Symbolic Computation of Recursion Operators for Nonlinear Differential-Difference Equations

Ünal Göktas¹, Willy Hereman²

¹Department of Computer Engineering, Turgut Özal University, Keçiören, Ankara 06010, Turkey
²Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado 80401-1887, U.S.A.

E-mail addresses: unalgoktas@ttmail.com, whereman@mines.edu

ABSTRACT – An algorithm for the symbolic computation of recursion operators for systems of nonlinear differential-difference equations (DDEs) is presented. Recursion operators allow one to generate an infinite sequence of generalized symmetries. The existence of a recursion operator therefore guarantees the complete integrability of the DDE. The algorithm is based in part on the concept of dilation invariance and uses our earlier algorithms for the symbolic computation of conservation laws and generalized symmetries.

The algorithm has been applied to a number of well-known DDEs, including the Kac-van Moerbeke (Volterra), Toda, and Ablowitz-Ladik lattices, for which recursion operators are shown. The algorithm has been implemented in Mathematica, a leading computer algebra system. The package DDERecursionOperator.m is briefly discussed.

Keywords: Conservation law, generalized symmetry, recursion operator, nonlinear differential-difference equation

I. INTRODUCTION

A number of interesting problems can be modeled with nonlinear differential-difference equations (DDEs) [1]-[3], including particle vibrations in lattices, currents in electrical networks, and pulses in biological chains. Nonlinear DDEs also play a role in queuing problems and discretizations in solid state and quantum physics, and arise in the numerical solution of nonlinear PDEs.

The study of complete integrability of nonlinear DDEs largely parallels that of nonlinear partial differential equations (PDEs) [4]-[7]. Indeed, as in the continuous case, the existence of large numbers of generalized (higher-order) 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 higher-order symmetries of the equation into new higher-order symmetries. The existence of a recursion operator, which allows one to generate an infinite set of such symmetries step-by-step, then confirms complete integrability.

There is a vast body of work on the complete integrability of DDEs. Consult, e.g., [5, 8] for additional references. In this article we describe an algorithm to symbolically compute recursion operators for DDEs. The design of this algorithm relies on our related work for PDEs and DDEs [9]-[11] and seminal work by Oevel et al [12] and Zhang et al [13].

In contrast to the general symmetry approach in [5], our algorithms rely on specific assumptions. For example, we will use the dilation invariance of DDEs in the construction of densities, higher-order symmetries, and recursion operators. At the cost of generality, our algorithms can be implemented in major computer algebra systems.


The paper is organized as follows. In Section II, we present key definitions, necessary tools, and prototypical examples which will be used throughout the paper. The examples include the Kac-van Moerbeke (KvM) [16] and Toda [17] lattices. An algorithm for the computation of recursion operators is outlined in Section III. Usage of our package is demonstrated on an example in Section IV. The paper concludes with two more examples in Section V, namely the Ablowitz-Ladik (AL) [18] and Relativistic-Toda (RT) [19] lattices.

II. KEY DEFINITIONS

A. Definition

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

\[ \dot{u}_n = F(u_{n-1}, u_n, u_{n+1}, \ldots) \]

where \( u \) and \( F \) are vector-valued functions with \( N \) components. The subscript \( n \) corresponds to the label of the discretized space variable; the dot denotes differentiation with respect to the continuous time variable \( t \). Throughout
the paper, for simplicity we denote the components of \( \mathbf{u}_n \) by \((u_n, v_n, w_n, \ldots)\) 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. If 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 \( \mathbf{F} \).

B. Definition

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

C. Example

The Kac-van Moerbeke (KvM) lattice \([16]\),

\[
\dot{u}_n = u_n(u_{n+1} - u_{n-1}),
\]

(2)
arises in the study of Langmuir oscillations in plasmas, population dynamics, etc. Equation (2) is invariant under \((t, u_n) \rightarrow (\lambda^{-1}t, \lambda u_n)\), where \( \lambda \) is an arbitrary scaling parameter.

D. Example

The Toda lattice \([17]\) in polynomial form \([20]\),

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

(3)
models the vibration of masses in a lattice with exponential interaction force. Equation (3) is invariant under the scaling symmetry

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

(4)

E. Definition

We define the weight, \( w \), of a variable as the exponent in the scaling parameter \( \lambda \) which multiplies the variable.

