

Symbolic Computation of Polynomial Conserved Densities, Generalized Symmetries, and Recursion Operators for Nonlinear Differential-Difference Equations

Willy Hereman, Jan A. Sanders, Jack Sayers, and Jing Ping Wang

This paper is dedicated to Ryan Sayers (1982–2003)

ABSTRACT. Algorithms for the symbolic computation of polynomial conserved densities, fluxes, generalized symmetries, and recursion operators for systems of nonlinear differential-difference equations are presented. In the algorithms we use discrete versions of the Fréchet and variational derivatives and the Euler and homotopy operators.

The algorithms are illustrated for prototypical nonlinear polynomial lattices, including the Kac–van Moerbeke (Volterra) and Toda lattices. Results are shown for the modified Volterra and Ablowitz–Ladik lattices.

1. Introduction

The study of complete integrability of nonlinear differential-difference equations (DDEs) largely parallels that of PDEs [1, 3, 24]. Indeed, as in the continuous case, the existence of large numbers of generalized symmetries and conserved densities is a good indicator for complete integrability. Albeit useful, such predictors do not provide proof of complete integrability. Based on the first few densities and symmetries, quite often one can explicitly construct a recursion operator which maps symmetries of the equation into new symmetries. The existence of a recursion operator, which allows one to generate commuting symmetries, then confirms complete integrability.

There is a vast body of work on the complete integrability of DDEs. Consult e.g. [3, 13] for various approaches and additional references. In this article we describe algorithms to symbolically compute polynomial conservation laws, generalized symmetries, and recursion operators for DDEs. The design of these algorithms

2000 *Mathematics Subject Classification*. Primary: 37J35, 37K10; Secondary: 35Q58, 37K05.

This material is based upon work supported by the National Science Foundation under Grant no. CCR-9901929.

M. Hickman and B. Deconinck are gratefully acknowledged for valuable discussions. D. Baldwin is thanked for proof reading the manuscript.

This is the final form of the paper.

relies on related work for PDEs [6, 11, 12] and work by Oevel *et al* [19] and Zhang *et al* [25].

In our algorithms we use discrete versions of the Fréchet and variational derivatives, as well as discrete Euler and homotopy operators. We draw on the close analogy between the continuous and discrete (in space) cases. As shown in [14, 16], this analogy can be completely formalized and both theories can be formulated in terms of complexes. The same applies for the formulation in terms of Lie algebra complexes in [24]. This allows one to translate by analogy the existing algorithms immediately (although complications arise when there is explicit dependence on the lattice point, n , in the discrete case). One of the more useful tools following from the abstract theory is the homotopy operator, which is based on scaling vector fields, and goes back to Poincaré in the continuous case. This operator allows one to directly integrate differential forms and can be straightforwardly implemented in computer algebra systems (CAS), since it reduces to integration over one scaling parameter. The discrete analogue as described in [14, 16] does the corresponding job in the discrete case and we use it to compute fluxes. In this paper we do not explicitly use the abstract framework, yet, it has been a motivating force for the development of our algorithms.

In contrast to the symmetry approach in [3], our algorithms rely on specific assumptions. For example, we will use the dilation invariance of DDEs in the construction of densities, symmetries, and recursion operators. At the cost of generality, our algorithms can be implemented in major CAS. Our Mathematica package `InvariantsSymmetries.m` [7] computes densities and symmetries, and therefore aids in automated testing of complete integrability of semi-discrete lattices. Mathematica code to compute recursion operators of DDEs is under development.

The paper is organized as follows. In Section 2, we present key definitions, necessary tools, and prototypical examples which will be used throughout the paper. The examples include the Kac–van Moerbeke (KvM) [15], Toda [23] and Ablowitz–Ladik (AL) lattices [2]. The discrete higher Euler operators (variational derivatives) and the discrete homotopy operator are introduced in Section 3. These operators are applied in the construction of densities and fluxes in Section 5. The algorithm for higher-order symmetries is outlined in Section 6. The algorithms for scalar and matrix recursion operators are given in Section 7 and 8. The paper concludes with two more examples in Section 9.

2. Key definitions and prototypical examples

DEFINITION 2.1. A nonlinear (autonomous) DDE is an equation of the form

$$(2.1) \quad \dot{\mathbf{u}}_n = \mathbf{F}(\dots, \mathbf{u}_{n-1}, \mathbf{u}_n, \mathbf{u}_{n+1}, \dots),$$

where \mathbf{u}_n and \mathbf{F} are vector-valued functions with N components. The integer n corresponds to discretization in space; the dot denotes one derivative with respect to the continuous time variable (t).

For simplicity, we denote the components of \mathbf{u}_n by (u_n, v_n, w_n, \dots) and write $\mathbf{F}(\mathbf{u}_n)$, although \mathbf{F} typically depends on \mathbf{u}_n and a finite number of its forward and backward shifts. We assume that \mathbf{F} is polynomial with constant coefficients. In present, parameters are denoted by lower-case Greek letters. No restrictions are imposed on the forward or backward shifts or the degree of nonlinearity in \mathbf{F} .

EXAMPLE 2.2. The AL lattice [2],

$$(2.2) \quad \begin{aligned} \dot{u}_n &= (u_{n+1} - 2u_n + u_{n-1}) + u_n v_n (u_{n+1} + u_{n-1}), \\ \dot{v}_n &= -(v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}), \end{aligned}$$

is a completely integrable discretization of the nonlinear Schrödinger equation.

DEFINITION 2.3. A DDE is said to be dilation invariant if it is invariant under a scaling (dilation) symmetry.

EXAMPLE 2.4. The KvM lattice [15],

$$(2.3) \quad \dot{u}_n = u_n (u_{n+1} - u_{n-1}),$$

is invariant under $(t, u_n) \rightarrow (\lambda^{-1}t, \lambda u_n)$, where λ is an arbitrary scaling parameter.

EXAMPLE 2.5. The Toda lattice [23] in polynomial form [8],

$$(2.4) \quad \dot{u}_n = v_{n-1} - v_n, \quad \dot{v}_n = v_n (u_n - u_{n+1}),$$

is invariant under the scaling symmetry

$$(2.5) \quad (t, u_n, v_n) \rightarrow (\lambda^{-1}t, \lambda^1 u_n, \lambda^2 v_n).$$

DEFINITION 2.6. We define the weight, w , of a variable as the exponent in the scaling parameter λ which multiplies the variable.

As a result of definition 2.6, t is always replaced by $\frac{t}{\lambda}$ and $w(d/dt) = w(D_t) = 1$. In view of (2.5), we have $w(u_n) = 1$, and $w(v_n) = 2$ for the Toda lattice.

