Symbolic Computation of Conservation Laws, Generalized Symmetries, and Recursion Operators for Nonlinear Differential-Difference Equations ⋆

Ünal Göktaş * **Willy Hereman** **

* *Department of Computer Engineering, Turgut Özal University, Keçiören, Ankara 06010, Turkey (e-mail: unalgoktas@ttmail.com)*
** *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado 80401-1887, U.S.A. (e-mail: whereman@mines.edu)*

Abstract: Algorithms for the symbolic computation of polynomial conservation laws, generalized symmetries, and recursion operators for systems of nonlinear differential-difference equations (DDEs) are presented. The algorithms can be used to test the complete integrability of nonlinear DDEs. The ubiquitous Toda lattice illustrates the steps of the algorithms, which have been implemented in *Mathematica*. The codes InvariantsSymmetries.m and DDERecursionOperator.m can aid researchers interested in properties of nonlinear DDEs.

Keywords: generalized symmetry, conservation law, recursion operator, complete integrability, differential-difference equation.

1. INTRODUCTION

A large number of physically important nonlinear models are completely integrable, i.e. they can be linearized via an explicit transformation or can be solved with the Inverse Scattering Transform. Completely integrable continuous and discrete models arise in many branches of the applied sciences and engineering, including classical, quantum, and plasma physics, optics, electrical circuits, to name a few. Mathematically, nonlinear models can be represented by ordinary and partial differential equations (ODEs and PDEs), differential-difference equations (DDEs), or ordinary and partial difference equations (OΔEs and PΔEs). This paper deals with integrable nonlinear DDEs.

Completely integrable equations have remarkable analytic and geometric properties reflecting their rich mathematical structure. For instance, completely integrable PDEs and DDEs possess infinitely many conserved quantities and generalized (higher-order) symmetries of successive orders. The existence of an infinite set of generalized symmetries can be established by explicitly constructing recursion operators which connect such symmetries. Finding generalized symmetries and recursion operators is a nontrivial task, in particular, if attempted by hand. For example, Göktaş (1998) and Hereman and Göktaş (1999) present an algorithm to compute recursion operators for completely integrable PDEs, which was only recently implemented in *Mathematica* by Baldwin and Hereman (2010).

Based on earlier work by Göktaş (1998) and Göktaş and Hereman (1998,1999), we present in this paper algorithms for the symbolic computation of conserved densities, generalized symmetries, and recursion operators of nonlinear systems of DDEs. Such systems must be polynomial and of evolution type, i.e. the DDEs must be of first order in (continuous) time. The number of equations in the system, degree of nonlinearity, and order (shift levels) are arbitrary. Furthermore, the current algorithms only cover *polynomial* densities, symmetries, and recursion operators.

We use the dilation (scaling) invariance of the system of DDEs to determine the *candidate* density, symmetry, or recursion operator. Indeed, these candidates are linear combinations with undetermined coefficients of scaling invariant terms. Upon substitution of the candidates into the corresponding defining equations, one has to solve a linear system for the undetermined coefficients. After doing so, the coefficients are substituted into the density, symmetry, or recursion operator. If so desired, the results can be tested one more time, in particular, by applying the recursion operators to generate the successive symmetries.

If the system of DDEs contains constant parameters, the eliminant of the linear system for the undetermined coefficients gives the necessary conditions for the parameters, so that the given DDEs admit the required density or symmetry. In analogy with the PDE case in Göktaş and Hereman (1997), the algorithms can thus be used to classify DDEs with parameters according to their complete integrability as illustrated by Göktaş and Hereman (1998,1999).

As shown by Fokas (1980), once the generalized symmetries are explicitly known, it is often possible to find the recursion operator by inspection. If the recursion operator is *hereditary*, as defined by Fuchssteiner et al. (1987), then the equation will possess infinitely many symmetries.

⋆ This material is based upon work supported by the National Science Foundation (U.S.A.)under Grant No. CCF-0830783.

If, in addition, the recursion operator is *factorizable* then the equation has infinitely many conserved quantities.

Computer algebra systems can greatly help with the search for conservation laws, generalized symmetries, and recursion operators. The algorithms in this paper have been implemented in *Mathematica*. The computer codes (see Hereman (2010)), can be used to test the complete integrability of systems of nonlinear DDEs, provided they are polynomial and of first order (or can be written in that form after a suitable transformation).

With InvariantsSymmetries.m, Göktaş (1998) and Göktaş and Hereman (1998,1999) computed polynomial conserved densities and generalized symmetries of many well-known systems of DDEs. The existence of, say, a half dozen conserved densities or generalized symmetries is a *predictor* for complete integrability. Finding a recursion operator then becomes within reach. An existence proof (showing that there are indeed infinitely many densities or generalized symmetries) must be done analytically, e.g., by explicitly constructing the recursion operator which allows one to generate the generalized symmetries order by order. Numerous explicit examples have been reported in the literature but novices could start with the book by Olver (1993) to learn about recursion operators for PDEs. To alleviate the burden of trying to find a recursion operator by trial and error, we present a new *Mathematica* program, DDERecursionOperator.m, based on the algorithm in Section 5. Like InvariantsSymmetries.m, after thorough testing, DDERecursionOperator.m will be available from Hereman (2010).