As a result of this definition, \( t \) is always replaced by \( \frac{t}{\lambda} \) and \( w(d/dt) = w(D) = 1 \). In view of (3), we have \( w(u_n) = 1 \), and \( w(v_n) = 2 \) for the Toda lattice.

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

F. Definition

The rank 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.

G. Example

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. Indeed,

\[
w(u_n) + 1 = w(v_n), \quad w(v_n) + 1 = w(u_n) + w(v_n),
\]

(5)
yields

\[
w(u_n) = 1, \quad w(v_n) = 2,
\]

(6)
which is consistent with (4).

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.

H. Definition

A scalar function \( \rho_n(u_n) \) is a conserved density of (1) if there exists a scalar function \( J_n(u_n) \), called the associated flux, such that \([21]\)

\[
D_n \rho_n + \Delta J_n = 0
\]

(7)
is satisfied on the solutions of (1).

In (7), we used the (forward) difference operator,

\[
\Delta J_n = (D - 1) J_n = J_{n+1} - J_n,
\]

(8)
where \( D \) denotes the up-shift (forward or right-shift) operator, \( D J_n = J_{n+1} \), and 1 is the identity operator.

The operator \( \Delta \) 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 \( u_n \).

I. Example

The first three density-flux pairs \([11]\) for (2) are

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

(9)
\[
\rho_n^{(1)} = u_n, \quad J_n^{(1)} = u_n u_{n-1},
\]

(10)
\[
\rho_n^{(2)} = \frac{1}{2} u_n^2 + u_n u_{n-1}, \quad J_n^{(2)} = u_{n-1} u_n (u_n + u_{n+1}).
\]

(11)

J. Example

The first four density-flux pairs \([20]\) for (3) are

\[
\rho_n^{(0)} = \ln(v_n), \quad J_n^{(0)} = v_n,
\]

(12)
\[
\rho_n^{(1)} = u_n, \quad J_n^{(1)} = v_{n-1},
\]

(13)
\[
\rho_n^{(2)} = \frac{1}{2} u_n^2 + v_n, \quad J_n^{(2)} = u_n v_{n-1},
\]

(14)
\[
\rho_n^{(3)} = \frac{1}{3} u_n^3 + u_n (v_{n-1} + v_n), \quad J_n^{(3)} = u_{n-1} u_n v_{n-1} + v_n^2.
\]

(15)
The densities in (12)-(15) are uniform of ranks 0 through 3, respectively. The corresponding fluxes are also uniform in rank with ranks 1 through 4, respectively. In general, if in (7) \( \rho_n = R \) then rank \( J_n = R + 1 \), since \( w(D) = 1 \). All the pieces in (7) are uniform in rank. This comes as no surprise since the conservation law (7) holds on solutions of (1), hence it ‘inherits’ the dilation symmetry of (1).

Consult \([9]\) for our algorithm to compute polynomial conserved densities and fluxes, where we use (3) to illu-
strate the steps. Non-polynomial densities (which are rare and usually easy to find by hand) can be computed with the method given in [8].

K. Definition

A vector function $\mathbf{G}(\mathbf{u}_n)$ is called a generalized (higher-order) symmetry of (1) if the infinitesimal transformation $\mathbf{u}_n \rightarrow \mathbf{u}_n + \varepsilon \mathbf{G}$ leaves (1) invariant up to order $\varepsilon$. Consequently, $\mathbf{G}$ must satisfy [21]

$$D_t \mathbf{G} = F'(\mathbf{u}_n) \mathbf{G}$$

(16)
on solutions of (1). $F'(\mathbf{u}_n) \mathbf{G}$ is the Fréchet derivative of $F$ in the direction of $\mathbf{G}$.