Weights of dependent variables are nonnegative, rational, and independent of n . For example, $w(u_{n-1}) = w(u_n) = w(u_{n+1})$, etc.

DEFINITION 2.7. The rank of a monomial is defined as the total weight of the monomial. An expression is uniform in rank if all its terms have the same rank.

EXAMPLE 2.8. In the first equation of (2.4), 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 (2.4) allows one to compute the weights of the dependent variables (and thus the scaling symmetry) with simple linear algebra. Indeed,

$$(2.6) \quad w(u_n) + 1 = w(v_n), \quad w(v_n) + 1 = w(u_n) + w(v_n),$$

yields

$$(2.7) \quad w(u_n) = 1, \quad w(v_n) = 2,$$

which is consistent with (2.5).

Dilation symmetries, which are special Lie-point symmetries, are common to many lattice equations. Lattices described by 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.

EXAMPLE 2.9. The AL lattice is not dilation invariant. We therefore introduce an auxiliary parameter α . Replacing (2.2) by

$$(2.8) \quad \begin{aligned} \dot{u}_n &= \alpha(u_{n+1} - 2u_n + u_{n-1}) + u_n v_n (u_{n+1} + u_{n-1}), \\ \dot{v}_n &= -\alpha(v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}), \end{aligned}$$

and requiring uniformity in rank then gives

$$(2.9) \quad \begin{aligned} w(u_n) + 1 &= w(\alpha) + w(u_n) = 2w(u_n) + w(v_n), \\ w(v_n) + 1 &= w(\alpha) + w(v_n) = 2w(v_n) + w(u_n). \end{aligned}$$

Obviously, $w(u_n) + w(v_n) = w(\alpha) = 1$. Several scales are possible. The choice $w(u_n) = w(v_n) = \frac{1}{2}, w(\alpha) = 1$, corresponds to the scaling symmetry

$$(2.10) \quad (t, u_n, v_n, \alpha) \rightarrow (\lambda^{-1}t, \lambda^{1/2}u_n, \lambda^{1/2}v_n, \lambda\alpha).$$

DEFINITION 2.10. A scalar function $\rho_n(\mathbf{u}_n)$ is a conserved density of (2.1) if there exists a scalar function $J_n(\mathbf{u}_n)$, called the associated flux, such that [20]

$$(2.11) \quad D_t \rho_n + \Delta J_n = 0$$

is satisfied on the solutions of (2.1).

In (2.11), we used the (forward) difference operator, $\Delta = D - I$, defined by

$$(2.12) \quad \Delta J_n = (D - I)J_n = J_{n+1} - J_n,$$

where D denotes the up-shift (forward or right-shift) operator, $DJ_n = J_{n+1}$, and I is the identity operator. The operator Δ takes the role of a spatial derivative on the shifted variables as many DDEs arise from discretization of a PDE in $(1 + 1)$ variables. Most, but not all, densities are polynomial in \mathbf{u}_n .

EXAMPLE 2.11. The first three density-flux pairs [12] for (2.3) are

$$(2.13) \quad \rho_n^{(0)} = \ln(u_n), \quad J_n^{(0)} = u_n + u_{n-1},$$

$$(2.14) \quad \rho_n^{(1)} = u_n, \quad J_n^{(1)} = u_n u_{n-1},$$

$$(2.15) \quad \rho_n^{(2)} = \frac{1}{2}u_n^2 + u_n u_{n+1}, \quad J_n^{(2)} = u_{n-1}u_n(u_n + u_{n+1}).$$

EXAMPLE 2.12. The first four density-flux pairs [8] for (2.4) are

$$(2.16) \quad \rho_n^{(0)} = \ln(v_n), \quad J_n^{(0)} = u_n,$$

$$(2.17) \quad \rho_n^{(1)} = u_n, \quad J_n^{(1)} = v_{n-1},$$

$$(2.18) \quad \rho_n^{(2)} = \frac{1}{2}u_n^2 + v_n, \quad J_n^{(2)} = u_n v_{n-1},$$

$$(2.19) \quad \rho_n^{(3)} = \frac{1}{3}u_n^3 + u_n(v_{n-1} + v_n), \quad J_n^{(3)} = u_{n-1}u_n v_{n-1} + v_{n-1}^2.$$

The densities in (2.16)–(2.19) are uniform of ranks 0 through 3. The corresponding fluxes are also uniform in rank with ranks 1 through 4. In general, if in (2.11) rank $\rho_n = R$ then rank $J_n = R + 1$, since $w(D_t) = 1$. All the pieces in (2.11) are uniform in rank. This comes as no surprise since the conservation law (2.11) holds on solutions of (2.1), hence it ‘inherits’ the dilation symmetry of (2.1).

In Section 5 we will give an algorithm to compute polynomial conserved densities and fluxes and use (2.4) to illustrate the steps. Non-polynomial densities (which are usually easy to find by hand) can be computed with the method given in [13].

DEFINITION 2.13. Compositions of D and its inverse D^{-1} define an equivalence relation (\equiv) on monomials in the components of \mathbf{u}_n . All shifted monomials are equivalent.

EXAMPLE 2.14. For example, $u_{n-1}v_{n+1} \equiv u_n v_{n+2} \equiv u_{n+1}v_{n+3} \equiv u_{n+2}v_{n+4}$.

Factors in a monomial in u_n and its shifts are ordered by $u_{n+j} \prec u_{n+k}$ if $j < k$.

DEFINITION 2.15. The main representative of an equivalence class of monomials in u_n and its shifts is the monomial with u_n in the first position.

EXAMPLE 2.16. For example, $u_n u_{n+2}$ (not $u_{n-2} u_n$) is the main representative of the class $\{\dots, u_{n-2} u_n, u_{n-1} u_{n+1}, u_n u_{n+2}, u_{n+1} u_{n+3}, \dots\}$.

For monomials involving u_n, v_n, w_n , etc. and their shifts, we lexicographically order the variables, i.e. $u_n \prec v_n \prec w_n$, etc.

EXAMPLE 2.17. For example, $u_n v_{n+2}$ (not $u_{n-2} v_n$) is the main representative of the class $\{\dots, u_{n-2} v_n, u_{n-1} v_{n+1}, u_n v_{n+2}, \dots\}$.

