is called a *symmetry* of (7.1) if the infinitesimal transformation

$$\mathbf{u}_n \to \mathbf{u}_n + \epsilon \mathbf{G}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...) \tag{7.2}$$

leaves (7.1) invariant within order ϵ. Consequently, $\mathbf{G}$ must satisfy the linearized equation [9, 11]

$$\mathrm{D}_t\mathbf{G} = \mathbf{F}'(\mathbf{u}_n)[\mathbf{G}], \tag{7.3}$$

where $\mathbf{F}'$ is the Fréchet derivative of $\mathbf{F}$, defined as

$$\mathbf{F}'(\mathbf{u}_n)[\mathbf{G}] = \frac{\partial}{\partial \epsilon}\mathbf{F}(\mathbf{u}_n + \epsilon\mathbf{G})_{|\epsilon=0}. \tag{7.4}$$

Of course, (7.2) means that $\mathbf{u}_{n+k}$ is replaced by $\mathbf{u}_{n+k} + \epsilon\mathbf{G}_{|n\to n+k}$. For compactness of notation, in (7.3) and (7.4) we used $\mathbf{F}'(\mathbf{u}_n)$ instead of $\mathbf{F}'(\ldots, \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, \ldots)$.

7.2 Conservation Law

For (7.1), we define a local *conservation law* by

$$\dot{\rho}_n = J_n - J_{n+1}, \tag{7.5}$$

where ρ_n is the *invariant* (conserved density) and J_n is the associated *flux*. Both functionals are assumed to be polynomials in $\mathbf{u}_n$ and its shifts. Also, (7.5) is satisfied on solutions of (7.1). Obviously,

$$\frac{\mathrm{d}}{\mathrm{dt}}\sum_n \rho_n = \sum_n \dot{\rho}_n = \sum_n (J_n - J_{n+1}), \tag{7.6}$$

and the telescopic series $\sum_n (J_n - J_{n+1})$ vanishes for a bounded periodic lattice or a bounded lattice resting at infinity. In that case, $\sum_n \rho_n$ is constant in time. So, we have a conserved quantity.

Example 7.1 Consider the one-dimensional lattice [25, 67]

$$\ddot{y}_n = \exp(y_{n-1} - y_n) - \exp(y_n - y_{n+1}), \tag{7.7}$$

due to Toda. In (7.7), y_n is the displacement from equilibrium of the nth particle with unit mass under an exponential decaying interaction force between nearest neighbors. With the change of variables,

$$u_n = \dot{y}_n, \qquad v_n = \exp(y_n - y_{n+1}), \tag{7.8}$$

the Toda lattice (7.7) can be written in polynomial form

$$\dot{u}_n = v_{n-1} - v_n, \qquad \dot{v}_n = v_n(u_n - u_{n+1}). \tag{7.9}$$

System (7.9) is completely integrable. The first two invariant-flux pairs

$$\rho_n^{(1)} = u_n, \quad J_n^{(1)} = v_{n-1}, \quad \text{and} \quad \rho_n^{(2)} = \tfrac{1}{2}u_n^2 + v_n, \quad J_n^{(2)} = u_n v_{n-1}, \tag{7.10}$$

can be easily computed by hand. Moreover, one higher-order symmetry of (7.9) is:

$$G_1 = v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} + u_n), \quad G_2 = v_n(u_{n+1}^2 - u_n^2) + v_n(v_{n+1} - v_{n-1}). \tag{7.11}$$

7.3 Key Concept: Dilation Invariance

Recall that scaling invariance, which results from a special Lie-point symmetry, is an intrinsic property of many integrable nonlinear PDEs and DDEs. Indeed, observe that (7.9), and the couples $\rho_n^{(1)}, J_n^{(1)}$ and $\rho_n^{(2)}, J_n^{(2)}$ in (7.10) (after inserting them in

(7.5)), and (7.11) are all invariant under the dilation symmetry

$$(t, u_n, v_n) \rightarrow (\lambda^{-1} t, \lambda u_n, \lambda^2 v_n), \tag{7.12}$$

where λ is an arbitrary parameter. Stated differently, u_n corresponds to one derivative with respect to t; denoted by $u_n \sim \frac{\mathrm{d}}{\mathrm{dt}}$. Similarly, $v_n \sim \frac{\mathrm{d}^2}{\mathrm{dt}^2}$.

Analogous to the PDE case, we will exploit this scaling invariance to find symmetries and invariants.

For an algorithmic determination of dilation invariance, let us give a couple of definitions. In contrast to the PDE case, we have to define the *weight*, w, of variables in terms of the number of derivatives with respect to t, and we set $w(\frac{\mathrm{d}}{\mathrm{dt}}) = 1$. Weights of dependent variables are nonnegative, rational, and independent of n. In view of (7.12), we have $w(u_n) = 1$, and $w(v_n) = 2$.

The *rank* of a monomial is defined as the total weight of the monomial, again in terms of derivatives with respect to t. Observe that in the first equation of (7.9), all the monomials have the same rank, namely 2, and in the second equation, all the terms have rank 3.

Conversely, requiring uniformity in rank for each equation in (7.9) allows one to compute the weights of the dependent variables. Indeed,

$$w(u_n) + 1 = w(v_n), \quad w(v_n) + 1 = w(u_n) + w(v_n),$$

yields $w(u_n) = 1$, $w(v_n) = 2$, which is consistent with (7.12).

7.4 Equivalence Criterion

We introduce a few concepts that will be used in our algorithm for conservation laws. Let U denote the *shift-up* operator, defined on functions of the dependent variables. Its inverse, $D = U^{-1}$, is the *shift-down* operator. Both are defined on the set of all monomials in $\mathbf{u}_n$ and its shifts. If m is such a monomial then $Dm = m|_{n \to n-1}$ and $Um = m|_{n \to n+1}$. For example, $Du_{n+2}v_n = u_{n+1}v_{n-1}$ and $Uu_{n-2}v_{n-1} = u_{n-1}v_n$. It is easy to verify that compositions of D and U define an *equivalence relation* on monomials. Simply stated, all shifted monomials are *equivalent*, e.g. $u_{n-1}v_{n+1} \equiv u_{n+2}v_{n+4} \equiv u_{n-3}v_{n-1}$. This equivalence relation holds for any function of the dependent variables, but for the construction of conserved invariants we will apply it only to monomials.

In the algorithm, we will use the following *equivalence criterion*: if two monomials m_1 and m_2 are equivalent, $m_1 \equiv m_2$, then $m_1 = m_2 + [M_n - M_{n+1}]$ for some polynomial M_n that depends on $\mathbf{u}_n$ and its shifts. For example, $u_{n-2}u_n \equiv u_{n-1}u_{n+1}$ since $u_{n-2}u_n = u_{n-1}u_{n+1} + [u_{n-2}u_n - u_{n-1}u_{n+1}] = u_{n-1}u_{n+1} + [M_n - M_{n+1}]$, with $M_n = u_{n-2}u_n$.

The *main* representative of an equivalence class is the monomial of that class with n as *lowest* label on u (or v). For example, u_nu_{n+2} is the main representative of the class with elements $u_{n-1}u_{n+1}, u_{n+1}u_{n+3}$, etc. Lexicographical ordering is used to resolve conflicts. For example, u_nv_{n+2} (not $u_{n-2}v_n$) is the main representative in the class with elements $u_{n-3}v_{n-1}, u_{n+2}v_{n+4}$, etc.

Chapter 8

SYMMETRIES OF LATTICES

In this chapter, we describe our algorithm for the computation of symmetries of lattice equations. The algorithm is first illustrated on a scaling invariant example. Later in the chapter, we show how to handle nonuniform systems.

Recall that for a system of DDEs,

$$\dot{\mathbf{u}}_n = \mathbf{F}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...), \tag{8.1}$$

a vector function $\mathbf{G}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...)$ is called a *symmetry* of (8.1) if $\mathbf{G}$ satisfies the linearized equation [9, 11]

$$\mathrm{D}_t\mathrm{G} = \mathbf{F}'(\mathbf{u}_n)[\mathrm{G}], \tag{8.2}$$

where $\mathbf{F}'$ is the Fréchet derivative (7.4) of $\mathbf{F}$.

8.1 Algorithm

As the leading example, we consider the Toda lattice (7.9) with scaling properties such that $w(u_n) = 1$ and $w(v_n) = 2$. Our 2-step algorithm exploits this scaling property to find symmetries.

8.1.1 Step 1: Construct the form of the symmetry

As an example, we compute the form of the symmetry (G_1, G_2) of rank $(3,4)$. Start by listing all monomials in u_n and v_n of ranks 3 and 4, or less:

$$\mathcal{L}_1 = \{u_n^3, u_n^2, u_n v_n, u_n, v_n\}, \quad \mathcal{L}_2 = \{u_n^4, u_n^3, u_n^2 v_n, u_n^2, u_n v_n, u_n, v_n^2, v_n\}.$$

Next, for each monomial in $\mathcal{L}_1$ and $\mathcal{L}_2$, introduce the necessary t-derivatives. so that each term has exactly rank 3 and 4, respectively. At the same time, use (7.9) to remove all $t-$derivatives. Doing so, based on $\mathcal{L}_1$, we obtain

$$\frac{\mathrm{d}^0}{\mathrm{d}t^0}(u_n^3) = u_n^3, \qquad \frac{\mathrm{d}^0}{\mathrm{d}t^0}(u_n v_n) = u_n v_n,$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(u_n^2) = 2u_n \dot{u}_n = 2u_n v_{n-1} - 2u_n v_n, \qquad \frac{\mathrm{d}}{\mathrm{d}t}(v_n) = \dot{v}_n = u_n v_n - u_{n+1} v_n,$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}(u_n) = \frac{\mathrm{d}}{\mathrm{d}t}(\dot{u}_n) = \frac{\mathrm{d}}{\mathrm{d}t}(v_{n-1} - v_n) = u_{n-1} v_{n-1} - u_n v_{n-1} - u_n v_n + u_{n+1} v_n.$$