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

$$F'(u_n) G = \frac{\partial}{\partial \varepsilon} F(u_n + \varepsilon \mathbf{G})|_{\varepsilon = 0} = \sum_k \frac{\partial F}{\partial u_{nk}} D^k G_k,$$

(17)

which defines the Fréchet derivative operator

$$F'(u_n) = \sum_k \frac{\partial F}{\partial u_{nk}} D^k.$$

(18)

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

$$F'(u_n) = \sum_k \frac{\partial F}{\partial u_{nk}} D^k G_k.$$

(19)

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

$$F'(\mathbf{u}_n) \mathbf{G} = \sum_k \frac{\partial F}{\partial u_{nk}} D^k G_1 + \sum_k \frac{\partial F}{\partial v_{nk}} D^k G_2, i = 1, 2.$$

(20)

In (17) and (20) the summation is over all positive and negative shifts (including the term without shift $k = 0$).

For $k > 0$, $D^k = D \circ D \circ \ldots \circ D$ ($k$ times). Similarly, for $k < 0$ the down-shift operator $D^{-1}$ is applied repeatedly. The generalization of (19) to a system with $N$ components should be obvious.

L. Example

The first two symmetries [11] of (2) are

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

(21)

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

(22)

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

M. Example

The first two non-trivial symmetries [11] of (3) are

$$G^{(1)} = \left( \begin{array}{c} v_{n-1} \\ v_n \\ v_{n+1} \end{array} \right) = \left( \begin{array}{c} v_n - v_{n-1} \\ v_n (u_{n+1} - u_n) \end{array} \right),$$

(23)

$$G^{(2)} = \left( \begin{array}{c} v_n (u_{n+1} + u_{n+2} - u_n) - v_{n+1} (u_{n+1} + u_n) \\ v_n (u_{n+2} - u_n + v_{n+1} - v_n) \end{array} \right).$$

(24)

The above symmetries are uniform in rank. For example, rank $G^{(2)}_1 = 3$ and rank $G^{(2)}_2 = 4$. The symmetries of lower ranks are trivial.

An algorithm to compute polynomial generalized symmetries is described in detail in [22].

III. COMPUTATION OF RECURSION OPERATORS

A. Definition

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

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

(25)

where $j = 1, 2, \ldots$, 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 $\mathcal{R}$ [6, 21] is

$$D_t \mathcal{R} + [\mathcal{R}, F'(\mathbf{u}_n)] \frac{\partial \mathcal{R}}{\partial \varepsilon} + [\mathcal{R} \circ F'(\mathbf{u}_n) - F'(\mathbf{u}_n) \circ \mathcal{R}, 0] = 0,$$

(26)

where the bracket $[\cdot, \cdot]$ denotes the commutator of operators and $\circ$ the composition of operators. The operator $F'(\mathbf{u}_n)$ was defined in (19). $\mathcal{R}[\mathbf{F}]$ 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

$$\mathcal{R} = U(\mathbf{u}_n) O((D - I)^{-1}, D^{-1}, I, D) V(\mathbf{u}_n),$$

(27)

and in that case

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

(28)

For the vector case, for the examples we studied, the elements of the $N \times N$ operator matrix $\mathcal{R}$ are of the form

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

Thus, for the two-component case

$$\mathcal{R}[\mathbf{F}]_{ij} = \sum_k (D^k \mathcal{F}_i) \frac{\partial U_{ij}}{\partial u_{nk}} O_{ij} V_{kj} + \sum_k U_{ij} O_{ij} (D^k \mathcal{F}_i) \frac{\partial V_{kj}}{\partial u_{nk}} + \sum_k U_{ij} O_{ij} (D^k \mathcal{F}_i) \frac{\partial V_{kj}}{\partial u_{nk}}.$$

(29)
B. Example

The KvM lattice (2) has recursion operator [7]

\[ \mathcal{R} = u_n (I + D)(u_T D - D^t u_n)(D - I)^{-1} \frac{1}{u_n} I \]

\[ = u_n D^t + (u_n + u_{n+1}) I + u_n D \]

\[ + u_n (u_n - u_{n-1}) (D - I)^{-1} \frac{1}{u_n} I. \]

(30)

C. Example

The Toda lattice (3) has recursion operator [7]

\[ \mathcal{R} = \begin{pmatrix} u_n & I \\
D^t + I + (v_n - v_{n-1})(D - I)^{-1} \frac{1}{v_n} I \\
v_n I + v_n D & u_n, I + v_n u_n - u_{n+1} (D - I)^{-1} \frac{1}{v_n} I \end{pmatrix}. \]

(31)

D. Algorithm for computation of recursion operators

We will now construct the recursion operator (31) for (3). In this case all the terms in (29) are 2 x 2 matrix operators. The construction uses the following steps:

Step 1 (Determine the rank of the recursion operator): The difference in rank of symmetries is used to compute the rank of the elements of the recursion operator.