If one cannot find a sufficient large number of densities or symmetries (let alone, a recursion operator), then it is unlikely that the DDE system is completely integrable, at least in that coordinate representation. However, our software does not allow one to conclude that a DDE is *not* completely integrable merely based on the fact that polynomial conserved densities and generalized symmetries could not be found. Polynomial DDEs that lack the latter may accidentally have non-polynomial densities or symmetries, or a complicated recursion operator, which is outside the scope of the algorithm described in Section 5.

The paper is organized as follows. Basic definitions are given in Section 2. In Section 3, we show the algorithm for conservation laws, using the Toda lattice as an example. Using the same example, Sections 4 and 5 cover the algorithms for generalized symmetries and recursion operators, respectively.

2. KEY DEFINITIONS

Consider a system of nonlinear DDEs of first order,

$$\dot{\mathbf{u}}_n = \mathbf{F}(\mathbf{u}_{n-\ell}, ..., \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, ..., \mathbf{u}_{n+m}), \qquad (1)$$

where $\mathbf{u}_n$ and $\mathbf{F}$ are vector-valued functions with N components. This paper only covers DDEs with *one* discrete variable, denoted by integer n, which often corresponds to the discretization of a space variable. The dot stands for differentiation with respect to the continuous variable (often time t). Each component of $\mathbf{F}$ is assumed to be a polynomial with constant coefficients. If parameters are present in (1), they will be denoted by lower-case Greek letters. $\mathbf{F}$ depends on $\mathbf{u}_n$ and a finite number of forward and backward shifts of $\mathbf{u}_n$. We denote by ℓ (m, respectively), the furthest negative (positive, respectively) shift of any variable in the system. Restrictions are neither imposed on the degree of nonlinearity of $\mathbf{F}$, nor on the integers l and m, which measure the degree of non-locality in (1).

2.1 Leading Example: The Toda Lattice

One of the earliest and most famous examples of completely integrable DDEs is the Toda lattice, discussed in, for instance, Toda (1981):

$$\ddot{y}_n = \exp(y_{n-1} - y_n) - \exp(y_n - y_{n+1}), \qquad (2)$$

where y_n is the displacement from equilibrium of the nth particle with unit mass under an exponential decaying interaction force between nearest neighbors. In new variables (u_n, v_n), defined by $u_n = \dot{y}_n, v_n = \exp(y_n - y_{n+1})$, lattice (2) can be written in polynomial form

$$\dot{u}_n = v_{n-1} - v_n, \quad \dot{v}_n = v_n(u_n - u_{n+1}). \qquad (3)$$

The Toda lattice (3) will be used to illustrate the various algorithms presented in subsequent sections of this paper.

2.2 Dilation Invariance

A DDE is *dilation invariant* if it is invariant under a dilation (scaling) symmetry.

Example Lattice (3) is invariant under scaling symmetry

$$(t, u_n, v_n) \to (\lambda^{-1}t, \lambda^1 u_n, \lambda^2 v_n). \qquad (4)$$

2.3 Uniformity in Rank

We define the *weight*, w, of a variable as the exponent of the scaling parameter (λ) which multiplies that variable. Since λ can be selected at will, t will always be replaced by $\frac{t}{\lambda}$ and, thus, $w(\frac{\mathrm{d}}{\mathrm{dt}}) = w(\mathrm{D}_t) = 1$.

Weights of dependent variables are nonnegative, rational, and independent of n. For example, $w(u_{n-3}) = \cdots = w(u_n) = \cdots = w(u_{n+2})$.

The *rank*, denoted by R, of a monomial is defined as the total weight of the monomial. An expression is *uniform in rank* if all of its terms have the same rank.

Dilation symmetries, which are special Lie-point symmetries, are common to many DDEs. Polynomial DDEs that do not admit a dilation symmetry can be made scaling invariant by extending the set of dependent variables with auxiliary parameters with appropriate scales as discussed by Göktaş and Hereman (1998,1999).

Example In view of (4), we have $w(u_n) = 1$, and $w(v_n) = 2$ for the Toda lattice.

In the first equation of (3), all the monomials have rank 2; in the second equation all the monomials have rank 3. Conversely, requiring uniformity in rank for each equation in (3) allows one to compute the weights of the dependent variables (and, thus, the scaling symmetry) with simple linear algebra. Balancing the weights of the various terms,

$$w(u_n) + 1 = w(v_n), \quad w(v_n) + 1 = w(u_n) + w(v_n), \qquad (5)$$

yields

$$w(u_n) = 1, \quad w(v_n) = 2, \qquad (6)$$

which confirms (4).