Gather the resulting terms in a set: $\mathcal{R}_1 = \{u_n^3, u_{n-1}v_{n-1}, u_n v_{n-1}, u_n v_n, u_{n+1} v_n\}$. Similarly, based on the monomials in $\mathcal{L}_2$, we get

$$\begin{aligned}\mathcal{R}_2 \ = \ & \{u_n^4, u_{n-1}^2 v_{n-1}, u_{n-1} u_n v_{n-1}, u_n^2 v_{n-1}, v_{n-2} v_{n-1}, v_{n-1}^2, u_n^2 v_n, \\ & u_n u_{n+1} v_n, u_{n+1}^2 v_n, v_{n-1} v_n, v_n^2, v_n v_{n+1}\}.\end{aligned}$$

Linear combination of the monomials in $\mathcal{R}_1$ and $\mathcal{R}_2$ with constant coefficients c_i gives the explicit form of the symmetry:

$$G_1 \ = \ c_1\, u_n^3 + c_2\, u_{n-1} v_{n-1} + c_3\, u_n v_{n-1} + c_4\, u_n v_n + c_5\, u_{n+1} v_n,$$

$$G_2 = c_6\, u_n^4 + c_7\, u_{n-1}^2 v_{n-1} + c_8\, u_{n-1} u_n v_{n-1} + c_9\, u_n^2 v_{n-1} + c_{10}\, v_{n-2} v_{n-1} + c_{11}\, v_{n-1}^2 + c_{12}\, u_n^2 v_n + c_{13}\, u_n u_{n+1} v_n + c_{14}\, u_{n+1}^2 v_n + c_{15}\, v_{n-1} v_n + c_{16}\, v_n^2 + c_{17}\, v_n v_{n+1}. \tag{8.3}$$

8.1.2 Step 2: Determine the unknown coefficients in the symmetry

To determine the coefficients c_i, we require that (8.2) holds on any solution of (8.1). Compute $\mathrm{D}_t\mathbf{G}$ and use (8.1) to remove all $\dot{\mathbf{u}}_{n-1}, \dot{\mathbf{u}}_n, \dot{\mathbf{u}}_{n+1}$, etc. Compute the Fréchet derivative (7.4) and, in view of (8.2), equate the resulting expressions. Considering as independent all the monomials in $\mathbf{u}_n$ and their shifts, we obtain the linear system for the coefficients c_i.

Applied to (7.9) with (8.3), we obtain the solution

$$\begin{aligned} &c_1 = c_6 = c_7 = c_8 = c_9 = c_{10} = c_{11} = c_{13} = c_{16} = 0, \\ &-c_2 = -c_3 = c_4 = c_5 = -c_{12} = c_{14} = -c_{15} = c_{17}. \end{aligned} \tag{8.4}$$

Therefore, with the choice $c_{17} = 1$, the symmetry is

$$G_1 = v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} + u_n), \quad G_2 = v_n(u_{n+1}^2 - u_n^2) + v_n(v_{n+1} - v_{n-1}). \tag{8.5}$$

It is easy to produce new completely integrable DDEs based on these symmetries. For instance, the DDE system

$$\begin{aligned} \dot{u}_n &= G_1 = v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} + u_n), \\ \dot{v}_n &= G_2 = v_n(u_{n+1}^2 - u_n^2) + v_n(v_{n+1} - v_{n-1}). \end{aligned}$$

is also completely integrable.

Example 8.1 To illustrate the effectiveness of our algorithm to filter out integrable cases among systems of DDEs with parameters, consider a parameterized version of the Toda lattice,

$$\dot{u}_n = \alpha\, v_{n-1} - v_n, \quad \dot{v}_n = v_n\,(\beta\, u_n - u_{n+1}), \tag{8.6}$$

where α and β are *nonzero* constants. In [53], it was shown that (8.6) is completely integrable if and only if $\alpha = \beta = 1$.

Using our algorithm, one can easily compute the *compatibility conditions* for α and β, so that (8.6) admits a polynomial symmetry, say, of rank $(3, 4)$. The steps are the same as for (7.9). However, the linear system for the c_i is parameterized by α and β and must be analyzed carefully. This analysis leads to the condition $\alpha = \beta = 1$.

For $\alpha = \beta = 1$, (8.6) coincides with (7.9), for which we computed symmetries with ranks $(4, 5)$ and $(5, 6)$. They are:

$$\begin{aligned}
G_1^{(1)} &= u_n^2 v_n + u_n u_{n+1} v_n + u_{n+1}^2 v_n + v_n^2 + v_n v_{n+1} - u_{n-1}^2 v_{n-1} - u_{n-1} u_n v_{n-1} \\
&\quad - u_n^2 v_{n-1} - v_{n-2} v_{n-1} - v_{n-1}^2, \\
G_2^{(1)} &= u_{n+1} v_n^2 + 2u_{n+1} v_n v_{n+1} + u_{n+2} v_n v_{n+1} - u_n^3 v_n + u_{n+1}^3 v_n \\
&\quad - u_{n-1} v_{n-1} v_n - 2u_n v_{n-1} v_n - u_n v_n^2, \qquad\qquad (8.7) \\
G_1^{(2)} &= u_n^3 v_n + u_n^2 u_{n+1} v_n + u_n u_{n+1}^2 v_n + u_{n+1}^3 v_n + 2u_n v_n^2 + 2u_{n+1} v_n^2 \\
&\quad + u_n v_n v_{n+1} + 2u_{n+1} v_n v_{n+1} + u_{n+2} v_n v_{n+1} - u_{n-1}^3 v_{n-1} - u_{n-1}^2 u_n v_{n-1} \\
&\quad - u_{n-1} u_n^2 v_{n-1} - u_n^3 v_{n-1} - u_{n-2} v_{n-2} v_{n-1} - 2u_{n-1} v_{n-2} v_{n-1} \\
&\quad - u_n v_{n-2} v_{n-1} - 2u_{n-1} v_{n-1}^2 - 2u_n v_{n-1}^2, \\
G_2^{(2)} &= u_{n+1}^4 v_n - u_n^4 v_n - u_{n-1}^2 v_{n-1} v_n - 2u_{n-1} u_n v_{n-1} v_n - 3u_n^2 v_{n-1} v_n \\
&\quad - v_{n-2} v_{n-1} v_n - v_{n-1}^2 v_n - 2u_n^2 v_n^2 + 2u_{n+1}^2 v_n^2 - v_{n-1} v_n^2 + 3u_{n+1}^2 v_n v_{n+1}
\end{aligned}$$

$$+2u_{n+1}u_{n+2}v_nv_{n+1} + u_{n+2}^2v_nv_{n+1} + v_n^2v_{n+1} + v_nv_{n+1}^2 + v_nv_{n+1}v_{n+2}. \quad (8.8)$$

8.2 Nonuniform Systems

As in the PDE case, for scaling invariant systems such as (7.9), it suffices to consider the scaling symmetry on the space of independent and dependent variables. For systems that are not scaling invariant, such as the example given below, we use our trick: introduce one (or more) auxiliary parameter(s), and treat them as dependent variables with the appropriate scaling.

8.2.1 Ablowitz-Ladik Discretization of the NLS Equation

In [3, 4], Ablowitz and Ladik studied some of the properties of the following integrable discretization of the NLS equation:

$$i\,\dot{u}_n = u_{n+1} - 2u_n + u_{n-1} \pm u_n^* u_n(u_{n+1} + u_{n-1}), \quad (8.9)$$

where u_n^* is the complex conjugate of u_n. We continue with the plus sign; the other case would be treated analogously. Instead of splitting u_n into its real and imaginary parts, we treat u_n and $v_n = u_n^*$ as independent variables and augment (8.9) with its complex conjugate equation. Absorbing i into the scale on t, we get

$$\begin{aligned} \dot{u}_n &= u_{n+1} - 2u_n + u_{n-1} + u_nv_n(u_{n+1} + u_{n-1}), \\ \dot{v}_n &= -(v_{n+1} - 2v_n + v_{n-1}) - u_nv_n(v_{n+1} + v_{n-1}). \end{aligned} \quad (8.10)$$

Since $v_n = u_n^*$, we have $w(v_n) = w(u_n)$. Neither of the equations in (8.10) is uniform in rank. To circumvent this problem, we introduce an auxiliary parameter α with weight, and replace (8.10) by

$$\begin{aligned}\dot{u}_n &= \alpha(u_{n+1} - 2u_n + u_{n-1}) + u_n v_n (u_{n+1} + u_{n-1}),\\ \dot{v}_n &= -\alpha(v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}).\end{aligned} \tag{8.11}$$

Uniformity in rank requires that

$$\begin{aligned}w(u_n) + 1 &= w(\alpha) + w(u_n) = 2w(u_n) + w(v_n) = 3w(u_n),\\ w(v_n) + 1 &= w(\alpha) + w(v_n) = 2w(v_n) + w(u_n) = 3w(v_n),\end{aligned}$$

which yields $w(u_n) = w(v_n) = \frac{1}{2}, w(\alpha) = 1$, or, $u_n^2 \sim v_n^2 \sim \alpha \sim \frac{\mathrm{d}}{\mathrm{dt}}$.