DEFINITION 2.18. A vector function $\mathbf{G}(\mathbf{u}_n)$ is called a generalized symmetry of (2.1) if the infinitesimal transformation $\mathbf{u}_n \rightarrow \mathbf{u}_n + \varepsilon \mathbf{G}$ leaves (2.1) invariant up to order ε . Consequently, \mathbf{G} must satisfy [20]

$$(2.20) \quad D_t \mathbf{G} = \mathbf{F}'(\mathbf{u}_n)[\mathbf{G}]$$

on solutions of (2.1). $\mathbf{F}'(\mathbf{u}_n)[\mathbf{G}]$ is the Fréchet derivative of \mathbf{F} in the direction of \mathbf{G} .

For the scalar case ($N = 1$), the Fréchet derivative in the direction of G is

$$(2.21) \quad F'(u_n)[G] = \frac{\partial}{\partial \varepsilon} F(u_n + \varepsilon G)|_{\varepsilon=0} = \sum_k \frac{\partial F}{\partial u_{n+k}} D^k G,$$

which defines the Fréchet derivative operator

$$(2.22) \quad F'(u_n) = \sum_k \frac{\partial F}{\partial u_{n+k}} D^k.$$

In the vector case with components u_n and v_n , the Fréchet derivative operator is

$$(2.23) \quad \mathbf{F}'(\mathbf{u}_n) = \begin{pmatrix} \sum_k \frac{\partial F_1}{\partial u_{n+k}} D^k & \sum_k \frac{\partial F_1}{\partial v_{n+k}} D^k \\ \sum_k \frac{\partial F_2}{\partial u_{n+k}} D^k & \sum_k \frac{\partial F_2}{\partial v_{n+k}} D^k \end{pmatrix}.$$

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

$$(2.24) \quad F'_i(\mathbf{u}_n)[\mathbf{G}] = \sum_k \frac{\partial F_i}{\partial u_{n+k}} D^k G_1 + \sum_k \frac{\partial F_i}{\partial v_{n+k}} D^k G_2, \quad i = 1, 2.$$

In (2.21) and (2.24) summation is over all positive and negative shifts (including $k = 0$). For $k > 0$, $D^k = D \circ D \circ \dots \circ D$ (k times). Similarly, for $k < 0$ the down-shift operator D^{-1} is applied repeatedly. The generalization of (2.23) to a system with N components should be obvious.

EXAMPLE 2.19. The first two symmetries [12] of (2.3) are

$$(2.25) \quad G^{(1)} = u_n(u_{n+1} - u_{n-1}),$$

$$(2.26) \quad G^{(2)} = u_n u_{n+1}(u_n + u_{n+1} + u_{n+2}) - u_{n-1} u_n(u_{n-2} + u_{n-1} + u_n).$$

These symmetries are uniform in rank (rank 2 and 3, respectively). The symmetries of ranks 0 and 1 are both zero.

EXAMPLE 2.20. The first two non-trivial symmetries [12] of (2.4) are

$$(2.27) \quad \mathbf{G}^{(1)} = \begin{pmatrix} v_{n-1} - v_n \\ v_n(u_n - u_{n+1}) \end{pmatrix},$$

$$(2.28) \quad \mathbf{G}^{(2)} = \begin{pmatrix} v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} + u_n) \\ v_n(u_{n+1}^2 - u_n^2 + v_{n+1} - v_{n-1}) \end{pmatrix}.$$

The above symmetries are uniform in rank. For example, $\text{rank } G_1^{(2)} = 3$ and $\text{rank } G_2^{(2)} = 4$. Symmetries of lower ranks are trivial.

An algorithm to compute polynomial generalized symmetries will be outlined in Section 6 and applied to (2.4).

DEFINITION 2.21. A recursion operator \mathcal{R} connects symmetries

$$(2.29) \quad \mathbf{G}^{(j+s)} = \mathcal{R}\mathbf{G}^{(j)},$$

where $j = 1, 2, \dots$, 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.

The defining equation for \mathcal{R} [20, 24] is

$$(2.30) \quad D_t \mathcal{R} + [\mathcal{R}, \mathbf{F}'(\mathbf{u}_n)] = \frac{\partial \mathcal{R}}{\partial t} + \mathcal{R}'[\mathbf{F}] + \mathcal{R} \circ \mathbf{F}'(\mathbf{u}_n) - \mathbf{F}'(\mathbf{u}_n) \circ \mathcal{R} = 0,$$

where $[\ , \]$ denotes the commutator and \circ the composition of operators. The operator $\mathbf{F}'(\mathbf{u}_n)$ was defined in (2.23). $\mathcal{R}'[\mathbf{F}]$ 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

$$(2.31) \quad \mathcal{R} = U(u_n)\mathcal{O}((D - I)^{-1}, D^{-1}, I, D)V(u_n),$$

and then

$$(2.32) \quad \mathcal{R}'[F] = \sum_k (D^k F) \frac{\partial U}{\partial u_{n+k}} \mathcal{O}V + \sum_k U \mathcal{O}(D^k F) \frac{\partial V}{\partial u_{n+k}}.$$

For the vector case, the elements of the $N \times N$ operator matrix \mathcal{R} are often of the form $\mathcal{R}_{ij} = U_{ij}(\mathbf{u}_n)\mathcal{O}_{ij}((D - I)^{-1}, D^{-1}, I, D)V_{ij}(\mathbf{u}_n)$. For the two-component case

$$(2.33) \quad \begin{aligned} \mathcal{R}'[\mathbf{F}]_{ij} = & \sum_k (D^k F_1) \frac{\partial U_{ij}}{\partial u_{n+k}} \mathcal{O}_{ij} V_{ij} + \sum_k (D^k F_2) \frac{\partial U_{ij}}{\partial v_{n+k}} \mathcal{O}_{ij} V_{ij} \\ & + \sum_k U_{ij} \mathcal{O}_{ij} (D^k F_1) \frac{\partial V_{ij}}{\partial u_{n+k}} + \sum_k U_{ij} \mathcal{O}_{ij} (D^k F_2) \frac{\partial V_{ij}}{\partial v_{n+k}}. \end{aligned}$$

EXAMPLE 2.22. The KvM lattice (2.3) has recursion operator

$$(2.34) \quad \begin{aligned} \mathcal{R} = & u_n(I + D)(u_n D - D^{-1}u_n)(D - I)^{-1} \frac{1}{u_n} I \\ = & u_n D^{-1} + (u_n + u_{n+1})I + u_n D + u_n(u_{n+1} - u_{n-1})(D - I)^{-1} \frac{1}{u_n} I. \end{aligned}$$

EXAMPLE 2.23. The Toda lattice (2.4) has recursion operator

$$(2.35) \quad \mathcal{R} = \begin{pmatrix} -u_n I & -D^{-1} - I + (v_{n-1} - v_n)(D - I)^{-1} \frac{1}{v_n} I \\ -v_n I - v_n D & -u_{n+1} I + v_n(u_n - u_{n+1})(D - I)^{-1} \frac{1}{v_n} I \end{pmatrix}.$$

In Section 7 we will give an algorithm for the computation of scalar recursion operators like (2.34). In Section 8 we cover the matrix case and show how (2.35) is computed. The algorithms complement those for recursion operators of PDEs presented in [11] and elsewhere in these proceedings [4].