2.4 Up-Shift and Down-Shift Operator

We define the shift operator D by $\mathsf{D}\mathbf{u}_n = \mathbf{u}_{n+1}$. The operator D is often called the *up-shift operator* or forward- or right-shift operator. The inverse, D^{-1}, is the *down-shift operator* or backward- or left-shift operator, $\mathsf{D}^{-1}\mathbf{u}_n = \mathbf{u}_{n-1}$. Shift operators apply to functions by their action on the arguments of the functions. For example,

$$\begin{aligned}&\mathsf{D}\mathbf{F}(\mathbf{u}_{n-\ell},\cdots,\mathbf{u}_{n-1},\mathbf{u}_n,\mathbf{u}_{n+1},\cdots,\mathbf{u}_{n+m})\\&\quad=\mathbf{F}(\mathsf{D}\mathbf{u}_{n-\ell},\cdots,\mathsf{D}\mathbf{u}_{n-1},\mathsf{D}\mathbf{u}_n,\mathsf{D}\mathbf{u}_{n+1},\ldots,\mathsf{D}\mathbf{u}_{n+m})\\&\quad=\mathbf{F}(\mathbf{u}_{n-\ell+1},\ldots,\mathbf{u}_n,\mathbf{u}_{n+1},\mathbf{u}_{n+2},\cdots,\mathbf{u}_{n+m+1}).\end{aligned}\tag{7}$$

2.5 Conservation Law

A *conservation law* of (1),

$$\mathsf{D}_t\,\rho+\Delta\,J=0,\tag{8}$$

connects a *conserved density* ρ to an *associated flux* J, where both are scalar functions depending on $\mathbf{u}_n$ and its shifts. In (8), which *must holds* on solutions of (1), D_t is the total derivative with respect to time, $\Delta = \mathsf{D} - \mathsf{I}$ is the *forward difference operator*, and I is the identity operator. For readability (in particular, in the examples), the components of $\mathbf{u}_n$ will be denoted by u_n, v_n, w_n, etc. In what follows we consider only autonomous functions, i.e. $\mathbf{F}, \rho$, and J do not explicitly depend on t and n.

A density is *trivial* if there exists a function ψ so that $\rho = \Delta\psi$. We say that two densities, $\rho^{(1)}$ and $\rho^{(2)}$, are *equivalent* if and only if $\rho^{(1)} + k\rho^{(2)} = \Delta\psi$, for some ψ and some non-zero scalar k. It is paramount that the density is free of equivalent terms for if such terms were present, they could be moved into the flux J.

Compositions of D or D^{-1} define an *equivalence relation* ($\equiv$) on monomial terms. Simply stated, all shifted terms are *equivalent*, e.g., $u_{n-1}v_{n+1} \equiv u_nv_{n+2} \equiv u_{n+2}v_{n+4} \equiv u_{n-3}v_{n-1}$ since

$$\begin{aligned}u_{n-1}v_{n+1}&=u_nv_{n+2}-\Delta(u_{n-1}v_{n+1})\\&=u_{n+2}v_{n+4}-\Delta(u_{n+1}v_{n+3}+u_nv_{n+2}+u_{n-1}v_{n+1})\\&=u_{n-3}v_{n-1}+\Delta(u_{n-2}v_n+u_{n-3}v_{n-1}).\end{aligned}\tag{9}$$

This equivalence relation also holds for any function of the dependent variables, but for the construction of conserved densities we will apply it only to monomial terms (t_i) in the *same* density, thereby achieving high computational efficiency. In the algorithm used in Section 3, we will use the following *equivalence criterion*: two monomial terms, t_1 and t_2, are equivalent, $t_1 \equiv t_2$, if and only if $t_1 = \mathsf{D}^r\,t_2$ for some integer r. If $t_1 \equiv t_2$ then $t_1 = t_2 + \Delta J$ for some J dependent on $\mathbf{u}_n$ and its shifts. For example, $u_{n-2}u_n \equiv u_{n-1}u_{n+1}$ because $u_{n-2}u_n = \mathsf{D}^{-1}u_{n-1}u_{n+1}$. Hence, $u_{n-2}u_n = u_{n-1}u_{n+1} + [-u_{n-1}u_{n+1} + u_{n-2}u_n] = u_{n-1}u_{n+1} + \Delta J$ with $J = -u_{n-2}u_n$.

For efficiency, we need a criterion to choose a unique representative from each equivalence class. There are a number of ways to do this. We define the *canonical* representative as that member that has (i) no negative shifts and (ii) a non-trivial dependence on the *local* (that is, zero-shifted) variable. For example, u_nu_{n+2} is the canonical representative of the class

$$\{\cdots, u_{n-2}u_n, u_{n-1}u_{n+1}, u_nu_{n+2}, u_{n+1}u_{n+3}, \cdots\}.$$

In the case of, e.g., two variables (u_n and v_n), $u_{n+2}v_n$ is the canonical representative of the class

$$\{\cdots, u_{n-1}v_{n-3}, u_nv_{n-2}, u_{n+1}v_{n-1}, u_{n+2}v_n, u_{n+3}v_{n+1}, \cdots\}.$$

Alternatively, one could choose a variable ordering and then choose the member that depends on the zero-shifted variable of lowest lexicographical order. The code in Hereman (2010) uses lexicographical ordering of the variables, i.e. $u_n \prec v_n \prec w_n$, etc. Thus, u_nv_{n-2} (instead of $u_{n+2}v_n$) is chosen as the canonical representative of $\{\cdots, u_{n-1}v_{n-3}, u_nv_{n-2}, u_{n+1}v_{n-1}, u_{n+2}v_n, u_{n+3}v_{n+1}, \cdots\}$.