Recall that the 'uniformity in rank' requirement is essential for the first step of the algorithm. However, after step 1, we may set $\alpha = 1$. The computations now proceed as in the previous example. We searched for symmetries of (8.10) of ranks $(2,2)$ through $(7/2, 7/2)$, and found symmetries of ranks $(5/2, 5/2)$ and $(7/2, 7/2)$. To save space, we only list the symmetries of rank $(5/2, 5/2)$:

$$\begin{aligned}G_1^{(1)} &= -u_{n+2} - u_n u_{n+1} v_{n-1} - u_{n+1}^2 v_n - u_n u_{n+2} v_n - u_n^2 u_{n+1} v_{n-1} v_n\\ &\quad -u_n u_{n+1}^2 v_n^2 - u_{n+1} u_{n+2} v_{n+1} - u_n u_{n+1} u_{n+2} v_n v_{n+1},\\ G_2^{(1)} &= v_{n-2} + u_{n-1} v_{n-2} v_{n-1} + u_n v_{n-1}^2 + u_n v_{n-2} v_n + u_{n+1} v_{n-1} v_n\\ &\quad +u_{n-1} u_n v_{n-2} v_{n-1} v_n + u_n^2 v_{n-1}^2 v_n + u_n u_{n+1} v_{n-1} v_n^2,\end{aligned} \tag{8.12}$$

$$\begin{aligned}G_1^{(2)} &= -u_{n-2} - u_{n-2} u_{n-1} v_{n-1} - u_{n-1}^2 v_n - u_{n-2} u_n v_n - u_{n-2} u_{n-1} u_n v_{n-1} v_n\\ &\quad -u_{n-1}^2 u_n v_n^2 - u_{n-1} u_n v_{n+1} - u_{n-1} u_n^2 v_n v_{n+1},\\ G_2^{(2)} &= u_{n-1} v_n v_{n+1} + u_{n-1} u_n v_n^2 v_{n+1} + u_n v_{n+1}^2 + u_n^2 v_n v_{n+1}^2\\ &\quad +v_{n+2} + u_n v_n v_{n+2} + u_{n+1} v_{n+1} v_{n+2} + u_n u_{n+1} v_n v_{n+1} v_{n+2}.\end{aligned} \tag{8.13}$$

Chapter 9

INVARIANTS OF LATTICES

In this chapter, we describe our algorithm for the computation of conservation laws of lattice equations. The algorithm is illustrated with an example. Also, the usefulness of the equivalence relation that we introduced in Chapter 7 becomes clear here.

Recall that for a system of DDEs,

$$\dot{\mathbf{u}}_n = \mathbf{F}(..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ...), \tag{9.1}$$

a local *conservation law* is of the form

$$\dot{\rho}_n = J_n - J_{n+1}, \tag{9.2}$$

where ρ_n is the *invariant* and J_n is the associated *flux*. Our algorithm is currently designed for $\dot{\rho}_n = J_{n-p} - J_{n+1}$ with $p = 0$. Modifications would be needed if $p = 1$ were used.

9.1 Algorithm

To illustrate the algorithm, once again, we consider the Toda lattice (7.9). Keeping in mind that $w(u_n) = 1$ and $w(v_n) = 2$, we now start the description of our algorithm.

9.1.1 Step 1: Construct the form of the invariant

As an example, let us compute the form of the invariant of rank 3. List all monomials in u_n and v_n of rank 3 or less: $\mathcal{G} = \{u_n^3, u_n^2, u_n v_n, u_n, v_n\}$.

Next, for each monomial in $\mathcal{G}$, introduce the necessary t-derivatives, so that each term has exactly weight 3. Thus, using (7.9),

$$\frac{\mathrm{d}^0}{\mathrm{dt}^0}(u_n^3) = u_n^3, \qquad \frac{\mathrm{d}^0}{\mathrm{dt}^0}(u_n v_n) = u_n v_n,$$

$$\frac{\mathrm{d}}{\mathrm{dt}}(u_n^2) = 2u_n\dot{u}_n = 2u_n v_{n-1} - 2u_n v_n, \qquad \frac{\mathrm{d}}{\mathrm{dt}}(v_n) = \dot{v}_n = u_n v_n - u_{n+1} v_n,$$

$$\frac{\mathrm{d}^2}{\mathrm{dt}^2}(u_n) = \frac{\mathrm{d}}{\mathrm{dt}}(\dot{u}_n) = \frac{\mathrm{d}}{\mathrm{dt}}(v_{n-1} - v_n) = u_{n-1}v_{n-1} - u_n v_{n-1} - u_n v_n + u_{n+1} v_n.$$

Gather the resulting terms in a set $\mathcal{H} = \{u_n^3, u_n v_{n-1}, u_n v_n, u_{n-1} v_{n-1}, u_{n+1} v_n\}$. Identify members that belong to the same equivalence classes and replace them by the main representatives. For example, since $u_n v_{n-1} \equiv u_{n+1} v_n$ the latter is replaced by $u_n v_{n-1}$. Doing so, $\mathcal{H}$ is replaced by $\mathcal{I} = \{u_n^3, u_n v_{n-1}, u_n v_n\}$, which contains the building blocks of the invariant. Linear combination of the monomials in $\mathcal{I}$ with constant coefficients c_i gives the form of the invariant:

$$\rho_n = c_1\, u_n^3 + c_2\, u_n v_{n-1} + c_3\, u_n v_n. \tag{9.3}$$

9.1.2 Step 2: Determine the unknown coefficients in the invariant

We determine the coefficients c_1 through c_3 by requiring that (9.2) holds. During this step, we also compute the unknown flux J_n.

Compute $\dot{\rho}_n$ using (9.3). Then use (7.9) to remove $\dot{u}_n, \dot{v}_n$, etc. After grouping the terms

$$\dot{\rho}_n = (3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n + (c_3 - c_2)v_{n-1}v_n$$
$$+c_2 u_{n-1}u_n v_{n-1} + c_2 v_{n-1}^2 - c_3 u_n u_{n+1} v_n - c_3 v_n^2. \tag{9.4}$$

Use the equivalence criterion to modify $\dot{\rho}_n$. For instance, replace $u_{n-1}u_n v_{n-1}$ by $u_n u_{n+1} v_n + [u_{n-1}u_n v_{n-1} - u_n u_{n+1} v_n]$. The goal is to introduce the main representatives. Therefore,

$$\dot{\rho}_n = (3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n \tag{9.5}$$
$$+(c_3 - c_2)v_n v_{n+1} + [(c_3 - c_2)v_{n-1}v_n - (c_3 - c_2)v_n v_{n+1}]$$
$$+c_2 u_n u_{n+1} v_n + [c_2 u_{n-1} u_n v_{n-1} - c_2 u_n u_{n+1} v_n]$$
$$+c_2 v_n^2 + [c_2 v_{n-1}^2 - c_2 v_n^2] - c_3 u_n u_{n+1} v_n - c_3 v_n^2. \tag{9.6}$$

Next, group the terms outside of the square brackets and move the pairs inside the square brackets to the bottom. Rearrange the latter terms so that they match the pattern $[J_n - J_{n+1}]$. Hence,

$$\dot{\rho}_n = (3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n$$
$$+(c_3 - c_2)v_n v_{n+1} + (c_2 - c_3)u_n u_{n+1} v_n + (c_2 - c_3)v_n^2$$
$$+[\{(c_3 - c_2)v_{n-1}v_n + c_2 u_{n-1}u_n v_{n-1} + c_2 v_{n-1}^2\}$$
$$-\{(c_3 - c_2)v_n v_{n+1} + c_2 u_n u_{n+1} v_n + c_2 v_n^2\}]. \tag{9.7}$$

The terms inside the square brackets determine:

$$J_n = (c_3 - c_2)v_{n-1}v_n + c_2 u_{n-1}u_n v_{n-1} + c_2 v_{n-1}^2. \tag{9.8}$$

The terms outside the square brackets must all vanish, yielding

$$\mathcal{S} = \{3c_1 - c_2 = 0, c_3 - 3c_1 = 0, c_2 - c_3 = 0\}. \tag{9.9}$$

The solution is $3c_1 = c_2 = c_3$. Since invariants can only be determined up to a multiplicative constant, we choose $c_1 = \frac{1}{3}$, so, $c_2 = c_3 = 1$, and substitute this into (9.3) and (9.8). Hence,

$$\rho_n = \tfrac{1}{3}\,u_n^3 + u_n(v_{n-1} + v_n), \qquad J_n = u_{n-1}u_n v_{n-1} + v_{n-1}^2.$$

Analogously, we computed invariants of rank ≤ 5 for (7.9). They are:

$$\rho_n^{(1)} = u_n, \quad \rho_n^{(2)} = \tfrac{1}{2}u_n^2 + v_n, \quad \rho_n^{(3)} = \tfrac{1}{3}u_n^3 + u_n(v_{n-1} + v_n), \tag{9.10}$$

$$\rho_n^{(4)} = \tfrac{1}{4}u_n^4 + u_n^2(v_{n-1} + v_n) + u_n u_{n+1} v_n + \tfrac{1}{2}v_n^2 + v_n v_{n+1}, \tag{9.11}$$

$$\begin{aligned} \rho_n^{(5)} &= \tfrac{1}{5}u_n^5 + u_n^3(v_{n-1} + v_n) + u_n u_{n+1} v_n (u_n + u_{n+1}) \\ &\quad + u_n v_{n-1}(v_{n-2} + v_{n-1} + v_n) + u_n v_n (v_{n-1} + v_n + v_{n+1}). \end{aligned} \tag{9.12}$$

Ignoring irrelevant shifts in n, these invariants agree with the results in [25].

Example 9.1 Consider the parameterized Toda lattice (8.6). In [53] it was shown that (8.6) is completely integrable if $\alpha = \beta = 1$.

Using our algorithm, we search for the conditions on α and β, so that (8.6) admits a polynomial invariant of, say, rank 3. The steps are the same as for (7.9). However, (9.9) must be replaced by

$$\mathcal{S} = \{3\alpha c_1 - c_2 = 0, \beta c_3 - 3c_1 = 0, \alpha c_3 - c_2 = 0, \beta c_2 - c_3 = 0, \alpha c_2 - c_3 = 0\}.$$

A non-trivial solution $3c_1 = c_2 = c_3$ will exist if and only if $\alpha = \beta = 1$.

Analogously, (8.6) has invariant $\rho_n^{(1)} = u_n$ of rank 1 if $\alpha = 1$, and invariant $\rho_n^{(2)} = \frac{\beta}{2}u_n^2 + v_n$ of rank 2 if $\alpha\beta = 1$. Only when $\alpha = \beta = 1$ will (8.6) have invariants of rank ≥ 3.

9.2 Nonuniform Systems

For nonuniform systems, we introduce auxiliary parameters with weights.