We now introduce two powerful tools which will be used in the computation of densities and fluxes.

3. The discrete variational derivative (Euler operator)

DEFINITION 3.1. A function $E_n(\mathbf{u}_n)$ is a total difference if there exists a function $J_n(\mathbf{u}_n)$ such that $E_n = \Delta J_n = (D - I)J_n$.

THEOREM 3.2. A necessary and sufficient condition for a function E_n with positive shifts up to level p_0 , to be a total difference is that

$$(3.1) \quad \mathcal{L}_{\mathbf{u}_n}^{(0)}(E_n) = 0.$$

$\mathcal{L}_{\mathbf{u}_n}^{(0)}$ is the discrete variational derivative (Euler operator) [3] defined by

$$(3.2) \quad \mathcal{L}_{\mathbf{u}_n}^{(0)} = \frac{\partial}{\partial \mathbf{u}_n} \left(\sum_{k=0}^{p_0} D^{-k} \right) = \frac{\partial}{\partial \mathbf{u}_n} (I + D^{-1} + D^{-2} + \dots + D^{-p_0}).$$

A proof is given in e.g. [13].

REMARK 3.3. To verify that an expression $E(u_{n-q}, \dots, u_n, \dots, u_{n+p})$ involving negative shifts is a total difference, one must first remove the negative shifts by replacing E_n by $\tilde{E}_n = D^q E_n$. Applied to \tilde{E}_n , (3.2) terminates at $p_0 = p + q$.

We now introduce a tool to invert the total difference operator $\Delta = D - I$.

4. The discrete homotopy operator

We assume that $\mathcal{L}_{\mathbf{u}_n}^{(0)}(E_n) = 0$. So, $\exists J_n$ such that $E_n = \Delta J_n$. To compute $J_n = \Delta^{-1}(E_n)$ one must invert the operator $\Delta = D - I$. Working with the formal inverse, $\Delta^{-1} = D^{-1} + D^{-2} + D^{-3} + \dots$, is impractical, perhaps impossible. We therefore present the (discrete) homotopy operator which circumvents the formal infinite series. In analogy to the continuous case [20], we first introduce the discrete higher Euler operators.

DEFINITION 4.1. The discrete higher Euler operators are defined by

$$(4.1) \quad \mathcal{L}_{\mathbf{u}_n}^{(i)} = \frac{\partial}{\partial \mathbf{u}_n} \left(\sum_{k=i}^{p_i} \binom{k}{i} D^{-k} \right), \quad i \geq 1.$$

The higher Euler operators all terminate at some maximal shift p_i .

EXAMPLE 4.2. For scalar component u_n , the first higher Euler operators are:

$$(4.2) \quad \mathcal{L}_{u_n}^{(1)} = \frac{\partial}{\partial u_n} (D^{-1} + 2D^{-2} + 3D^{-3} + 4D^{-4} + \dots + p_1 D^{-p_1}),$$

$$(4.3) \quad \mathcal{L}_{u_n}^{(2)} = \frac{\partial}{\partial u_n} (D^{-2} + 3D^{-3} + 6D^{-4} + 10D^{-5} + \dots + \frac{1}{2}p_2(p_2 - 1)D^{-p_2}),$$

where the p_i are the highest shifts in the arguments of the operators.

Similar formulae hold for $\mathcal{L}_{v_n}^{(i)}$. Next, we introduce the homotopy operators. For simplicity we show the formulae for the two component case $\mathbf{u}_n = (u_n, v_n)$.

DEFINITION 4.3. The homotopy operator is defined as,

$$(4.4) \quad \mathcal{H}_{\mathbf{u}_n} = \int_0^1 (f_{1,n}(\mathbf{u}_n)[\lambda\mathbf{u}_n] + f_{2,n}(\mathbf{u}_n)[\lambda\mathbf{u}_n]) \frac{d\lambda}{\lambda},$$

where

$$(4.5) \quad f_{1,n}(\mathbf{u}_n) = \sum_{i=0}^{m_1-1} (D - \mathbf{I})^i (u_n \mathcal{L}_{u_n}^{(i+1)}), \quad f_{2,n}(\mathbf{u}_n) = \sum_{i=0}^{m_2-1} (D - \mathbf{I})^i (v_n \mathcal{L}_{v_n}^{(i+1)}),$$

where m_1 and m_2 are the highest shifts in u_n and v_n in the arguments of the operators. Furthermore, $f_{1,n}(\mathbf{u}_n)[\lambda\mathbf{u}_n]$ means that in $f_{1,n}(\mathbf{u}_n)$ one replaces $\mathbf{u}_n \rightarrow \lambda\mathbf{u}_n$, and, of course, $\mathbf{u}_{n+1} \rightarrow \lambda\mathbf{u}_{n+1}$, $\mathbf{u}_{n+2} \rightarrow \lambda\mathbf{u}_{n+2}$, etc. Same for $f_{2,n}(\mathbf{u}_n)[\lambda\mathbf{u}_n]$.

THEOREM 4.4. $J_n = \Delta^{-1}(E_n)$ can be computed as $J_n = \mathcal{H}_{\mathbf{u}_n}(E_n)$.

A similar theorem and proof for continuous homotopy operators is given in [20, p. 372]. By constructing an appropriate discrete variational bicomplex one can prove the above theorem. See [14, 16] and elsewhere in these proceedings [17].

5. Computation of polynomial conserved densities and fluxes

As an example, we compute density-flux pair (2.19) for (2.4). Assuming that the weights (2.7) are computed and the rank of the density is selected ($R = 3$ here), the algorithm has three steps. See [8, 10, 12] for details about the algorithm and alternate ways to compute fluxes.