It was shown by Hickman (2008) that if ρ is a density then $\mathsf{D}^k\rho$ is also a density. Hence, using an appropriate "up-shift" all negative shifts in a density can be removed. Without loss of generality, we thus assume that a density that depends on q shifts has *canonical* form $\rho(\mathbf{u}_n, \mathbf{u}_{n+1}, \cdots, \mathbf{u}_{n+q})$.

Example Lattice (3) has infinitely many conservation laws (see, e.g., Hénon (1974)). Here we list the densities of rank $R \leq 4$:

$$\rho^{(1)} = u_n,\tag{10}$$

$$\rho^{(2)} = \tfrac{1}{2}u_n^2 + v_n,\tag{11}$$

$$\rho^{(3)} = \tfrac{1}{3}u_n^3 + u_n(v_{n-1} + v_n),\tag{12}$$

$$\begin{aligned}\rho^{(4)} = {}&\tfrac{1}{4}u_n^4 + u_n^2(v_{n-1} + v_n) + u_nu_{n+1}v_n\\&+\tfrac{1}{2}v_n^2 + v_nv_{n+1}.\end{aligned}\tag{13}$$

The first two density-flux pairs are easily computed by hand, and so is

$$\rho_n^{(0)} = \ln(v_n),\tag{14}$$

which is the only non-polynomial density (of rank 0).

2.6 Generalized Symmetry

A vector function $\mathbf{G}(\mathbf{u}_n)$ is called a *generalized symmetry* of (1) if the infinitesimal transformation $\mathbf{u}_n \to \mathbf{u}_n + \epsilon\mathbf{G}$ leaves (1) invariant up to order ϵ. As shown by Olver (1993), $\mathbf{G}$ must then satisfy

$$\mathrm{D}_t\mathbf{G} = \mathbf{F}'(\mathbf{u}_n)[\mathbf{G}]\tag{15}$$

on solutions of (1), where $\mathbf{F}'(\mathbf{u}_n)[\mathbf{G}]$ is the Fréchet derivative of $\mathbf{F}$ in the direction of $\mathbf{G}$.

For the scalar case ($N = 1$), the Fréchet derivative is

$$F'(u_n)[G] = \frac{\partial}{\partial\epsilon}F(u_n + \epsilon G)|_{\epsilon=0} = \sum_k \frac{\partial F}{\partial u_{n+k}}\mathrm{D}^kG,\tag{16}$$

which, in turn, defines the Fréchet derivative operator

$$F'(u_n) = \sum_k \frac{\partial F}{\partial u_{n+k}}\mathrm{D}^k.\tag{17}$$

In the vector case with, say, components u_n and v_n, the Fréchet derivative operator is a matrix operator:

$$\mathbf{F}'(\mathbf{u}_n) = \begin{pmatrix}\displaystyle\sum_k \frac{\partial F_1}{\partial u_{n+k}}\mathrm{D}^k & \displaystyle\sum_k \frac{\partial F_1}{\partial v_{n+k}}\mathrm{D}^k\\[2ex]\displaystyle\sum_k \frac{\partial F_2}{\partial u_{n+k}}\mathrm{D}^k & \displaystyle\sum_k \frac{\partial F_2}{\partial v_{n+k}}\mathrm{D}^k\end{pmatrix}.\tag{18}$$

Applied to $\mathbf{G} = (G_1\ G_2)^{\mathrm{T}}$, where T is transpose, one obtains

$$F_i{}'(\mathbf{u}_n)[\mathbf{G}] = \sum_k \frac{\partial F_i}{\partial u_{n+k}} \mathrm{D}^k G_1 + \sum_k \frac{\partial F_i}{\partial v_{n+k}} \mathrm{D}^k G_2, \quad (19)$$

with $i = 1, 2$. In (16) and (19) summation is over all positive and negative shifts (including $k = 0$). The generalization of (18) to a $N-$component system is straightforward.

Example As computed by Hereman et al. (1998), the first two non-trivial symmetries of (3) are

$$\mathbf{G}^{(1)} = \begin{pmatrix} v_n - v_{n-1} \\ v_n(u_{n+1} - u_n) \end{pmatrix}, \quad (20)$$

$$\mathbf{G}^{(2)} = \begin{pmatrix} v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} + u_n) \\ v_n(u_{n+1}^2 - u_n^2 + v_{n+1} - v_{n-1}) \end{pmatrix}. \quad (21)$$

2.7 Recursion Operator

A *recursion operator* $\mathcal{R}$ connects symmetries

$$\mathbf{G}^{(j+s)} = \mathcal{R}\mathbf{G}^{(j)}, \quad (22)$$

where $j = 1, 2, \cdots$, and s is the gap length. The symmetries are linked consecutively if $s = 1$. This happens in most (but not all) cases. For N-component systems, $\mathcal{R}$ is an $N \times N$ matrix operator.