Example 9.2 Working with the discretization of the NLS equation (8.9) written as (8.10), we were able to compute the invariants. Here are some of them:

$$\begin{aligned}
\rho_n^{(1)} &= c_1 u_n v_{n-1} + c_2 u_n v_{n+1}, && (9.13)\\
\rho_n^{(2)} &= c_1\left(\tfrac{1}{2}u_n^2 v_{n-1}^2 + u_n u_{n+1} v_{n-1} v_n + u_n v_{n-2}\right)\\
&+ c_2\left(\tfrac{1}{2}u_n^2 v_{n+1}^2 + u_n u_{n+1} v_{n+1} v_{n+2} + u_n v_{n+2}\right), && (9.14)\\
\rho_n^{(3)} &= c_1\left[\tfrac{1}{3}u_n^3 v_{n-1}^3 + u_n u_{n+1} v_{n-1} v_n (u_n v_{n-1} + u_{n+1} v_n + u_{n+2} v_{n+1})\right.\\
&+ \left. u_n v_{n-1}(u_n v_{n-2} + u_{n+1} v_{n-1}) + u_n v_n (u_{n+1} v_{n-2} + u_{n+2} v_{n-1}) + u_n v_{n-3}\right]\\
&+ c_2\left[\tfrac{1}{3}u_n^3 v_{n+1}^3 + u_n u_{n+1} v_{n+1} v_{n+2}(u_n v_{n+1} + u_{n+1} v_{n+2} + u_{n+2} v_{n+3})\right.\\
&+ \left. u_n v_{n+2}(u_n v_{n+1} + u_{n+1} v_{n+2}) + u_n v_{n+3}(u_{n+1} v_{n+1} + u_{n+2} v_{n+2}) + u_n v_{n+3}\right]. && (9.15)
\end{aligned}$$

Our results confirm those in [3]. Also, if defined on an infinite interval, (8.9) admits infinitely many independent conserved densities [3]. Although it is a constant of motion, we cannot find the Hamiltonian of (8.9),

$$H = -i\sum_n [u_n^*(u_{n-1} + u_{n+1}) - 2\ln(1 + u_n u_n^*)], \qquad (9.16)$$

for it has a logarithmic term [2].

Chapter 10

DDE CASE: EXAMPLES

In this chapter, we list symmetries and invariants for some lattice systems.

10.1 Discretizations of the KdV Equation

Consider the following integrable discretization of the KdV equation:

$$\dot{u}_n = u_n\,(u_{n+1} - u_{n-1}), \tag{10.1}$$

which is also known as the Kac-Van Moerbeke equation or a special form of the Volterra system. It arises in the study of Langmuir oscillations in plasmas and in population dynamics [1, 38, 69].

Notice that (10.1) is invariant under the scaling symmetry $(t, u_n) \rightarrow (\lambda^{-1}t, \lambda u_n)$.

We computed the symmetries of (10.1) with ranks 3 through 5. They are:

$$\begin{aligned}
G^{(1)} &= u_n u_{n+1}(u_n + u_{n+1} + u_{n+2}) - u_{n-1}u_n(u_{n-2} + u_{n-1} + u_n), && (10.2)\\
G^{(2)} &= u_n^3 u_{n+1} + 2u_n^2 u_{n+1}^2 + u_n u_{n+1}^3 + u_n^2 u_{n+1} u_{n+2} + 2u_n u_{n+1}^2 u_{n+2} \\
&\quad + u_n u_{n+1} u_{n+2}^2 + u_n u_{n+1} u_{n+2} u_{n+3} - u_{n-3}u_{n-2}u_{n-1}u_n - u_{n-2}^2 u_{n-1} u_n \\
&\quad - 2u_{n-2}u_{n-1}^2 u_n - u_{n-1}^3 u_n - u_{n-2}u_{n-1}u_n^2 - u_{n-1}u_n^3, && (10.3)\\
G^{(3)} &= u_n^4 u_{n+1} + u_{n-1}u_n^2 u_{n+1}^2 + 3u_n^3 u_{n+1}^2 + 3u_n^2 u_{n+1}^3 + u_n u_{n+1}^4 + u_n^3 u_{n+1} u_{n+2} \\
&\quad + 4u_n^2 u_{n+1}^2 u_{n+2} + 3u_n u_{n+1}^3 u_{n+2} + u_n^2 u_{n+1} u_{n+2}^2 + 3u_n u_{n+1}^2 u_{n+2}^2 \\
&\quad + u_n u_{n+1} u_{n+2}^3 + u_n^2 u_{n+1} u_{n+2} u_{n+3} + 2u_n u_{n+1}^2 u_{n+2} u_{n+3} && (10.4)
\end{aligned}$$

$$+u_n u_{n+1} u_{n+2} u_{n+3}^2 + u_n u_{n+1} u_{n+2} u_{n+3} u_{n+4} - u_{n-4} u_{n-3} u_{n-2} u_{n-1} u_n$$
$$-u_{n-3}^2 u_{n-2} u_{n-1} u_n - 2u_{n-3} u_{n-2}^2 u_{n-1} u_n - u_{n-2}^3 u_{n-1} u_n$$
$$-2u_{n-3} u_{n-2} u_{n-1}^2 u_n - 3u_{n-2}^2 u_{n-1}^2 u_n - 3u_{n-2} u_{n-1}^3 u_n - u_{n-1}^4 u_n$$
$$-u_{n-3} u_{n-2} u_{n-1} u_n^2 - u_{n-2} u_{n-1}^2 u_n^2 - 4u_{n-2} u_{n-1}^2 u_n^2 - 3u_{n-1}^3 u_n^2$$
$$-u_{n-2} u_{n-1} u_n^3 - 3u_{n-1}^2 u_n^3 - u_{n-1} u_n^4 - u_{n-1}^2 u_n^2 u_{n+1}. \qquad (10.5)$$

Ignoring a trivial misprint in [47], Mikhailov *et al.* listed the symmetry $\dot{G}^{(1)}$.

Analogously, for (10.1) we computed the invariants of rank ≤ 5:

$$\rho_n^{(1)} = u_n, \qquad \rho_n^{(2)} = \tfrac{1}{2}u_n^2 + u_n u_{n+1}, \qquad (10.6)$$
$$\rho_n^{(3)} = \tfrac{1}{3}u_n^3 + u_n u_{n+1}(u_n + u_{n+1} + u_{n+2}), \qquad (10.7)$$
$$\rho_n^{(4)} = \tfrac{1}{4}u_n^4 + u_n^3 u_{n+1} + \tfrac{3}{2}u_n^2 u_{n+1}^2 + u_n u_{n+1}^2(u_{n+1} + u_{n+2})$$
$$+u_n u_{n+1} u_{n+2}(u_n + u_{n+1} + u_{n+2} + u_{n+3}), \qquad (10.8)$$
$$\rho_n^{(5)} = \tfrac{1}{5}u_n^5 + u_n u_{n+1}(u_n^3 + u_{n+1}^3) + 2u_n^2 u_{n+1}^2(u_n + u_{n+1})$$
$$+u_n u_{n+1} u_{n+2}(u_n^2 + u_n u_{n+2} + u_{n+1} u_{n+3}) + 3u_n u_{n+1}^2 u_{n+2}$$
$$(u_n + u_{n+1} + u_{n+2}) + u_n u_{n+1} u_{n+2}^2(u_{n+2} + u_{n+3})$$
$$+u_n u_{n+1} u_{n+2} u_{n+3}(u_n + u_{n+1} + u_{n+2} + u_{n+3} + u_{n+4}). \qquad (10.9)$$

We also computed invariants for the following completely integrable discretization [66]

$$\dot{u}_n = (1 + \tfrac{1}{6}u_n)\Big[\tfrac{1}{2}(u_{n+2} - u_{n-2}) - (u_{n+1} - u_{n-1}) + \tfrac{1}{12}[u_{n+1}^2 - u_{n-1}^2 +$$
$$u_{n+1}(u_n + u_{n+2}) - u_{n-1}(u_n + u_{n-2})]\Big], \qquad (10.10)$$

of the KdV equation $u_t + uu_x + u_{3x} = 0$.

To make (10.10) uniform in rank, we introduce auxiliary parameters α and β with weight, and replace (10.10) by

$$\begin{aligned} \dot{u}_n &= (\alpha + \tfrac{1}{6}u_n)\Big[\beta[\tfrac{1}{2}(u_{n+2} - u_{n-2}) - (u_{n+1} - u_{n-1})] + \tfrac{1}{12}[u_{n+1}^2 - u_{n-1}^2 + \\ &\quad u_{n+1}(u_n + u_{n+2}) - u_{n-1}(u_n + u_{n-2})]\Big], \end{aligned} \tag{10.11}$$

then, $u_n^2 \sim \alpha^2 \sim \beta^2 \sim \frac{\mathrm{d}}{\mathrm{dt}}$. We computed the invariants of rank $\frac{1}{2}$ and $\frac{3}{2}$ which, upon decomposition in independent pieces, yield

$$\rho_n^{(1)} = u_n, \qquad \rho_n^{(2)} = u_n(\tfrac{1}{2}u_n + u_{n+1}), \tag{10.12}$$

$$\rho_n^{(3)} = u_n(\tfrac{1}{3}u_n^2 + u_n u_{n+1} + u_{n+1}^2 + 6u_{n+2} + u_{n+1}u_{n+2}). \tag{10.13}$$

Two more independent invariants for rank $\frac{5}{2}$ were computed.

10.2 Extended Lotka-Volterra Equation

Itoh [37] studied the following extended version of the Lotka-Volterra equation (10.1),

$$\dot{u}_n = \sum_{r=1}^{k-1}(u_{n-r} - u_{n+r})u_n. \tag{10.14}$$

For $k = 2$, (10.14) is (10.1), for which we listed its symmetries and invariants in the previous section.