STEP 1 (Construct the form of the density). List all monomials in u_n and v_n of rank 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 weight 3. Thus, using (2.4),

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

Gather all terms in $\mathcal{N} = \{u_n^3, u_n v_{n-1}, u_n v_n, u_{n-1} v_{n-1}, u_{n+1} v_n\}$. Identify members belonging to the same equivalence classes and replace them by their main representatives. For example, $u_n v_{n-1} \equiv u_{n+1} v_n$, so the latter is replaced by $u_n v_{n-1}$. So, \mathcal{N} is replaced by $\mathcal{P} = \{u_n^3, u_n v_{n-1}, u_n v_n\}$, which has the building blocks of the density. Linear combination of the monomials in \mathcal{P} with constant coefficients c_i gives the candidate density,

$$(5.1) \quad \rho_n = c_1 u_n^3 + c_2 u_n v_{n-1} + c_3 u_n v_n.$$

STEP 2 (Determine the coefficients). Require that (2.11) holds. Compute $D_t \rho_n$. Use (2.4) to remove \dot{u}_n and \dot{v}_n and their shifts. Thus,

$$(5.2) \quad E_n = D_t \rho_n = (3c_1 - c_2) u_n^2 v_{n-1} + (c_3 - 3c_1) u_n^2 v_n + (c_3 - c_2) v_{n-1} v_n + c_2 u_{n-1} u_n v_{n-1} + c_2 v_{n-1}^2 - c_3 u_n u_{n+1} v_n - c_3 v_n^2.$$

TABLE 1. Computation of $f_{1,n}(\mathbf{u}_n)(-\tilde{E}_n)$.

| i | $\mathcal{L}_{u_n}^{(i+1)}(-\tilde{E}_n)$ | $(D - I)^i(u_n \mathcal{L}_{u_n}^{(i+1)}(-\tilde{E}_n))$ |
|-----|---|--|
| 0 | $u_{n-1}v_{n-1} + u_{n+1}v_n$ | $u_n u_{n-1} v_{n-1} + u_n u_{n+1} v_n$ |
| 1 | $u_{n-1}v_{n-1}$ | $u_{n+1}u_n v_n - u_n u_{n-1} v_{n-1}$ |

 TABLE 2. Computation of $f_{2,n}(\mathbf{u}_n)(-\tilde{E}_n)$.

| i | $\mathcal{L}_{v_n}^{(i+1)}(-\tilde{E}_n)$ | $(D - I)^i(v_n \mathcal{L}_{v_n}^{(i+1)}(-\tilde{E}_n))$ |
|-----|---|--|
| 0 | $u_n u_{n+1} + 2v_n$ | $v_n u_n u_{n+1} + 2v_n^2$ |

Compute $\tilde{E}_n = DE_n$. First, apply (3.2) for component u_n to \tilde{E}_n :

$$\begin{aligned}
 (5.3) \quad \mathcal{L}_{u_n}^{(0)}(\tilde{E}_n) &= \frac{\partial}{\partial u_n} (I + D^{-1} + D^{-2})(\tilde{E}_n) \\
 &= 2(3c_1 - c_2)u_n v_{n-1} + 2(c_3 - 3c_1)u_n v_n \\
 &\quad + (c_2 - c_3)u_{n-1}v_{n-1} + (c_2 - c_3)u_{n+1}v_n.
 \end{aligned}$$

Second, apply (3.2) for component v_n to \tilde{E}_n :

$$\begin{aligned}
 (5.4) \quad \mathcal{L}_{v_n}^{(0)}(\tilde{E}_n) &= \frac{\partial}{\partial v_n} (I + D^{-1})(\tilde{E}_n) \\
 &= (3c_1 - c_2)u_{n+1}^2 + (c_3 - c_2)v_{n+1} + (c_2 - c_3)u_n u_{n+1} \\
 &\quad + 2(c_2 - c_3)v_n + (c_3 - 3c_1)u_n^2 + (c_3 - c_2)v_{n-1}.
 \end{aligned}$$

Both (5.3) and (5.4) must vanish identically. Solve the linear system

$$(5.5) \quad \mathcal{S} = \{3c_1 - c_2 = 0, c_3 - 3c_1 = 0, c_2 - c_3 = 0\}.$$

The solution is $3c_1 = c_2 = c_3$. Substituting $c_1 = \frac{1}{3}, c_2 = c_3 = 1$, into (5.1)

$$(5.6) \quad \rho_n = \frac{1}{3}u_n^3 + u_n(v_{n-1} + v_n).$$

STEP 3 (Compute the flux). In view of (2.11), we will compute $J_n = \Delta^{-1}(-E_n)$ with the homotopy operator introduced in Section 4. Alternative methods are described in [8, 13].

Insert $c_1 = \frac{1}{3}, c_2 = c_3 = 1$ into (5.2) and compute

$$(5.7) \quad \tilde{E}_n = DE_n = u_n u_{n+1} v_n + v_n^2 - u_{n+1} u_{n+2} v_{n+1} - v_{n+1}^2.$$

We apply formulae (4.5) to $-\tilde{E}_n$. The pieces are listed in Tables 1 and 2.

Adding the terms in the right columns in Table 1 and Table 2,

$$(5.8) \quad f_{1,n}(\mathbf{u}_n)(-\tilde{E}_n) = 2u_n u_{n+1} v_n, \quad f_{2,n}(\mathbf{u}_n)(-\tilde{E}_n) = u_n u_{n+1} v_n + 2v_n^2.$$

Thus, the homotopy operator (4.4) gives

$$\begin{aligned}
 (5.9) \quad \tilde{J}_n &= \int_0^1 (f_{1,n}(\mathbf{u}_n)(-\tilde{E}_n)[\lambda \mathbf{u}_n] + f_{2,n}(\mathbf{u}_n)(-\tilde{E}_n)[\lambda \mathbf{u}_n]) \frac{d\lambda}{\lambda} \\
 &= \int_0^1 (3\lambda^2 u_n u_{n+1} v_n + 2\lambda v_n^2) d\lambda \\
 &= u_n u_{n+1} v_n + v_n^2.
 \end{aligned}$$

After a backward shift, $J_n = D^{-1} \tilde{J}_n$, we obtain the final result:

$$(5.10) \quad \rho_n = \frac{1}{3} u_n^3 + u_n (v_{n-1} + v_n), \quad J_n = u_{n-1} u_n v_{n-1} + v_{n-1}^2.$$

6. Computation of polynomial generalized symmetries

As an example, we compute symmetry (2.28) of rank (3, 4) for (2.4). The two steps of the algorithm [9] are similar to those in Section 5.