With reference to Olver (1993) and Wang (1998), the defining equation for $\mathcal{R}$ is

$$\begin{aligned} &\mathrm{D}_t\mathcal{R} + [\mathcal{R}, \mathbf{F}'(\mathbf{u}_n)] \\ &= \frac{\partial \mathcal{R}}{\partial t} + \mathcal{R}'[\mathbf{F}] + \mathcal{R} \circ \mathbf{F}'(\mathbf{u}_n) - \mathbf{F}'(\mathbf{u}_n) \circ \mathcal{R} = 0, \end{aligned} \quad (23)$$

where $[\ ,\ ]$ denotes the commutator and $\circ$ the composition of operators. The operator $\mathbf{F}'(\mathbf{u}_n)$ was defined in (18). $\mathcal{R}'[\mathbf{F}]$ is the Fréchet derivative of $\mathcal{R}$ in the direction of $\mathbf{F}$. For the scalar case, the operator $\mathcal{R}$ is often of the form

$$\mathcal{R} = U(u_n)\, \mathcal{O}((\mathrm{D} - \mathrm{I})^{-1}, \mathrm{D}^{-1}, \mathrm{I}, \mathrm{D})\, V(u_n), \quad (24)$$

and then

$$\mathcal{R}'[F] = \sum_k (\mathrm{D}^k F) \frac{\partial U}{\partial u_{n+k}} \mathcal{O} V + \sum_k U \mathcal{O} (\mathrm{D}^k F) \frac{\partial V}{\partial u_{n+k}}. \quad (25)$$

For the vector case, the elements of the $N \times N$ operator matrix $\mathcal{R}$ are often of the form

$$\mathcal{R}_{ij} = U_{ij}(\mathbf{u}_n)\, \mathcal{O}_{ij}((\mathrm{D} - \mathrm{I})^{-1}, \mathrm{D}^{-1}, \mathrm{I}, \mathrm{D})\, V_{ij}(\mathbf{u}_n). \quad (26)$$

Hence, for the 2-component case

$$\begin{aligned} \mathcal{R}'[\mathbf{F}]_{ij} = &\sum_k (\mathrm{D}^k F_1) \frac{\partial U_{ij}}{\partial u_{n+k}} \mathcal{O}_{ij}\, V_{ij} \\ &+ \sum_k (\mathrm{D}^k F_2) \frac{\partial U_{ij}}{\partial v_{n+k}} \mathcal{O}_{ij}\, V_{ij} \\ &+ \sum_k U_{ij} \mathcal{O}_{ij} (\mathrm{D}^k F_1) \frac{\partial V_{ij}}{\partial u_{n+k}} \\ &+ \sum_k U_{ij} \mathcal{O}_{ij} (\mathrm{D}^k F_2) \frac{\partial V_{ij}}{\partial v_{n+k}}. \end{aligned} \quad (27)$$

Example The recursion operator of (3) is

$$\mathcal{R} = \begin{pmatrix} u_n \mathrm{I} & \mathrm{D}^{-1} + \mathrm{I} + (v_n - v_{n-1})(\mathrm{D} - \mathrm{I})^{-1} \dfrac{1}{v_n} \mathrm{I} \\ v_n \mathrm{I} + v_n \mathrm{D} & u_{n+1}\mathrm{I} + v_n(u_{n+1} - u_n)(\mathrm{D} - \mathrm{I})^{-1} \dfrac{1}{v_n} \mathrm{I} \end{pmatrix}. \quad (28)$$

It is straightforward to verify that $\mathcal{R}G^{(1)} = G^{(2)}$ with $G^{(1)}$ in (20) and $G^{(2)}$ in (21).

3. ALGORITHM FOR CONSERVATION LAWS

As an example, we will compute the density $\rho^{(3)}$ (of rank $R = 3$) given in (12).

3.1 Construct the Form of the Density

Start from $\mathcal{V} = \{u_n, v_n\}$, the set of dependent variables with weights. List all monomials in u and v of rank $R = 3$ or less: $\mathcal{M} = \{u_n^3, u_n^2, u_n v_n, u_n, v_n\}$. Next, for each monomial in $\mathcal{M}$, introduce the correct number of t-derivatives so that each term has rank 3. Using (3), compute

$$\begin{aligned} &\frac{\mathrm{d}^0 u_n^3}{\mathrm{d}t^0} = u_n^3, \qquad \frac{\mathrm{d}^0 u_n v_n}{\mathrm{d}t^0} = u_n v_n, \\ &\frac{\mathrm{d}u_n^2}{\mathrm{d}t} = 2u_n \dot{u}_n = 2u_n v_{n-1} - 2u_n v_n, \\ &\frac{\mathrm{d}v_n}{\mathrm{d}t} = \dot{v}_n = u_n v_n - u_{n+1} v_n, \\ &\frac{\mathrm{d}^2 u_n}{\mathrm{d}t^2} = \frac{\mathrm{d}\dot{u}_n}{\mathrm{d}t} = \frac{\mathrm{d}(v_{n-1} - v_n)}{\mathrm{d}t} \\ &\qquad = u_{n-1}v_{n-1} - u_n v_{n-1} - u_n v_n + u_{n+1} v_n. \end{aligned} \quad (29)$$

Gather the terms in the right hand sides in (29) to get $\mathcal{R} = \{u_n^3, u_n v_{n-1}, u_n v_n, u_{n-1} v_{n-1}, u_{n+1} v_n\}$.