For (10.14), we computed 5 invariants and 2 higher-order symmetries for $k = 3$ through $k = 5$. Here is a partial list of our results:

Case 1: $k = 3$

Invariants:

$$\rho_1 = u_n, \qquad \rho_2 = \frac{1}{2}u_n^2 + u_n(u_{n+1} + u_{n+2}), \tag{10.15}$$

$$\begin{aligned} \rho_3 &= \frac{1}{3}u_n^3 + u_n^2(u_{n+1} + u_{n+2}) + u_n(u_{n+1} + u_{n+2})^2 \\ &\quad + u_n(u_{n+1}u_{n+3} + u_{n+2}u_{n+3} + u_{n+2}u_{n+4}). \end{aligned} \tag{10.16}$$

Higher-order symmetry:

$$\begin{aligned} G &= u_n^2(u_{n+1} + u_{n+2} - u_{n-2} - u_{n-1}) \\ &\quad + u_n[(u_{n+1} + u_{n+2})^2 - (u_{n-2} + u_{n-1})^2] \\ &\quad + u_n[u_{n+1}u_{n+3} + u_{n+2}u_{n+3} + u_{n+2}u_{n+4} \\ &\quad - (u_{n-4}u_{n-2} + u_{n-3}u_{n-2} + u_{n-3}u_{n-1})]. \end{aligned} \tag{10.17}$$

Case 2: $k = 4$

Invariants:

$$\rho_1 = u_n, \qquad \rho_2 = \frac{1}{2}u_n^2 + u_n(u_{n+1} + u_{n+2} + u_{n+3}), \tag{10.18}$$

$$\begin{aligned} \rho_3 &= \frac{1}{3}u_n^3 + u_n^2(u_{n+1} + u_{n+2} + u_{n+3}) + u_n(u_{n+1} + u_{n+2} + u_{n+3})^2 \\ &\quad + u_n(u_{n+1}u_{n+4} + u_{n+2}u_{n+4} + u_{n+3}u_{n+4} + u_{n+2}u_{n+5} \\ &\quad + u_{n+3}u_{n+5} + u_{n+3}u_{n+6}). \end{aligned} \tag{10.19}$$

Higher-order symmetry:

$$G = u_n[u_{n+1}u_{n+4} + u_{n+2}u_{n+4} + u_{n+3}u_{n+4} + u_{n+2}u_{n+5} + u_{n+3}u_{n+5} + u_{n+3}u_{n+6}$$

$$-(u_{n-6}u_{n-3} + u_{n-5}u_{n-3} + u_{n-4}u_{n-3} + u_{n-5}u_{n-2} - u_{n-4}u_{n-2} + u_{n-4}u_{n-1})]$$
$$+u_n[(u_{n+1} + u_{n+2} + u_{n+3})^2 - u_n(u_{n-3} + u_{n-2} + u_{n-1})^2]$$
$$+u_n^2[u_{n+1} + u_{n+2} + u_{n+3} - (u_{n-3} + u_{n-2} + u_{n-1})]. \qquad (10.20)$$

The integrability and other properties of (10.14) are discussed in [33].

10.3 A Discretized Modified KdV (MKdV) Equation

In [1], we found the following integrable discretization of the MKdV equation:

$$\dot{u}_n = (1 + u_n^2)(u_{n+1} - u_{n-1}). \qquad (10.21)$$

We computed four invariants of (10.21). The first three are:

$$\rho_n^{(1)} = u_n u_{n+1}, \qquad \rho_n^{(2)} = \tfrac{1}{2}u_n^2 u_{n+1}^2 + u_n u_{n+2}(1 + u_{n+1}^2), \qquad (10.22)$$
$$\rho_n^{(3)} = \tfrac{1}{3}u_n^3 u_{n+1}^3 + u_n u_{n+1} u_{n+2}(u_n + u_{n+2})(1 + u_{n+1}^2)$$
$$+u_n u_{n+3}(1 + u_{n+1}^2)(1 + u_{n+2}^2). \qquad (10.23)$$

10.4 Self-Dual Network Equations

The integrable nonlinear self-dual network equations [1, 62] can be written as:

$$\dot{u}_n = (1 + u_n^2)(v_n - v_{n-1}), \qquad \dot{v}_n = (1 + v_n^2)(u_{n+1} - u_n). \qquad (10.24)$$

We computed the first four invariants of (10.24). The first three are

$$\rho_n^{(1)} = u_n v_{n-1} + u_n v_n, \qquad (10.25)$$

$$\rho_n^{(2)} = \tfrac{1}{2}u_n^2(v_{n-1}^2+v_n^2)+u_nu_{n+1}(1+v_n^2)+v_n(u_n^2v_{n-1}+v_{n+1}), \tag{10.26}$$

$$\begin{aligned}\rho_n^{(3)} &= \tfrac{1}{3}u_n^3(v_{n-1}^3+v_n^3)+u_nu_{n+1}(u_nv_{n-1}+u_{n+1}v_n+u_nv_n)(1+v_n^2)\\ &+u_nv_{n-2}(1+v_{n-1}^2)+u_nv_{n-1}v_n(v_{n-1}+v_n)(1+u_n^2)\\ &+u_nv_{n+1}(1+u_{n+1}^2)(1+v_n^2).\end{aligned} \tag{10.27}$$

10.5 Generalized Toda Lattices

The integrable relativistic Toda lattice [50] is given as:

$$\dot{u}_n = u_n\,(v_{n+1}-v_n+u_{n+1}-u_{n-1}), \quad \dot{v}_n = v_n\,(u_n-u_{n-1}). \tag{10.28}$$

We computed the first five invariants of (10.28). Here we list only the first three:

$$\rho_1 = u_n+v_n, \qquad \rho_2 = \tfrac{1}{2}(u_n^2+v_n^2)+u_n(u_{n+1}+v_n+v_{n+1}), \tag{10.29}$$

$$\begin{aligned}\rho_3 &= \tfrac{1}{3}(u_n^3+v_n^3)+u_n^2(u_{n+1}+v_n+v_{n+1})+u_n[(u_{n+1}+v_{n+1})^2\\ &+u_{n+1}u_{n+2}+u_{n+1}v_n+u_{n+1}v_{n+2}+v_nv_{n+1}+v_n^2].\end{aligned} \tag{10.30}$$

Also the first two symmetries are:

$$G_1^{(1)} = u_n(u_{n+1}-u_{n-1}+v_{n+1}-v_n), \qquad G_2^{(1)}=v_n(u_n-u_{n-1}), \tag{10.31}$$

$$\begin{aligned}G_1^{(2)} &= u_n^2(u_{n+1}-u_{n-1}+v_{n+1}-v_n)+u_n[(u_{n+1}+v_{n+1})^2-(u_{n-1}+v_n)^2\\ &+u_{n+1}(u_{n+2}+v_{n+2})-u_{n-1}(u_{n-2}+v_{n-1})],\end{aligned} \tag{10.32}$$

$$\begin{aligned}G_2^{(2)} &= v_n^2(u_n-u_{n-1})+v_n(u_n^2-u_{n-1}u_{n-2}-u_{n-1}^2+u_nu_{n+1}\\ &-u_{n-1}v_{n-1}+u_nv_{n+1}).\end{aligned} \tag{10.33}$$

In [64], the integrability of the chain

$$\ddot{y}_n = \dot{y}_{n+1} e^{(y_{n+1}-y_n)} - e^{2(y_{n+1}-y_n)} - \dot{y}_{n-1} e^{(y_n-y_{n-1})} + e^{2(y_n-y_{n-1})}, \tag{10.34}$$

which is related to the relativistic Toda lattice has been studied. With the change of variables, $u_n = \dot{y}_n$, $v_n = \exp(y_{n+1} - y_n)$, lattice (10.34) can be written as

$$\dot{u}_n = v_n(u_{n+1} - v_n) - v_{n-1}(u_{n-1} - v_{n-1}), \qquad \dot{v}_n = v_n(u_{n+1} - u_n). \tag{10.35}$$

Here, $u_n \sim v_n \sim \frac{\mathrm{d}}{\mathrm{dt}}$. We computed a couple of symmetries for (10.35). One of them reads:

$$\begin{aligned} G_1 &= u_{n-1}^2 v_{n-1} + u_{n-1} u_n v_{n-1} + u_{n-2} v_{n-2} v_{n-1} - v_{n-2}^2 v_{n-1} - 2u_{n-1} v_{n-1}^2 \\ &\quad - u_n v_{n-1}^2 + v_{n-1}^3 - u_n u_{n+1} v_n - u_{n+1}^2 v_n + u_n v_n^2 + 2u_{n+1} v_n^2 \\ &\quad - v_n^3 - u_{n+2} v_n v_{n+1} + v_n v_{n+1}^2, \\ G_2 &= u_n^2 v_n - u_{n+1}^2 v_n + u_{n-1} v_{n-1} v_n - v_{n-1}^2 v_n - u_n v_n^2 + u_{n+1} v_n^2 \\ &\quad - u_{n+2} v_n v_{n+1} + v_n v_{n+1}^2. \end{aligned} \tag{10.36}$$

Also, we computed five invariants for (10.35). The first three are:

$$\rho_n^{(1)} = u_n - v_n, \qquad \rho_n^{(2)} = u_n^2 - v_n^2, \tag{10.37}$$

$$\rho_n^{(3)} = \tfrac{1}{3}(u_n^3 + 2v_n^3) - u_n(v_{n-1}^2 + v_n^2) + u_n u_{n+1} v_n. \tag{10.38}$$

In [65], Suris also investigated the integrability of

$$\ddot{y}_n = \dot{y}_n \left[\exp(y_{n+1} - y_n) - \exp(y_n - y_{n-1})\right], \tag{10.39}$$

which is closely related to the classical Toda lattice (7.7). The same change of variables as for (10.34) allows one to rewrite (10.39) as

$$\dot{u}_n = u_n(v_n - v_{n-1}), \qquad \dot{v}_n = v_n(u_{n+1} - u_n). \tag{10.40}$$