STEP 1 (Construct the form of the symmetry). Start by listing all monomials in u_n and v_n of ranks 3 and 4, or less:

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

As in Step 1 in Section 5, for each monomial in \mathcal{M}_1 and \mathcal{M}_2 , introduce the necessary t -derivatives so that each term has rank 3 and 4, respectively. At the same time, use (2.4) to remove all t -derivatives. Doing so, based on \mathcal{M}_1 we obtain

$$(6.1) \quad \mathcal{N}_1 = \{u_n^3, u_{n-1} v_{n-1}, u_n v_{n-1}, u_n v_n, u_{n+1} v_n\}.$$

Similarly, based on \mathcal{M}_2 , we find

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

In contrast to the strategy for densities, we do not introduce the main representatives, but linearly combine the monomials in \mathcal{N}_1 and \mathcal{N}_2 with constant coefficients c_i to get the form of the candidate symmetry:

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

To compute the coefficients c_i we require that (2.20) holds on solutions of (2.1). Compute $D_t G_1$ and $D_t G_2$ and use (2.4) to remove \dot{u}_n, \dot{v}_n , and their shifts. This gives the left hand sides of (2.20). Use (2.24) to get the right hand sides of (2.20):

$$\begin{aligned}
 (6.4) \quad F_1'(\mathbf{u}_n)[\mathbf{G}] &= D^{-1} G_2 - I G_2, \\
 F_2'(\mathbf{u}_n)[\mathbf{G}] &= v_n I G_1 - v_n D G_1 + (u_n - u_{n+1}) I G_2.
 \end{aligned}$$

Substitute (6.3) into (6.4) and equate the corresponding left and right hand sides. Since all monomials in u_n, v_n and their shifts are independent, one obtains the linear system that determines the coefficients c_i . The solution is

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

With the choice $c_{17} = 1$, symmetry (6.3) finally becomes

$$(6.6) \quad \begin{aligned} G_1 &= v_n(u_n + u_{n+1}) - v_{n-1}(u_{n-1} - u_n), \\ G_2 &= v_n(u_{n+1}^2 - u_n^2 + v_{n+1} - v_{n-1}). \end{aligned}$$

7. Computation of scalar recursion operators

Recursion operators can be constructed as $\mathcal{R} = H_2 H_1^{-1}$, where H_1 and H_2 are appropriate Hamiltonian operators [19]. We use a less general but direct method based on dilation invariance. Furthermore, we assume that the form of discrete recursion operators mimics its continuous counterpart [5, 18, 21, 22, 24]. With the appropriate ansatz for \mathcal{R} , defining equation (2.30) is then used to determine the coefficients.

In this section we show how to compute recursion operator (2.34) of (2.4). We observe that (2.34) in expanded form naturally splits into two pieces:

$$(7.1) \quad \mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1,$$

where \mathcal{R}_0 contains only terms with shift operators D^{-1} , I , and D , and \mathcal{R}_1 has terms involving $(D - I)^{-1}$.

STEP 1 (Determine the rank of the recursion operator). In view of (2.29), and assuming that the symmetries are linked consecutively ($s = 1$), the recursion operator \mathcal{R} has rank

$$(7.2) \quad R = \text{rank } \mathcal{R} = \text{rank } G^{(2)} - \text{rank } G^{(1)} = 3 - 2 = 1,$$

where we used (2.25) and (2.26). If the assumption turns out to be correct then the recursion operator has rank 1. If the assumption fails because symmetries are not linked consecutively, then R must be adjusted and the subsequent steps must be repeated. See [4] for examples of PDEs for which that happens.

STEP 2 (Determine the coefficients). We split this into two sub-steps.

(i) *Determine the form of \mathcal{R}_0 .* The candidate \mathcal{R}_0 is a sum of terms involving D^{-1} , I , and D . The coefficients of these terms are linear combinations of u_{n-1} , u_n , and u_{n+1} , so that all terms have the correct rank.

$$(7.3) \quad \begin{aligned} \mathcal{R}_0 &= (c_1 u_{n-1} + c_2 u_n + c_3 u_{n+1}) D^{-1} + (c_4 u_{n-1} + c_5 u_n + c_6 u_{n+1}) I \\ &\quad + (c_7 u_{n-1} + c_8 u_n + c_9 u_{n+1}) D, \end{aligned}$$

where the c_i are constant coefficients.

A few remarks are in place. First, in \mathcal{R}_0 we moved the operators D^{-1} , I , and D all the way to the right. Second, 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 \mathcal{R}_0 must have D, D^2, \dots, D^r . The same line of reasoning determines the minimum down-shift operator to be included. In our example, there is no need to include terms in D^{-2} , D^2 , etc. Third, the coefficients of the operators can be restricted to linear combinations of the terms appearing in F . Hence, no terms in $u_{n\pm 2}$, $u_{n\pm 3}$ and so on occur in (7.3).

(ii) *Determine the form of \mathcal{R}_1 .* As in the continuous case [11], \mathcal{R}_1 is a linear combination (with constant coefficients \tilde{c}_{jk}) of sums of all suitable products of

symmetries and covariants (Fréchet derivatives of densities) sandwiching $(D - I)^{-1}$, i.e.

$$(7.4) \quad \mathcal{R}_1 = \sum_j \sum_k \tilde{c}_{jk} G^{(j)} (D - I)^{-1} \rho_n^{(k)'}$$

For (2.3), $G^{(1)}$ in (2.25) and $\rho_n^{(0)} = \ln(u_n)$ in (2.13) are the only suitable pair. Indeed, using (2.22) we have $\rho_n^{(0)'} = \ln(u_n)' = 1/u_n I$, which has rank -1 . Combined with $G^{(1)}$ of rank 2, we have a term of rank 1. Other combinations of symmetries and covariants would exceed rank 1. Therefore,

$$(7.5) \quad \mathcal{R}_1 = \tilde{c}_{10} u_n (u_{n+1} - u_{n-1}) (D - I)^{-1} \left(\frac{1}{u_n} \right) I$$

where \tilde{c}_{10} a constant coefficient. Using (7.1) and renaming \tilde{c}_{10} to c_{10} ,

$$(7.6) \quad \mathcal{R} = (c_1 u_{n-1} + c_2 u_n + c_3 u_{n+1}) D^{-1} + (c_4 u_{n-1} + c_5 u_n + c_6 u_{n+1}) I \\ + (c_7 u_{n-1} + c_8 u_n + c_9 u_{n+1}) D + c_{10} u_n (u_{n+1} - u_{n-1}) (D - I)^{-1} \left(\frac{1}{u_n} \right) I$$

is the candidate recursion operator for (2.3).