Using (6), (23) and (24),

\[ \text{rank } \mathcal{G}^{(1)} = \frac{2}{3}, \text{ rank } \mathcal{G}^{(2)} = \frac{3}{4}. \]

(32)

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

\[ \text{rank } \mathcal{R} = \text{rank } \mathcal{G}^{(1,2)} - \text{rank } \mathcal{G}^{(4)}. \]

(33)

to compute a rank matrix associated to the operator \( \mathcal{R} \)

\[ \text{rank } \mathcal{R} = \begin{pmatrix} 1 & 0 \\
2 & 1 \end{pmatrix}. \]

(34)

Step 2 (Determine the form of the recursion operator): \( \mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 \) where \( \mathcal{R}_0 \) is a sum of terms involving \( D^t, I, \) and \( D. \) The coefficients of these terms are admissible power combinations of \( u_n, u_{n+1}, v_n, \) and \( v_n \) (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 \( u_n \), and the maximum up-shift in the next symmetry is \( u_{n+p}, \) then the associated piece that goes into \( \mathcal{R}_0 \) must have \( D, D^t, ..., D^t \). The same line of reasoning determines the minimum down-shift operator to be included. So, in our case

\[ \mathcal{R}_0 = \begin{pmatrix} (\mathcal{R}_0)^{11} & (\mathcal{R}_0)^{12} \\
(\mathcal{R}_0)^{21} & (\mathcal{R}_0)^{22} \end{pmatrix}, \]

(35)

with

\[ (\mathcal{R}_0)^{11} = (c_1 u_n + c_2 u_{n+1}) I, \]

\[ (\mathcal{R}_0)^{12} = c_3 D^t + c_4 I, \]

\[ (\mathcal{R}_0)^{21} = (c_5 u_n^2 + c_6 u_n u_{n+1} + c_7 u_{n+1}^2 + c_8 v_n + c_9) I \]

\[ + (c_{10} u_n^2 + c_{11} u_n u_{n+1} + c_{12} u_{n+1}^2 + c_{13} v_n) D, \]

\[ (\mathcal{R}_0)^{22} = (c_{14} u_n + c_{15} u_{n+1}) I. \]

(36)

As in the previous case [10], \( \mathcal{R}_1 \) is a linear combination (with constant coefficients \( \mathring{c}_{kj} \)) of sums of all suitable products of symmetries and covariants (Fréchet derivatives of conserved densities) sandwiching \( (D - I)^{-1} \). Hence,

\[ \sum_{k} \sum_{\ell} \mathring{c}_{kj} \mathcal{G}^{(j)} (D - I)^{-1} \otimes \rho_n^{(k)}, \]

(37)

where \( \otimes \) denotes the matrix outer product, defined as

\[ \begin{pmatrix} G^{(j)}_1 \\
G^{(j)}_2 \end{pmatrix} (D - I)^{-1} \otimes \begin{pmatrix} \rho_n^{(k)} \\\n\rho_n^{(k)} \end{pmatrix} = \begin{pmatrix} G^{(j)}_1 (D - I)^{-1} \rho_n^{(k)} & G^{(j)}_2 (D - I)^{-1} \rho_n^{(k)} \\
G^{(j)}_1 (D - I)^{-1} \rho_n^{(k)} & G^{(j)}_2 (D - I)^{-1} \rho_n^{(k)} \end{pmatrix}. \]

(38)

Only the pair \( (\mathcal{G}^{(1)}, \rho_n^{(0)}) \) can be used, otherwise the ranks in (34) would be exceeded. Using (12) and (20), we compute

\[ \rho_n^{(0)} = \begin{pmatrix} 0 \\
1/v_n I \end{pmatrix}. \]

(39)

Therefore, using (37) and renaming \( \mathring{c}_{16} \) to \( c_{17} \),

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

(40)