Again, $u_n \sim v_n \sim \frac{\mathrm{d}}{\mathrm{dt}}$, and we have computed three symmetries. Two of them are:

$$\begin{aligned}
G_1^{(1)} &= u_n^2 v_n + u_n u_{n+1} v_n + u_n v_n^2 - u_{n-1} u_n v_{n-1} - u_n^2 v_{n-1} - u_n v_{n-1}^2, \\
G_2^{(1)} &= u_{n+1}^2 v_n + u_{n+1} v_n^2 + u_{n+1} v_n v_{n+1} - u_n^2 v_n - u_n v_{n-1} v_n - u_n v_n^2, \qquad (10.41) \\
G_1^{(2)} &= u_n^3 v_n + u_n^2 u_{n+1} v_n + u_n u_{n+1}^2 v_n + 2u_n^2 v_n^2 + 2u_n u_{n+1} v_n^2 + u_n v_n^3 \\
&\quad + u_n u_{n+1} v_n v_{n+1} - u_{n-1}^2 u_n v_{n-1} - u_{n-1} u_n^2 v_{n-1} - u_n^3 v_{n-1} \\
&\quad - u_{n-1} u_n v_{n-2} v_{n-1} - 2u_{n-1} u_n v_{n-1}^2 - 2u_n^2 v_{n-1}^2 - u_n v_{n-1}^3, \qquad (10.42) \\
G_2^{(2)} &= u_{n+1}^3 v_n - u_n^3 v_n - u_{n-1} u_n v_{n-1} v_n - 2u_n^2 v_{n-1} v_n - u_n v_{n-1}^2 v_n - 2u_n^2 v_n^2 \\
&\quad + 2u_{n+1}^2 v_n^2 - u_n v_{n-1} v_n^2 - u_n v_n^3 + u_{n+1} v_n^3 + 2u_{n+1}^2 v_n v_{n+1} \\
&\quad + u_{n+1} u_{n+2} v_n v_{n+1} + u_{n+1} v_n^2 v_{n+1} + u_{n+1} v_n v_{n+1}^2. \qquad (10.43)
\end{aligned}$$

Also the first four invariants of (10.40) are

$$\begin{aligned}
\rho_n^{(1)} &= u_n + v_n, \qquad \rho_n^{(2)} = \tfrac{1}{2}(u_n^2 + v_n^2) + u_n(v_{n-1} + v_n), \qquad (10.44) \\
\rho_n^{(3)} &= \tfrac{1}{3}(u_n^3 + v_n^3) + u_n^2(v_{n-1} + v_n) + u_n(v_{n-1}^2 + v_n^2) + u_n v_n(v_{n-1} + u_{n+1}), \qquad (10.45) \\
\rho_n^{(4)} &= \tfrac{1}{4}(u_n^4 + v_n^4) + u_n^3(v_{n-1} + v_n) + u_n(v_{n-1}^3 + v_n^3) + \tfrac{3}{2}u_n^2(v_{n-1}^2 + v_n^2) \\
&\quad + u_n u_{n+1} v_n(u_n + u_{n+1}) + 2u_n v_n(u_n v_{n-1} + u_{n+1} v_n) \\
&\quad + u_n v_{n-1} v_n(v_{n-1} + v_n) + u_n u_{n+1} v_n(v_{n-1} + v_{n+1}). \qquad (10.46)
\end{aligned}$$

10.6 Generalized Lattices

Shabat and Yamilov [62] studied the following integrable Volterra lattice:

$$\dot{u}_n = u_n(v_{n+1} - v_n), \qquad \dot{v}_n = v_n(u_n - u_{n-1}). \tag{10.47}$$

With our program, we computed the first four invariants for this system:

$$\rho_n^{(1)} = u_n + v_n, \qquad \rho_n^{(2)} = \tfrac{1}{2}(u_n^2 + v_n^2) + u_n(v_n + v_{n+1}), \tag{10.48}$$

$$\rho_n^{(3)} = \tfrac{1}{3}(u_n^3+v_n^3)+u_n^2(v_n+v_{n+1})+u_n(v_n^2+v_{n+1}^2)+u_nv_{n+1}(u_{n+1}+v_n), \tag{10.49}$$

$$\begin{aligned}\rho_n^{(4)} &= \tfrac{1}{4}(u_n^4 + v_n^4) + u_n^3(v_n + v_{n+1}) + \tfrac{3}{2}u_n^2(v_n^2 + v_{n+1}^2) \\ &\quad + u_n(v_n^3 + v_{n+1}^3) + 2u_nv_{n+1}(u_n + u_{n+1}) \\ &\quad + u_nu_{n+1}v_{n+1}(u_n + u_{n+1} + v_n + v_{n+2}) + u_nv_nv_{n+1}(v_n + v_{n+1}).\end{aligned} \tag{10.50}$$

In [62], the following Hamiltonian lattice is also studied:

$$\dot{u}_n = u_{n+1} + u_n^2v_n, \qquad \dot{v}_n = -v_{n-1} - u_nv_n^2. \tag{10.51}$$

Four invariants of (10.51) are:

$$\rho_n^{(1)} = u_nv_n, \qquad \rho_n^{(2)} = \tfrac{1}{2}u_n^2v_n^2 + u_nv_{n-1}, \tag{10.52}$$

$$\rho_n^{(3)} = \tfrac{1}{3}u_n^3v_n^3 + u_nv_n(u_nv_{n-1} + u_{n+1}v_n) + u_nv_{n-2}, \tag{10.53}$$

$$\begin{aligned}\rho_n^{(4)} &= \tfrac{1}{4}u_n^4v_n^4 + u_n^2v_n^2(u_nv_{n-1} + u_{n+1}v_n) + \tfrac{1}{2}u_n^2v_{n-1}^2 + u_nv_{n-3} \\ &\quad + u_nv_n(u_nv_{n-2} + 2u_{n+1}v_{n-1} + u_{n+2}v_n + u_{n+1}^2v_nv_{n+1}).\end{aligned} \tag{10.54}$$

Chapter 11

SOFTWARE PACKAGES

We briefly review the software packages related to the symbolic computation of higher-order symmetries and conservation laws. In the first section, we describe the features, scope and limitations of our package. In the second section, we talk about other software packages. Apart from ours, we are not aware of any software to compute higher-order symmetries and conservation laws for DDEs.

11.1 The Integrability Package

We now briefly describe the use of our *Integrability Package*, which has the code for the computation of higher-order symmetries and invariants based on the algorithms described in this thesis. The program does not yet compute the recursion operators. The *Integrability Package* is written in *Mathematica* [72] syntax. Users are assumed to have access to *Mathematica* 3.0. All the necessary files are available in *MathSource* [21] including on-line help, documentation, and built-in examples. The corresponding files should be put in the appropriate places on your platform. Detailed instructions are given in the documentation. After launching *Mathematica*, type

```
In[1]:= <<Integrability`
```

to read in the code. Doing so, you will get the following statement:

```
Loading init.m for Integrability from AddOns.
```

For the purpose of symmetry computations, the functions **PDESymmetries** and **DDESymmetries**, and for the computation of invariants, the functions **PDEIn-**

variants and **DDEInvariants** are available.

Working with (5.12) as an example, the first two lines define the system ($r = q^*$), whereas the third line will produce the three symmetries listed in (5.13):

```
In[2]:= pde1:= D[q[x,t],t] - D[q[x,t],{x,2}] +
              2*q[x,t]^2*r[x,t] == 0;


In[3]:= pde2:= D[r[x,t],t] + D[r[x,t],{x,2}] -
              2*r[x,t]^2*q[x,t] == 0;


In[4]:= PDESymmetries[{pde1,pde2},{q,r},{x,t},{4,6},
            WeightRules->{Weight[q]->Weight[r]}]
```

Help about the functions and their options is obtained by typing

```
In[5]:= ??DDESymmetries

DDESymmetries[eqn, u, {n, t}, R, opts] finds the symmetry with rank
   R of a differential-difference equation for the function u.
   DDESymmetries[{eqn1, eqn2, ...}, {u1, u2, ...}, {n, t}, R, opts]
   finds the symmetry of a system of differential-difference
   equations, where R is the rank of the first equation in the
   desired symmetry. DDESymmetries[{eqn1, eqn2, ...},
   {u1, u2, ...}, {n, t}, {Rmax}, opts] finds the symmetries with
   rank 0 through Rmax. DDESymmetries[{eqn1, eqn2, ...},
   {u1, u2, ...}, {n, t}, {Rmin, Rmax}, opts] finds the symmetries
   with rank Rmin through Rmax. n is understood as the discrete
   space variable and t as the time variable.
```

```
Attributes[DDESymmetries] = {Protected, ReadProtected}

Options[DDESymmetries] =
  {WeightedParameters -> {}, WeightRules -> {},
   MaxExplicitDependency -> 0, UndeterminedCoefficients -> C}
```

and

```
In[6]:= ??WeightedParameters
WeightedParameters is an option that determines the parameters with
   weight. If WeightedParameters -> {p1, p2, ...},
   then p1, p2, .... are considered as constant parameters with
   weight. The default is WeightedParameters -> {}.

Attributes[WeightedParameters] = {Protected}
```

The option **WeightedParameters** is useful when working with systems that lack uniformity in rank. In such cases, the code tries to resolve the problem of lack of uniformity, and prints appropriate messages. If the code cannot automatically resolve the problem it suggests the use of the **WeightedParameters** option. Therefore, one should not use the option **WeightedParameters** unless it is explicitly suggested. For further descriptions of the functions and their options, we refer to the documentation in [21].

Our program can handle systems of evolutionary PDEs and systems of first order DDEs, that are polynomial in the dependent variables. For PDEs and DDEs, only two independent variables are allowed. In the latter case, one of the independent variables is discretized. No terms in the equations should have coefficients that *explicitly* depend on the independent variables. In contrast to the PDE case, the program only computes polynomial invariants and symmetries in the dependent variables and their shifts, without explicit dependencies on the independent variables. For PDEs, the **MaxExplicitDependency** option allows one to compute invariants or symmetries that explicitly depend on the independent variables.

Theoretically, there is no limit on the number of equations. In practice, for large systems, the computations may take a long time or require a lot of memory. The computational speed depends primarily on the amount of memory.

By design, the program constructs only symmetries and invariants that are uniform in rank. The uniform rank assumption for the monomials in symmetries and invariants allows one to compute *independent* invariants and symmetries piece by piece, without having to split linear combinations. Due to the superposition principle, a linear combination of invariants or symmetries of unequal rank is still an invariant or a symmetry. This situation arises frequently when parameters with weight are introduced in the system.

The input systems may have one or more parameters, which are assumed to be nonzero. If a system has parameters, the program will attempt to compute the compatibility conditions for these parameters such that symmetries or invariants (of a given rank) exist. The assumption that all parameters in the given system must be nonzero is necessary. Indeed, as a result of setting parameters to zero in a given system of equations, the scaling properties might change, which is the basis for our algorithms.