Identify members belonging to the same equivalence classes and replace them by their canonical representatives. For example, $u_n v_{n-1} \equiv u_{n+1} v_n$. Adhering to lexicographical ordering, use $u_n v_{n-1}$ instead of $u_{n+1} v_n$. Doing so, replace $\mathcal{R}$ by $\mathcal{S} = \{u_n^3, u_n v_{n-1}, u_n v_n\}$, which has the building blocks of the density. Linearly combine the monomials in $\mathcal{S}$ with undetermined coefficients c_i to get the candidate density of rank 3 :

$$\rho = c_1\, u_n^3 + c_2\, u_n v_{n-1} + c_3\, u_n v_n. \quad (30)$$

3.2 Compute the Undetermined Coefficients c_i

Compute $\mathrm{D}_t \rho$ and use (3) to eliminate $\dot{u}_n$ and $\dot{v}_n$ and their shifts. Next, introduce the main representatives to get

$$\begin{aligned} E = &(3c_1 - c_2)u_n^2 v_{n-1} + (c_3 - 3c_1)u_n^2 v_n + (c_3 - c_2) v_n v_{n+1} \\ &+ (c_2 - c_3) u_n u_{n+1} v_n + (c_2 - c_3) v_n^2 + \Delta J, \end{aligned} \quad (31)$$

with

$$J = (c_3 - c_2) v_{n-1} v_n + c_2 u_{n-1} u_n v_{n-1} + c_2 v_{n-1}^2. \quad (32)$$

Set $E - \Delta J \equiv 0$ to get the linear system

$$3c_1 - c_2 = 0, \quad c_3 - 3c_1 = 0, \quad c_2 - c_3 = 0. \quad (33)$$

Select $c_1 = \frac{1}{3}$ and substitute the solution $c_1 = \frac{1}{3}, c_2 = c_3 = 1$, into (30) and (32) to obtain $\rho^{(3)}$ in (12) with matching flux $J^{(3)} = u_{n-1} u_n v_{n-1} + v_{n-1}^2$.

4. ALGORITHM FOR SYMMETRIES

As an example, we will compute the symmetry $G^{(2)} = (G_1^{(2)}, G_2^{(2)})$ of rank $(3,4)$ given in (21).

4.1 Construct the Form of the Symmetry

Listing all monomials in u_n and v_n of ranks 3 and 4, or less:

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

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

$$\frac{\mathrm{d}^0}{\mathrm{d}t^0}(u_n^3) = u_n^3, \qquad \frac{\mathrm{d}^0}{\mathrm{d}t^0}(u_n v_n) = u_n v_n,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}(u_n^2) = 2u_n\dot{u}_n = 2u_n v_{n-1} - 2u_n v_n,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}(v_n) = \dot{v}_n = u_n v_n - u_{n+1} v_n,$$
$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}(u_n) = \frac{\mathrm{d}}{\mathrm{d}t}(\dot{u}_n) = \frac{\mathrm{d}}{\mathrm{d}t}(v_{n-1} - v_n)$$
$$= u_{n-1}v_{n-1} - u_n v_{n-1} - u_n v_n + u_{n+1} v_n.$$

Put the terms from the right hand sides into 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 $\mathcal{L}_2$, construct

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

Linearly combine the monomials in $\mathcal{R}_1$ and $\mathcal{R}_2$ with undetermined coefficients c_i to get the form of the components of the candidate symmetry:

$$G_1^{(2)} = c_1\, u_n^3 + c_2\, u_{n-1}v_{n-1} + c_3\, u_n v_{n-1} + c_4\, u_n v_n + c_5\, u_{n+1} v_n,$$
$$G_2^{(2)} = c_6\, u_n^4 + c_7\, u_{n-1}^2 v_{n-1} + c_8\, u_{n-1}u_n v_{n-1} + c_9\, u_n^2 v_{n-1} + c_{10}\, v_{n-2}v_{n-1} + c_{11}\, v_{n-1}^2 + c_{12}\, u_n^2 v_n + c_{13}\, u_n u_{n+1} v_n + c_{14}\, u_{n+1}^2 v_n + c_{15}\, v_{n-1}v_n + c_{16}\, v_n^2 + c_{17}\, v_n v_{n+1}. \quad (34)$$

4.2 Compute the Undetermined Coefficients c_i

To determine the coefficients c_i, require that (15) holds on any solution of (1). Compute $\mathrm{D}_t\mathbf{G}$ and use (1) to remove all $\dot{\mathbf{u}}_{n-1}, \dot{\mathbf{u}}_n, \dot{\mathbf{u}}_{n+1}$, etc. Compute the Fréchet derivative (19) and, in view of (15), equate the resulting expressions. Treat as independent all the monomials in $\mathbf{u}_n$ and their shifts, to obtain the linear system that determines the coefficients c_i.

Apply the strategy to (3) with (34), to get

$$c_1 = c_6 = c_7 = c_8 = c_9 = c_{10} = c_{11} = c_{13} = c_{16} = 0,$$
$$-c_2 = -c_3 = c_4 = c_5 = -c_{12} = c_{14} = -c_{15} = c_{17}.$$

Set $c_{17} = 1$ and substitute (35) into (34) to get $G^{(2)} = (G_1^{(2)}, G_2^{(2)})$, as given in (21).