Adding (35) and (40), we obtain

\[ \mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 = \begin{pmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\
\mathcal{R}_{21} & \mathcal{R}_{22} \end{pmatrix}. \]

(41)

Step 3 (Determine the unknown coefficients): All the terms in (29) need to be computed. Referring to [7] for details, the result is:

\[ c_1 = c_5 = c_6 = c_7 = c_8 = c_{10} = c_{11} = c_{12} = c_{13} = c_{14} = 0, \]

\[ c_4 = c_9 = c_{15} = c_{16} = 1, \text{ and } c_{17} = -1. \]

(42)

Substituting these constants into (41) finally gives

\[ \mathcal{R} = \begin{pmatrix} u_n I & D^t + I + (v_n - v_{n-1}) (D - I)^{-1} \frac{1}{v_n} I \\
v_n I + v_n D & u_n, I + v_n (u_n - u_{n+1}) (D - I)^{-1} \frac{1}{v_n} I \end{pmatrix}. \]

(43)

It is easy to verify that \( \mathcal{R} \mathcal{G}^{(1)} = \mathcal{G}^{(2)} \) with \( \mathcal{G}^{(1)} \) in (23) and \( \mathcal{G}^{(2)} \) in (24).
IV. THE MATHEMATICA PACKAGE

To use the code, first load the `Mathematica` package `DDERecursionOperator.m` using the command

```mathematica
In[2]:= << DDERecursionOperator.m;
```

Proceeding with the Kiviat lattice (2) as an example, call the function `DDERecursionOperator` (which is part of the package):

```mathematica
In[3]:= DDERecursionOperator [
    {D[u[n, t], t] - u[n, t]* (u[n + 1, t] - u[n - 1, t]) == 0},
    {u}, {n, t}]
```

Weight :: dilation : Dilation symmetry of the equation(s) is `{Weight[t] -> -1, Weight[u] -> 1}`.

```mathematica
Out[3]= {{\{DiscreteShift[#1, {n, -1}]\} u[n, t] + DiscreteShift[#1, {n, 1}] u[n, t],
         + u[1 + n, t] + \Delta^2 n [\#1/u[n, t], \{n, t\}] [(-1 + n, t] u[n, t]
         + u[n, t][u[1 + n, t] & \}]
```

Here \( \Delta^2_n = (D - I)^{-1} \). The first part of the output (which we assign to \( \text{R} \) for later use) is indeed the recursion operator given in (30).

```mathematica
In[4]:= R = First[\%];
```

Now using the first symmetry, generate the next symmetry by calling the function `GenerateSymmetries` (which is also part of the package):

```mathematica
In[5]:= firstsymmetry = \{u[n, t][u[n + 1, t] - u[n - 1, t]]\};
In[6]:= GenerateSymmetries[R, firstsymmetry, 1][1];
```

```mathematica
         + u[n, t][u[1 + n, t] & \}]
```

Evaluating the next five symmetries starting from the first one, can be done as follows:

```mathematica
In[7]:= TableForm[
    GenerateSymmetries[R, firstsymmetry, 5]
]
```

Due to the length of the output we do not show that result here. The `Mathematica` function `TableForm` will nicely reformat the output in a tabular form. Our package is available at [15].

V. CONCLUDING EXAMPLES

A. Ablowitz-Ladik (AL) Lattice

The AL lattice [18]

\[
\dot{u}_n = (u_{n+1} - 2u_n + u_{n-1}) + u_n v_n (u_{n+1} + u_{n-1}),
\]

\[
\dot{v}_n = -(v_{n+1} - 2v_n + v_{n-1}) - u_n v_n (v_{n+1} + v_{n-1}),
\]

is an integrable discretization of the nonlinear Schrödinger (NLS) equation. The two recursion operators [7] computed by our package are:

\[
\mathcal{R}^{(1)} = \left( \begin{array} {c c} \mathcal{R}^{(1)}_{11} & \mathcal{R}^{(1)}_{12} \\ \mathcal{R}^{(1)}_{21} & \mathcal{R}^{(1)}_{22} \end{array} \right),
\]