In general, the compatibility conditions for the parameters could be highly nonlinear, and there is no general algorithm to solve them. The program automatically generates the compatibility conditions, and solves them for parameters that occur linearly. Gröbner basis techniques could be used to reduce complicated nonlinear systems into equivalent, yet simpler, non-linear systems.

The assumption that the systems are uniform in rank is not very restrictive. If the uniformity condition is violated, parameters with weights can be introduced. This also allows for some flexibility in the form of the symmetries and invariants. Although built up with terms that are uniform in rank, they do not have to be uniform in rank with respect to the dependent variables *alone*.

For systems with free weights, the user can fix these free weights by using the option **WeightRules**. Negative weights for the dependent variables are not allowed. Zero weights are allowed, but at least one of the dependent variables must have positive weight.

Our program is a tool in the search of the first few conservation laws and symmetries. An existence proof (showing that there are indeed infinitely many) must be done analytically. If our program succeeds in finding a large set of independent conservation laws or symmetries, there is a good chance that the system has infinitely many. For instance, if the number of conservation laws is 3 or less, most likely the system is not integrable, at least not in that coordinate representation.

11.2 Other Software Packages

This section gives a review of other software for the computation of higher-order symmetries and invariants.

Higher-order symmetries can be computed with prolongation methods and numerous software packages are available that can aid in the tedious computations inherent to such methods. An extensive review of software for Lie symmetry computations, including generalized symmetries, can be found in [26, 27]. With prolongation methods, one generates, subsequently reduces and then solves a determining system of linear homogeneous partial differential equations for the unknown higher-order symmetry. In many cases, due to the length and complexity of that system, the general solution is out of reach and one resorts to making a polynomial *ansatz* for the symmetry.

Although restricted to polynomial higher-order symmetries and invariants, we believe that the methods presented in this thesis are much more straightforward. Furthermore, they do not require the application of prolongation methods or Lie algebraic techniques.

To avoid repeating the surveys [26, 27] here, we restrict our discussion to symbolic packages that allow one to compute generalized symmetries of PDEs, as they were defined in Chapter 2. We are not aware of software for DDEs to calculate symmetries and conservation laws.

Based on the alternative strategy discussed in Remark 2.1 and dilation invariance, Ito's programs in REDUCE [34, 35, 36] compute polynomial higher-order symmetries and invariants for systems of evolution equations that are uniform in rank (no weighted parameters can be introduced). Ito's programs cannot be used to compute symmetries and invariants that explicitly depend on the independent variables. For systems with or without parameters, Ito's programs give the same results as our program. However, Ito's programs do not properly handle systems with parameters. The programs stop after generating the necessary conditions on the parameters, which must be analyzed separately. Such analyses revealed that the conditions do not always lead to an invariant or a symmetry. Indeed, in solving for the undetermined coefficients, Ito considers all possible branches in the solution, irrespective of whether or not these branches lead to an invariant or a symmetry.

In [14], Fuchssteiner *et al.* present an algorithm to compute higher-order symmetries of evolution equations. Their algorithm is based on Lie algebraic techniques and uses commutator algebra on the Lie algebra of vector fields. Their approach is different from the usual prolongation method in that no determining equations are solved. Instead, all necessary generators of the finitely generated Virasoro algebra are computed from one given element by direct Lie algebra methods. Their code is available in MuPAD.

The REDUCE program *FS* for "formal symmetries" was written by Gerdt and Zharkov [17]. The code *FS* can be applied to polynomial nonlinear PDEs of evolution type which are linear with respect to the highest-order spatial derivatives and with non-degenerated, diagonal coefficient matrix for the highest derivatives. *FS* computes higher-order symmetries and conservation laws of polynomial type. The algorithm in *FS* requires that the evolution equations are of order two or higher in the spatial variable. However, this approach does not require that the evolution equations are uniform in rank. With *FS* one cannot compute symmetries that depend explicitly on the independent variables. Applied to equations with parameters, *FS* computes the conditions on the parameters using the symmetry approach.

The PC package *DELiA* for *Differential Equations with Lie Approach*, written in Turbo PASCAL by Bocharov [5] and co-workers, is a commercial computer algebra system for investigating differential equations using Lie's approach. The program deals with higher-order symmetries, conservation laws, integrability and equivalence problems. It has a special routine for systems of evolution equations. The program requires the presence of second or higher-order spatial derivative terms in all equations. For systems with parameters, *DELiA* does not automatically compute the invariants or symmetries corresponding to the (necessary) conditions on the parameters. One has to use *DELiA*'s integrability test first, which determines conditions based on the existence of formal symmetries. Since these integrability conditions are neither necessary nor sufficient for the existence of invariants, one must further analyze the conditions manually. Once the parameters are fixed, one can compute the invariants and symmetries.

Sanders and Wang [55, 56, 57, 58] have Maple and FORM software that aids in the computation of conservation laws and recursion operators. Their approach is more abstract and relies on the implicit function theorem. They use concepts from operator theory (kernels and images). In fact, they use an extension of the total derivative operator to a Heisenberg algebra which allows them to invert the total derivative on its image. In contrast to our algorithm, their code makes no assumptions about the form of the conservation law.

Wolf *et al.* [71] have a package in REDUCE for the computation of conservation laws. There is no limitation on the number of independent variables in his package. The approach uses Wolf's program *CRACK* for solving overdetermined systems of PDEs. See [26] for a review of *CRACK* and for additional references to Wolf's work. Wolf's algorithm is particularly efficient for showing the non-existence of conservation laws of high order. In contrast to our program, it also allows one to compute non-polynomial conservation laws.

Finally, Hickman [29] at the University of Canterbury, Christchurch, New Zealand, has implemented a slight variation of our algorithm in Maple. Instead of computing the differential monomials in the invariant by repeated differentiation, Hickman uses a tree structure combining the appropriately weighted building blocks.

Chapter 12

CONCLUSIONS

We have presented *new* direct algorithms to compute polynomial higher-order symmetries, and conservation laws of polynomial systems of evolution and lattice equations. These algorithms are based on the dilation invariance of the given equations. For systems that arise from a variational principle, conservation laws follow from higher-order symmetries (Noether's theorem) and vice versa. Currently, in our algorithms we are not exploiting such connections.

Based on the knowledge of scaling properties, symmetries and conservation laws, we also presented a *new* algorithm to find recursion operators for systems of nonlinear PDEs.

The algorithms for symmetries and conservation laws are implemented in *Mathematica*. Hence, we offer the scientific community an integrated *Mathematica* package which is available in *MathSource* [21] to carry out the tedious calculations of generalized symmetries and conservation laws.

For PDEs and DDEs with parameters, the software automatically determines the conditions on these parameters so that a sequence of polynomial symmetries or invariants exists. The existence of a large number of symmetries and invariants is an indicator of integrability. Therefore, by generating the conditions, one can analyze classes of parameterized PDEs or DDEs, and filter out the candidates for complete integrability.

Furthermore, some of the results of this thesis have already been published [19, 20, 22, 24] in research journals, or under review [23].

In the future, we will investigate generalizations of our methods to PDEs and DDEs in multiple space dimensions. We will also study the potential use of Lie-point symmetries other than dilation symmetries. We hope to extend the algorithms to non-polynomial equations, symmetries, conservation laws, and equations with mixed derivatives. The definition of a conservation law for DDEs (hence the algorithm) is based on the forward difference discretization of the first derivative. Modifications for the central difference scheme will be investigated. The implementation of the algorithm for recursion operators of PDEs (and integrating it into the *Integrability Package*), and possible extensions towards finding the recursion operators of DDEs are also for future work.

REFERENCES

[1] M. J. Ablowitz and P. A. Clarkson, *Solitons, Nonlinear Evolution Equations and Inverse Scattering*, Cambridge University Press, Cambridge, 1991.

[2] M. J. Ablowitz and B. M. Herbst, On homoclinic structure and numerically induced chaos for the nonlinear Schrödinger equation, SIAM J. Appl. Math., 50 (1990) 339–351.

[3] M. J. Ablowitz and J. F. Ladik, Nonlinear differential-difference equations and Fourier analysis, J. Math. Phys., 17 (1976) 1011–1018.

[4] M. J. Ablowitz and J. F. Ladik, A nonlinear difference scheme and inverse scattering, Stud. Appl. Math., 55 (1976) 213–229.

[5] A. V. Bocharov, *DELiA: a system for exact analysis of Differential Equations using S. Lie Approach*, DELiA 1.5.1 User Guide, Beaver Soft Programming Team, New York, 1991.

[6] I. Cherdantsev and R. Yamilov, Local master symmetries of differential-difference equations in: Symmetries and integrability of difference equations, D. Levi, L. Vinet and P. Winternitz eds., American Mathematical Society, Providence, Rhode Island, 1996, pp. 51–61.

[7] B. Deconinck, A constructive test for integrability of semi-discrete systems, Phys. Lett. A, 223 (1996) 45–54.

[8] R. K. Dodd and J. D. Gibbon, The prolongation structure of a higher order Korteweg-de Vries equation, Proc. R. Soc. Lond. A, 358 (1977) 287–296.

[9] N. Euler and W. H. Steeb, *Continuous Symmetries, Lie Algebras and Differential Equations*, Wissenschaftsverlag, Mannheim, 1992.

[10] A. S. Fokas, A symmetry approach to exactly solvable evolution equations, J. Math. Phys., 21 (1980) 1318–1325.

[11] A. S. Fokas, Symmetries and integrability, Stud. Appl. Math., 77 (1987) 253–299.

[12] A. Fordy and J. Gibbons, Some remarkable nonlinear transformations, Phys. Lett. A, 75 (1980) 325.

[13] J. Frank, W. Huang and B. Leimkuhler, Geometric integrators for classical spin systems, J. Comp. Phys., 133 (1997) 160–172.

[14] B. Fuchssteiner, S. Ivanov and W. Wiwianka, Algorithmic determination of infinite-dimensional symmetry groups for integrable systems in 1+1 dimensions, Mathl. Comput. Modelling, 25 (1997) 91–100.

[15] B. Fuchssteiner, W. Oevel and W. Wiwianka, Computer-algebra methods for investigation of hereditary operators of higher order soliton equations, Comput. Phys. Commun., 44 (1987) 47–55.