STEP 3 (Determine the coefficients). Starting from (7.6), we use (2.32) with $F = u_n (u_{n+1} - u_{n-1})$ to compute $\mathcal{R}'[F]$. The partial results will not be shown due to length. Using (2.21), we compute

$$(7.7) \quad F' = -u_n D^{-1} + (u_{n+1} - u_{n-1}) I + u_n D.$$

Then we compute $\mathcal{R} \circ F'$ and $F' \circ \mathcal{R}$. After substituting the pieces into (2.30) we simplify the resulting expression using rules such as

$$(7.8) \quad (D - I)^{-1} D = D(D - I)^{-1} = I + (D - I)^{-1},$$

$$(7.9) \quad (D - I)^{-1} D^{-1} = D^{-1} (D - I)^{-1} = -D^{-1} + (D - I)^{-1}.$$

We further simplify by recursively using formulas like

$$(7.10) \quad D U(u_n) (D - I)^{-1} V(u_n) I = U(u_{n+1}) V(u_n) I + U(u_{n+1}) (D - I)^{-1} V(u_n) I,$$

$$(7.11) \quad (D - I)^{-1} U(u_n) V(u_n) D = U(u_{n-1}) V(u_{n-1}) I + (D - I)^{-1} U(u_{n-1}) V(u_{n-1}) I.$$

Finally, we equate like terms to obtain a linear system for the c_i . Substituting the solution

$$(7.12) \quad c_1 = c_3 = c_4 = c_7 = c_9 = 0, \quad c_2 = c_5 = c_6 = c_8 = c_{10} = 1$$

into (7.6) we obtain the final result

$$(7.13) \quad \mathcal{R} = u_n D^{-1} + (u_n + u_{n+1}) I + u_n D + u_n (u_{n+1} - u_{n-1}) (D - I)^{-1} \frac{1}{u_n} I.$$

A straightforward computation confirms that $\mathcal{R} G^{(1)} = G^{(2)}$ with $G^{(1)}$ in (2.25) and $G^{(2)}$ in (2.26).

8. Computation of matrix recursion operators

We construct the recursion operator (2.35) for (2.4). Now all the terms in (2.30) are 2×2 matrix operators.

STEP 1 (Determine the rank of the recursion operator). The difference in rank of symmetries is again used to compute the rank of the elements of the recursion operator. Using (2.7), (2.27) and (2.28),

$$(8.1) \quad \text{rank } \mathbf{G}^{(1)} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad \text{rank } \mathbf{G}^{(2)} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

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

$$(8.2) \quad \text{rank } \mathcal{R}_{ij} = \text{rank } G_i^{(k+1)} - \text{rank } G_j^{(k)},$$

to compute a rank matrix associated to the operator

$$(8.3) \quad \text{rank } \mathcal{R} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}.$$

STEP 2 (Determine the form of the recursion operator). As in the scalar case, we build a candidate \mathcal{R}_0 :

$$(8.4) \quad \mathcal{R}_0 = \begin{pmatrix} (\mathcal{R}_0)_{11} & (\mathcal{R}_0)_{12} \\ (\mathcal{R}_0)_{21} & (\mathcal{R}_0)_{22} \end{pmatrix},$$

with

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

Analogous to the scalar case, the elements of matrix \mathcal{R}_1 are linear combinations with constant coefficients \tilde{c}_{jk} of all suitable products of symmetries and covariants sandwiching $(\mathbf{D} - \mathbf{I})^{-1}$. Hence,

$$(8.5) \quad \sum_j \sum_k \tilde{c}_{jk} \mathbf{G}^{(j)} (\mathbf{D} - \mathbf{I})^{-1} \otimes \rho_n^{(k)'},$$

where \otimes denotes the matrix outer product, defined as

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

Only the pair $(\mathbf{G}^{(1)}, \rho_n^{(0)'})$ can be used, otherwise the ranks in (8.3) would be exceeded. Using (2.24) and (2.16) we compute

$$(8.6) \quad \rho_n^{(0)'} = \begin{pmatrix} 0 & \frac{1}{v_n} \mathbf{I} \end{pmatrix},$$

Therefore, using (8.5) and renaming \tilde{c}_{10} to c_{17} ,

$$(8.7) \quad \mathcal{R}_1 = \begin{pmatrix} 0 & c_{17} (v_{n-1} - v_n) (\mathbf{D} - \mathbf{I})^{-1} \frac{1}{v_n} \mathbf{I} \\ 0 & c_{17} v_n (u_n - u_{n+1}) (\mathbf{D} - \mathbf{I})^{-1} \frac{1}{v_n} \mathbf{I} \end{pmatrix}.$$

Adding (8.4) and (8.7) we obtain

$$(8.8) \quad \mathcal{R} = \begin{pmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \mathcal{R}_{21} & \mathcal{R}_{22} \end{pmatrix},$$

with

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

STEP 3 (Determine the coefficients). All the terms in (2.30) need to be computed. The strategy is similar to the scalar case, yet the computations are much more cumbersome. Omitting the details, the result is: $c_2 = c_5 = c_6 = c_7 = c_8 = c_{10} = c_{11} = c_{12} = c_{13} = c_{15} = 0$, $c_1 = c_3 = c_4 = c_9 = c_{14} = c_{16} = -1$, and $c_{17} = 1$. Substitution the constants into (8.8) gives

$$(8.9) \quad \mathcal{R} = \begin{pmatrix} -u_n \mathbf{I} & -D^{-1} - \mathbf{I} + (v_{n-1} - v_n)(D - \mathbf{I})^{-1} \frac{1}{v_n} \mathbf{I} \\ -v_n \mathbf{I} - v_n D & -u_{n+1} \mathbf{I} + v_n (u_n - u_{n+1})(D - \mathbf{I})^{-1} \frac{1}{v_n} \mathbf{I} \end{pmatrix}.$$

It is straightforward to verify that $\mathcal{R}G^{(1)} = G^{(2)}$ with $G^{(1)}$ in (2.27) and $G^{(2)}$ in (2.28).