To show how our algorithm filters out completely integrable cases among parameterized systems of DDEs, consider

$$\dot{u}_n = \alpha\, v_{n-1} - v_n, \quad \dot{v}_n = v_n\,(\beta\, u_n - u_{n+1}), \qquad (35)$$

where α and β are *nonzero* constant parameters. Ramani et al. (1992) have shown that (35) 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 (35) admits a polynomial symmetry, say, of rank $(3,4)$. The steps are as above, however, the linear system for the c_i is parameterized by α and β and must be analyzed carefully (with, e.g., Gröbner basis methods). This analysis leads to the condition $\alpha = \beta = 1$. Details are given in Göktaş and Hereman (1998,1999).

5. ALGORITHM FOR RECURSION OPERATORS

We will now construct the recursion operator (28) for (3). In this case all the terms in (23) are 2×2 matrix operators.

5.1 Determine the Rank of the Recursion Operator

The difference in the ranks of symmetries is used to compute the rank of the elements of the recursion operator. Use (6), (20) and (21) to compute

$$\operatorname{rank}\mathbf{G}^{(1)} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad \operatorname{rank}\mathbf{G}^{(2)} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}. \qquad (36)$$

Assume that $\mathcal{R}\,\mathbf{G}^{(1)} = \mathbf{G}^{(2)}$ and use the formula

$$\operatorname{rank}\mathcal{R}_{ij} = \operatorname{rank}G_i^{(k+1)} - \operatorname{rank}G_j^{(k)}, \qquad (37)$$

to compute a rank matrix associated to the operator $\mathcal{R}$:

$$\operatorname{rank}\mathcal{R} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}. \qquad (38)$$

5.2 Determine the Form of the Recursion Operator

We assume that $\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1$, where $\mathcal{R}_0$ is a sum of terms involving D^{-1}, I, and D. (The form of $\mathcal{R}_1$ will be discussed below.) The coefficients of these terms are admissible power combinations of u_n, u_{n+1}, v_n, and v_{n-1} (which come from the terms on the right hand sides of (3)), so that all the terms have the correct rank. The maximum up-shift and down-shift operator that should be included can be determined by comparing two consecutive symmetries. Indeed, if the maximum up-shift in the first symmetry is u_{n+p} and the maximum up-shift in the next symmetry is u_{n+p+r}, then the associated piece that goes into $\mathcal{R}_0$ must have $\mathrm{D}, \mathrm{D}^2, \ldots, \mathrm{D}^r$. The same argument determines the minimum down-shift operator to be included. For (3), get

$$\mathcal{R}_0 = \begin{pmatrix} (\mathcal{R}_0)_{11} & (\mathcal{R}_0)_{12} \\ (\mathcal{R}_0)_{21} & (\mathcal{R}_0)_{22} \end{pmatrix}, \qquad (39)$$

with

$$
\begin{aligned}
(\mathcal{R}_0)_{11} &= (c_1 u_n + c_2 u_{n+1})\,\mathrm{I},\\
(\mathcal{R}_0)_{12} &= c_3 \mathrm{D}^{-1} + c_4 \mathrm{I},\\
(\mathcal{R}_0)_{21} &= (c_5 u_n^2 + c_6 u_n u_{n+1} + c_7 u_{n+1}^2 + c_8 v_{n-1} + c_9 v_n)\,\mathrm{I}\\
&\quad +(c_{10} u_n^2 + c_{11} u_n u_{n+1} + c_{12} u_{n+1}^2 + c_{13} v_{n-1}\\
&\quad +c_{14} v_n)\,\mathrm{D},\\
(\mathcal{R}_0)_{22} &= (c_{15} u_n + c_{16} u_{n+1})\,\mathrm{I}.
\end{aligned}
$$

As Hereman and Göktaş (1999) showed for the continuous case, $\mathcal{R}_1$ is a linear combination (with undetermined coefficients $\tilde{c}_{jk}$) of all suitable products of symmetries and covariants, i.e. Fréchet derivatives of densities, sandwiching $(\mathrm{D}-\mathrm{I})^{-1}$. Hence,

$$
\sum_j \sum_k \tilde{c}_{jk} \mathbf{G}^{(j)} (\mathrm{D}-\mathrm{I})^{-1} \otimes \rho_n^{(k)\prime}, \tag{40}
$$

where $\otimes$ denotes the matrix outer product, defined as

$$
\begin{pmatrix} G_1^{(j)} \\ G_2^{(j)} \end{pmatrix} (\mathrm{D}-\mathrm{I})^{-1} \otimes \begin{pmatrix} \rho_{n,1}^{(k)\prime} & \rho_{n,2}^{(k)\prime} \end{pmatrix} = \begin{pmatrix} G_1^{(j)} (\mathrm{D}-\mathrm{I})^{-1} \rho_{n,1}^{(k)\prime} & G_1^{(j)} (\mathrm{D}-\mathrm{I})^{-1} \rho_{n,2}^{(k)\prime} \\ G_2^{(j)} (\mathrm{D}-\mathrm{I})^{-1} \rho_{n,1}^{(k)\prime} & G_2^{(j)} (\mathrm{D}-\mathrm{I})^{-1} \rho_{n,2}^{(k)\prime} \end{pmatrix}. \tag{41}
$$

Only the pair $(\mathbf{G}^{(1)}, \rho_n^{(0)\prime})$ can be used, otherwise the ranks in (38) would be exceeded. Use (14) and (19), to compute

$$
\rho_n^{(0)\prime} = \begin{pmatrix} 0 & \dfrac{1}{v_n}\mathrm{I} \end{pmatrix}, \tag{42}
$$