with

\[
\mathcal{R}^{(1)}_{11} = P_n D^{-1} - u_n \Delta^2 v_n I - u_{n-1} P_n \Delta^2 v_n I,
\]

\[
\mathcal{R}^{(1)}_{12} = -u_n u_{n-1} I - u_n \Delta^2 u_{n+1} I - u_{n-1} P_n \Delta^2 u_{n-1} I,
\]

\[
\mathcal{R}^{(1)}_{21} = v_n v_{n+1} I + v_n \Delta^2 v_{n+1} I + v_{n+1} P_n \Delta^2 v_n I,
\]

\[
\mathcal{R}^{(1)}_{22} = (u_n v_{n+1} + u_{n-1} v_n) I + P_n D + v_n \Delta^2 u_{n-1} I
\]

and

\[
\mathcal{R}^{(2)} = \left( \begin{array} {c c} \mathcal{R}^{(2)}_{11} & \mathcal{R}^{(2)}_{12} \\ \mathcal{R}^{(2)}_{21} & \mathcal{R}^{(2)}_{22} \end{array} \right),
\]

with

\[
\mathcal{R}^{(2)}_{11} = -P_n D - u_n \Delta^2 v_n I - (u_{n+1} v_{n-1} + u_{n+1} v_n) I
\]

\[
- u_{n+1} P_n \Delta^2 v_n I,
\]

\[
\mathcal{R}^{(2)}_{12} = -u_n u_{n+1} I - u_n \Delta^2 u_{n+1} I - u_{n+1} P_n \Delta^2 u_n I,
\]

\[
\mathcal{R}^{(2)}_{21} = v_n v_{n+1} I + v_n \Delta^2 v_{n+1} I + v_{n+1} P_n \Delta^2 v_n I,
\]

\[
\mathcal{R}^{(2)}_{22} = -P_n D - v_n \Delta^2 u_{n-1} I + v_{n+1} P_n \Delta^2 u_n I,
\]

where

\( P_n = 1 + u_n v_n \) and \( \Delta = D - I \).

B. Relativistic Toda (RT) Lattice

The RT lattice [19] is given as

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

and the recursion operator found by our package coincides with the one in [19]:

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

This material is based upon work supported by the National Science Foundation (U.S.A.) under Grant Nos. CCF-0830783 and CCR-9901929. J. A. Sanders, J.-P. Wang, M. Hickman and B. Deconinck are gratefully acknowledged for valuable discussions.

REFERENCES


BIOGRAPHIES

Ünal Göktaş – Dr. Göktaş holds a BSc. degree (1993) in mathematics from Boğaziçi University (Turkey), and MSc. (1996) and Ph.D. (1998) degrees in mathematical and computer sciences from the Colorado School of Mines (U.S.A.).

His research focuses on integrability tests of nonlinear partial differential equations and nonlinear semi-discrete lattices. He also works on symbolic solution methods of differential and difference equations, symbolic summation methods and various other aspects of symbolic computation.

Dr. Göktaş was employed by Wolfram Research, Inc. (makers of Mathematica) as a senior kernel developer. He recently joined Turgut Ozal University (Turkey).

Willy Hereman – Prof. Hereman received his Bachelor’s (1974), Master’s (1976) and Ph.D. (1982) degrees in applied mathematics from the University of Ghent, Belgium. He held NATO Research Fellowships at the Department of Electrical and Computer Engineering of The University of Iowa (1983-84, 1985-86). After a three-year Van Vleck Assistant Professorship at the Department of Mathematics of the University of Wisconsin-Madison, he joined the Colorado School of Mines in 1989 where he is Professor of Mathematical and Computer Sciences.

He has published over ninety research papers in acousto-optics, scattering theory, soliton theory, nonlinear wave phenomena, wavelets, and symbolic methods for nonlinear partial differential equations and lattices. Supported by the National Science Foundation of the U.S.A., he has developed several methods and symbolic software in Mathematica for the investigation of integrability, symmetries, conservation laws, and exact solutions of nonlinear PDEs, differential-difference equations, and difference equations.

Prof. Hereman is a laureate of the Royal Academy of Sciences of Belgium, and a member of the American Mathematical Society, the Society of Industrial and Applied Mathematics, and the Mathematical Association of America.