9. More examples

EXAMPLE 9.1. The modified Volterra lattice [3, 13],

$$(9.1) \quad \dot{u}_n = u_n^2 (u_{n+1} - u_{n-1}),$$

has two non-polynomial densities $\rho_n^{(0)} = 1/u_n$ and $\rho_n^{(1)} = \ln(u_n)$, and infinitely many polynomial densities. The first two symmetries,

$$(9.2) \quad G^{(1)} = u_n^2 (u_{n+1} - u_{n-1}),$$

$$(9.3) \quad G^{(2)} = u_n^2 u_{n+1}^2 (u_n + u_{n+2}) - u_{n-1}^2 u_n^2 (u_{n-2} + u_n),$$

are linked by the recursion operator

$$(9.4) \quad \mathcal{R} = u_n^2 D^{-1} + 2u_n u_{n+1} \mathbf{I} + u_n^2 D + 2u_n^2 (u_{n+1} - u_{n-1})(D - \mathbf{I})^{-1} \frac{1}{u_n} \mathbf{I}.$$

EXAMPLE 9.2. The AL lattice (2.2) has infinitely many densities [8] and symmetries [9]. The recursion operator is of the form (8.8) with

$$\begin{aligned} \mathcal{R}_{11} &= P_n D^{-1} - u_n \Delta^{-1} v_{n+1} \mathbf{I} - u_{n-1} P_n \Delta^{-1} \frac{v_n}{P_n} \mathbf{I}, \\ \mathcal{R}_{12} &= -u_n u_{n-1} \mathbf{I} - u_n \Delta^{-1} u_{n-1} \mathbf{I} - u_{n-1} P_n \Delta^{-1} \frac{u_n}{P_n} \mathbf{I}, \\ \mathcal{R}_{21} &= v_n v_{n+1} \mathbf{I} + v_n \Delta^{-1} v_{n+1} \mathbf{I} + v_{n+1} P_n \Delta^{-1} \frac{v_n}{P_n} \mathbf{I}, \\ \mathcal{R}_{22} &= (u_n v_{n+1} + u_{n-1} v_n) \mathbf{I} + P_n D + v_n \Delta^{-1} v_{n-1} \mathbf{I} + v_{n+1} P_n \Delta^{-1} \frac{u_n}{P_n} \mathbf{I}, \end{aligned}$$

where $P_n = 1 + u_n v_n$ and $\Delta = D - \mathbf{I}$.

This recursion operator has an inverse which is quite exceptional. For further information see e.g. [25].

References

1. M. J. Ablowitz and P. A. Clarkson, *Solitons, nonlinear evolution equations and inverse scattering*, London Math. Soc. Lecture Note Ser. **149**, Cambridge University Press, Cambridge, U.K., 1991.
2. M. J. Ablowitz and J. F. Ladik, *Nonlinear differential-difference equations*, J. Math. Phys. **16** (1975), 598–603.
3. V. E. Adler, A. B. Shabat, and R. I. Yamilov, *Symmetry approach to the integrability problem*, Theoret. and Math. Phys. **125** (2000), 1603–1661.
4. D. Baldwin, W. Hereman, and J. Sayers, *Symbolic algorithms for the Painlevé test, special solutions and recursion operators for nonlinear PDEs*, this volume.
5. A. H. Bilge, *On the equivalence of linearization and formal symmetries as integrability tests for evolution equations*, J. Phys. A **26** (1993), 7511–7519.
6. Ü. Göktaş and W. Hereman, *Symbolic computation of conserved densities for systems of nonlinear evolution equations*, J. Symbolic Comput. **24** (1997), 591–621.
7. ———, **Mathematica package** *InvariantsSymmetries.m* is available since 1997 from <http://www.mathsource.com/cgi-bin/msitem?0208-932>.
8. ———, *Computation of conserved densities for nonlinear lattices*, Phys. D **123** (1998), 425–436.
9. ———, *Algorithmic computation of higher-order symmetries for nonlinear evolution and lattice equations*, Adv. Comput. Math. **11** (1999), 55–80.
10. 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.
11. W. Hereman and Ü. Göktaş, *Integrability tests for nonlinear evolution equations*, Computer Algebra Systems: A Practical Guide (M. Wester, ed.), Wiley, New York, 1999, pp. 211–232.
12. W. Hereman, Ü. Göktaş, M. D. Colagrosso, and A. J. Miller, *Algorithmic integrability tests for nonlinear differential and lattice equations*, Comput. Phys. Comm. **115** (1998), 428–446.
13. M. S. Hickman and W. A. Hereman, *Computation of densities and fluxes of nonlinear differential-difference equations*, Proc. Roy. Soc. Lond. Ser. A **459** (2003), 2705–2729.
14. P. E. Hydon and E. L. Mansfield, *A variational complex for difference equations*, Found. Comput. Math. **4** (2004), 187–217.
15. 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.
16. E. L. Mansfield and P. E. Hydon, *On a variational complex for difference equations*, The Geometrical Study of Differential Equations (Washington, DC, 2000) (J. A. Leslie and T. P. Robart, eds.), Proc. NSF-CBMS Conf., Comtemp. Math., vol. 285, Amer. Math. Soc., Providence, R.I., 2001, pp. 121–129.
17. E. L. Mansfield and G. R. W. Quispel, *Towards a variational complex for the finite element method*, this volume.
18. A. G. Meshkov, *Computer package for investigation of the complete integrability*, Proc. Inst. Maths. NAS Ukraine **30** (2000), 35–46.
19. W. Oevel, H. Zhang, and B. Fuchssteiner, *Mastersymmetries and multi-Hamiltonian formulations for some integrable lattice systems*, Progr. Theoret. Phys. **81** (1989), 294–308.
20. P. J. Olver, *Applications of lie groups to differential equations*, 2nd. ed., Grad. Texts in Math. **107**, Springer, New York, 1993.
21. J. A. Sanders and J. P. Wang, *Integrable systems and their recursion operators*, Nonlinear Anal. **47** (2001), 5213–5240.
22. ———, *On recursion operators*, Phys. D **149** (2001), 1–10.
23. M. Toda, *Theory of nonlinear lattices*, Springer, Berlin, 1981.
24. J. P. Wang, *Symmetries and conservation laws of evolution equations*, Ph.D. Thesis, Thomas Stieltjes Institute for Mathematics, Amsterdam, 1998.
25. H. Zhang, G. Tu, W. Oevel, and B. Fuchssteiner, *Symmetries, conserved quantities, and hierarchies for some lattice systems with soliton structure*, J. Math. Phys. **32** (1991), 1908–1918.

DEPARTMENT OF MATHEMATICAL AND COMPUTER SCIENCES, COLORADO SCHOOL OF MINES,
GOLDEN, COLORADO 80401-1887, USA.

E-mail address: `whereman@mines.edu`

DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCES, FREE UNIVERSITY, DE BOELELAAN
1081A, 1081 HV AMSTERDAM, THE NETHERLANDS

E-mail address: `jansa@cs.vu.nl`

PHYSICS DEPARTMENT 59-33, CALIFORNIA INSTITUTE OF TECHNOLOGY, PASADENA, CALIFOR-
NIA 91125

E-mail address: `jack@its.caltech.edu`

INSTITUTE OF MATHEMATICS, STATISTICS & ACTUARIAL SCIENCE, UNIVERSITY OF KENT, CAN-
TERBURY, CT2 7NF, UK.

E-mail address: `J.Wang@kent.ac.uk`