From (40), after renaming $\tilde{c}_{10}$ to c_{17}, obtain

$$
\mathcal{R}_1 = \begin{pmatrix} 0 & c_{17}(v_{n-1}-v_n)(\mathrm{D}-\mathrm{I})^{-1} \dfrac{1}{v_n}\mathrm{I} \\ 0 & c_{17} v_n (u_n - u_{n+1})(\mathrm{D}-\mathrm{I})^{-1} \dfrac{1}{v_n}\mathrm{I} \end{pmatrix}. \tag{43}
$$

Add (39) and (43), to get

$$
\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 = \begin{pmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \mathcal{R}_{21} & \mathcal{R}_{22} \end{pmatrix}, \tag{44}
$$

with

$$
\begin{aligned}
\mathcal{R}_{11} &= (c_1 u_n + c_2 u_{n+1})\,\mathrm{I},\\
\mathcal{R}_{12} &= c_3 \mathrm{D}^{-1} + c_4 \mathrm{I} + c_{17}(v_{n-1} - v_n)(\mathrm{D}-\mathrm{I})^{-1} \frac{1}{v_n}\mathrm{I},\\
\mathcal{R}_{21} &= (c_5 u_n^2 + c_6 u_n u_{n+1} + c_7 u_{n+1}^2 + c_8 v_{n-1} + c_9 v_n)\,\mathrm{I}\\
&\quad +(c_{10} u_n^2 + c_{11} u_n u_{n+1} + c_{12} u_{n+1}^2 + c_{13} v_{n-1} + c_{14} v_n)\mathrm{D},\\
\mathcal{R}_{22} &= (c_{15} u_n + c_{16} u_{n+1})\mathrm{I} + c_{17} v_n (u_n - u_{n+1})(\mathrm{D}-\mathrm{I})^{-1} \frac{1}{v_n}\mathrm{I}.
\end{aligned}
$$

5.3 Determine the unknown coefficients

Compute all the terms in (23) to find the c_i. Refer to Hereman et al. (2004) for the details of the computation, resulting in $c_2 = c_5 = c_6 = c_7 = c_8 = c_{10} = c_{11} = c_{12} = c_{13} = c_{15} = 0$, $c_1 = c_3 = c_4 = c_9 = c_{14} = c_{16} = 1$, and $c_{17} = -1$. Substitute the constants into (44) to get (28).

ACKNOWLEDGEMENTS

J.A. Sanders, J.-P. Wang, M. Hickman, and B. Deconinck are gratefully acknowledged for valuable discussions.

REFERENCES

D.E. Baldwin and W. Hereman, A symbolic algorithm for computing recursion operators of nonlinear partial differential equations, Int. J. Comput. Math., 87:1094–1119, 2010.

A.S. Fokas, A symmetry approach to exactly solvable evolution equations, J. Math. Phys., 21:1318–1325, 1980.

A.S. Fokas, Symmetries and integrability, Stud. Appl. Math., 77:253–299, 1987.

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

Ü. Göktaş, *Algorithmic Computation of Symmetries, Invariants and Recursion Operators for Systems of Nonlinear Evolution and Differential-difference Equations*, Ph.D. Thesis, Colorado School of Mines, Golden, Colorado, 1998.

Ü. Göktaş and W. Hereman, Symbolic computation of conserved densities for systems of nonlinear evolution equations, J. Symb. Comput., 24:591–621, 1997.

Ü. Göktaş and W. Hereman, Computation of conservation laws for nonlinear lattices, Physica D, 132:425–436, 1998.

Ü. Göktaş and W. Hereman, Algorithmic computation of higher-order symmetries for nonlinear evolution and lattice equations, Adv. Comput. Math., 11:55–80, 1999.

M. Hénon, Integrals of the Toda lattice, Phys. Rev. B, 9:1921–1923, 1974.

W. Hereman, Software available at `http://www.mines.edu/~whereman/`, 2010.

W. Hereman, Ü. Göktaş, M.D. Colagrosso and A.J. Miller, Algorithmic integrability tests for nonlinear differential and lattice equations, Comput. Phys. Comm., 115:428–446, 1998.

W. Hereman and Ü. Göktaş, Integrability tests for nonlinear evolution equations. In M. Wester, editor, *Computer Algebra Systems: A Practical Guide*, pp. 211–232. Wiley, New York, 1999.

W. Hereman, J.A. Sanders, J. Sayers, and J.-P. Wang, Symbolic computation of polynomial conserved densities, generalized symmetries, and recursion operators for nonlinear differential-difference equations. In P. Winternitz *et al.*, editors, *Group Theory and Numerical Analysis*, CRM Proc. & Lect. Ser., volume 39, pp. 267–282. AMS, Providence, Rhode Island, 2004.

M. Hickman, Leading order integrability conditions for differential-difference equations. J. Nonl. Math. Phys., 15:66–86, 2008.

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

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

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

J.-P. Wang, *Symmetries and Conservation Laws of Evolution Equations*, Ph.D. Thesis, Thomas Stieltjes Institute for Mathematics, Amsterdam, 1998.