[16] V. P. Gerdt, Computer algebra, symmetry analysis and integrability of nonlinear evolution equations, Int. J. Mod. Phys. C, 4 (1993) 279–286.

[17] V. P. Gerdt and A. Y. Zharkov, Computer generation of necessary integrability conditions for polynomial-nonlinear evolution systems, in *Proc. ISSAC '90*, S. Watenabe and M. Nagata eds., Academic Press, New York, 1990, pp. 250–254.

[18] Ü. Göktaş, Symbolic computation of conserved densities for systems of nonlinear evolution equations, MSc. Thesis, Colorado School of Mines, Colden, CO, 1996.

[19] Ü. Göktaş and W. Hereman, Symbolic computation of conserved densities for systems of nonlinear evolution equations, J. Symbolic Computation, 24 (1997) 591–621.

[20] Ü. Göktaş and W. Hereman, Computation of conservation laws for nonlinear lattices, Physica D (1998) in press.

[21] Ü. Göktaş and W. Hereman, The software package and the related files are available at http://www.mathsource.com/cgi-bin/msitem?0208-932. *MathSource* is a vast electronic library of *Mathematica* material.

[22] Ü. Göktaş and W. Hereman, Invariants and symmetries for partial differential equations and lattices, in: *Proc. Fourth International Conference on Mathematical and Numerical Aspects of Wave Propagation*, Ed.: J. A. DeSanto, SIAM, Philadelphia, 1998, pp. 403–407.

[23] Ü. Göktaş and W. Hereman, Computation of higher-order symmetries for nonlinear evolution and lattice equations, Adv. in Comp. Math. (1998) submitted.

[24] Ü. Göktaş, W. Hereman and G. Erdmann, Computation of conserved densities for systems of nonlinear differential-difference equations, Phys. Lett. A, 236 (1997) 30–38.

[25] M. Hénon, Integrals of the Toda lattice, Phys. Rev. B, 9 (1974) 1921–1923.

[26] W. Hereman, Symbolic software for Lie symmetry analysis, in: CRC Handbook of Lie Group Analysis of Differential Equations, Volume 3: New Trends in Theoretical Developments and Computational Methods, Chapter 13, Ed.: N.H. Ibragimov, CRC Press, Boca Raton, Florida (1996) 367-413.

[27] W. Hereman, Review of symbolic software for Lie symmetry analysis, Mathl. Comput. Modelling, 25 (1997) 115-132.

[28] F. J. Hickernell, The evolution of large-horizontal-scale disturbances in marginally stable, inviscid, shear flows, Stud. Appl. Math., 69 (1983) 23–49.

[29] M. Hickman, Private communication, 1998.

[30] R. Hirota and M. Ito, Resonance of solitons in one dimension, J. Phys. Soc. Jpn., 52 (1983) 744–748.

[31] R. Hirota and J. Satsuma, Soliton solutions of a coupled Korteweg-de Vries equation, Phys. Lett. A, 85 (1981) 407–408.

[32] E. G. B. Hohler and K. Olaussen, Using conservation laws to solve Toda field theories, Int. J. Mod. Phys. A, 11 (1996) 1831–1853.

[33] X-B. Hu and R. K. Bullough, Bäcklund transformation and nonlinear superposition formula of an extended Lotka-Volterra equation, J. Phys. A: Math. Gen., 30 (1997) 3635–3641.

[34] M. Ito, A REDUCE program for finding symmetries of nonlinear evolution equations with uniform rank, Comput. Phys. Commun., 42 (1986) 351–357.

[35] M. Ito, SYMCD - a REDUCE package for finding symmetries and conserved densities of systems of nonlinear evolution equations, Comput. Phys. Commun., 79 (1994) 547–554.

[36] M. Ito and F. Kako, A REDUCE program for finding conserved densities of partial differential equations with uniform rank, Comput. Phys. Commun., 38 (1985) 415–419.

[37] Y. Itoh, Integrals of a Lotka-Volterra system of odd number of variables, Prog. Theor. Phys., 78 (1987) 507–510.

[38] M. Kac and P. van Moerbeke, On an explicitly soluble system of nonlinear differential equations related to certain Toda lattices, Adv. Math., 16 (1975) 160–169.

[39] D. J. Kaup, On the inverse scattering problem for cubic eigenvalue problems of the class $\Psi_{xxx} + 6Q\Psi_x + 6R\Psi = \lambda\Psi$, Stud. Appl. Math., 62 (1980) 189–216.

[40] B. A. Kupershmidt, Mathematics of dispersive wave equations, Commun. Math Phys., 99 (1985) 51–73.

[41] B. A. Kupershmidt and G. Wilson, Modifying Lax equations and the second Hamiltonian structure, Invent. Math., 62 (1981) 403–436.

[42] P. D. Lax, Integrals of nonlinear equations of evolution and solitary waves, Commun. Pure Appl. Math., 21 (1968) 467–490.

[43] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Lec. in Math., Birkhäuser Verlag, Basel, 1992.

[44] D. Levi and O. Ragnisco, Extension of the spectral transform method for solving nonlinear differential-difference equations, Lett. Nuovo Cimento, 22 (1978) 691–696.

[45] D. Levi and P. Winternitz, Symmetries and conditional symmetries of differential-difference equations, J. Math. Phys., 34 (1993) 3713–3730.

[46] D. Levi and R. Yamilov, Conditions for the existence of higher symmetries of evolutionary equations on the lattice, J. Math. Phys., 38 (1997) 6648–6674.

[47] A. V. Mikhailov, A. B. Shabat and V. V. Sokolov, The symmetry approach to classification of integrable equations, in *What Is Integrability?*, V. E. Zakharov ed., Springer-Verlag, Berlin Heidelberg, 1991, pp. 115–184.

[48] R. M. Miura, Korteweg-de Vries equation and generalizations. I. A remarkable explicit nonlinear transformation, J. Math. Phys., 9 (1968) 1202–1204.

[49] A. C. Newell, The history of the soliton, J. Appl. Mech., 50 (1983) 1127–1137.

[50] J. M. Nunes da Costa and C.-M. Marle, Master symmetries and bi-Hamiltonian structures for the relativistic Toda lattice, J. Phys. A: Math. Gen., 30 (1997) 7551–7556.

[51] P. J. Olver, Evolution equations possessing infinitely many symmetries, J. Math. Phys., 18 (1977) 1212–1215.

[52] P. J. Olver, *Applications of Lie Groups to Differential Equations*, 2nd Edition, Springer Verlag, New York, 1993.

[53] A. Ramani, B. Grammaticos and K. M. Tamizhmani, An integrability test for differential-difference systems, J. Phys. A: Math. Gen., 25 (1992) L883–L886.

[54] V. Rosenhaus and G. H. Katzin, On symmetries, conservation laws, and variational problems for partial differential equations, J. Math. Phys., 35 (1994) 1998–2012.

[55] J. A. Sanders and J. P. Wang, Hodge decomposition and conservation laws, Math. Comp. in Simulation, 44 (1997) 483–493.

[56] J. A. Sanders and J. P. Wang, Classification of conservation laws for KdV-like equations, Math. Comp. in Simulation, 44 (1997) 471–481.

[57] J. Sanders and J. P. Wang, On hereditary recursion operators, Report WS-472, Department of Mathematics and Computer Sciences, Free University, Amsterdam, The Netherlands, 1997.

[58] J. Sanders and J. P. Wang, On the integrability of homogeneous scalar evolution equations, J. Differential Equations, (1998) to appear.

[59] J. M. Sanz-Serna, An explicit finite-difference scheme with exact conservation laws, J. Comput. Phys., 47 (1982) 199–210.

[60] J. Satsuma and D. J. Kaup, A Bäcklund transformation for a higher order Korteweg-De Vries equation, J. Phys. Soc. Jpn., 43 (1977) 692–697.

[61] K. Sawada and T. Kotera, A method for finding N-Soliton solutions of the K.d.V. equation and K.d.V.-like equation, Prog. Theor. Phys., 51 (1974) 1355–1367.

[62] A. B. Shabat and R. I. Yamilov, Symmetries of nonlinear chains, Leningrad Math. J., 2 (1991) 377–400.

[63] V. V. Sokolov and A. B. Shabat, Classification of integrable evolution equations, in: Soviet Sci. Rev. Sec. C, Vol. 4, Harwood Academic Publishers, New York, 1984, 221–280.

[64] Y. B. Suris, New integrable systems related to the relativistic Toda lattice, J. Phys. A: Math. Gen., 30 (1997) 1745–1761.

[65] Y. B. Suris, On some integrable systems related to the Toda lattice, J. Phys. A: Math. Gen., 30 (1997) 2235–2249.

[66] T. R. Taha, A differential-difference equation for a KdV-MKdV equation, Maths. Comput. in Simul., 35 (1993) 509–512.

[67] M. Toda, *Theory of Nonlinear Lattices*, Springer Verlag, Berlin, 1981.

[68] F. Verheest, Integrability, invariants and bi-Hamiltonian structure of vector nonlinear evolution equations, in: *Proc. Fourth International Conference on Mathematical and Numerical Aspects of Wave Propagation*, Ed.: J. A. DeSanto, SIAM, Philadelphia, 1998, pp. 398–402.

[69] M. Wadati, Transformation theories for nonlinear discrete systems, Prog. Theor. Phys. Suppl., 59 (1976) 36–63.

[70] G. B. Whitham, *Linear and Nonlinear Waves*, Wiley & Sons, Inc., New York, 1974.

[71] T. Wolf, A. Brand and M. Mohammadzadeh, Computer algebra algorithms and routines for the computation of conservation laws and fixing of gauge in differential expressions, J. Symbolic Computation (1998) submitted.

[72] S. Wolfram, *The Mathematica Book*, 3rd Edition, Wolfram Media, Urbana-Champaign, Illinois & Cambridge University Press, London, 1996.

[73] R. Yamilov, Classification of Toda type scalar lattices, in: Proc. 8th int. workshop on nonlinear evolution equations and dynamical systems, World Scientific Publishing, Singapore, 1993, 423–431

[74] V. E. Zakharov ed., *What is Integrability?*, Springer-Verlag, Berlin Heidelberg, 1990.