

Computing symmetries and recursion operators of evolutionary super-systems using the SsTools environment

Arthemy V. Kiselev ^a and Andrey O. Krutov ^b and Thomas Wolf ^c

^a *Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, P.O.Box 407, 9700 AK Groningen, The Netherlands.*

^b *Independent University of Moscow, Bolshoj Vlasjevskij Pereulok 11, 119002, Moscow, Russia.*

^c *Department of Mathematics and Statistics, Brock University, 500 Glenridge Avenue, St.Catharines, Ontario, Canada L2S 3A1*

Abstract

At a *very* informal but practically convenient level, we discuss the step-by-step computation of nonlocal recursions for symmetry algebras of nonlinear coupled boson-fermion $N = 1$ supersymmetric systems by using the SsTOOLS environment.

The principle of symmetry plays an important role in modern mathematical physics. The differential equations that constitute integrable models practically always admit symmetry transformations. The presence of symmetry transformation in a system yields two types of explicit solutions: those which are invariant under a transformation (sub)group and the solutions obtained by propagating a know solution by the same group. The recursion operator is a (pseudo)differential operator which maps symmetries of a given system to into symmetries of the same system. The recursion operators allow to obtain new symmetries for a given seed symmetry.

It is common for important equations of mathematical physics not to have local recursion operators other than the identity $\text{id}: \varphi \mapsto \varphi$. Instead, they often admit nonlocal recursions which involve integrations such as taking the inverse of the total derivative D_x with respect to the independent variable x . To describe such nonlocal structures we use the approach of nonlocalities. By nonlocalities we mean an extension of the initial system by new fields such that the initial fields are differential consequences of the new ones. In the case of recursion operators such fields often arise for conservation laws. We refer to a recursion operator for the Korteweg–de Vries equations as a motivating example of a nonlocal recursion operator, see Example 7 on page 11.

The supersymmetric integrable systems, i.e. systems involving commuting (bosonic, or even) and anticommuting (fermionic, or odd) independent variables and/or unknown functions, have found remarkable applications in modern mathematical physics (for example supergravity models, perturbed conformal field theory [12]; we refer to [1, 4] for a general overview). When dealing with supersymmetric models of theoretical physics, it is often hard to predict whether a certain mathematical

approximation will be truly integrable or not. Therefore we apply the symbolic computation to exhibit necessary integrability features. In what follows we restrict ourselves to the case of $N = 1$ (where N refers to the number of odd anticommuting independent variables θ_i). Nevertheless, the techniques and computer programs described below could be easily applied to the case of arbitrary N . Usually, the N is not bigger than 8, see for example [11] and [3]. It is an interesting open problem to establish criteria that set a limit on N in “ N -extended” supersymmetric equations of mathematical physics.

The latest version of SStools can be found at [16], see also [8]. We refer to [1, 4] and [2, 5, 6, 9, 13] for reviews of the geometry and supergeometry of partial differential equations. We refer to [10] for an overview of other software that could be used for similar computational tasks.

1 Notation and definitions

We fix notation first. Let x be the independent variable, $u = (u^1, \dots, u^m)$ denote the unknown functions irrespective of their (anti-)commutation properties, and $u_k = \{u_k^j | u_k^j = \partial^k u^j / \partial x^k\}$ denote the partial derivatives of u^j of order $k \in \mathbb{N}$. We extend the independent variable x by the pair (x, θ) , where θ is the Grassmann variable such that $\theta^2 = 0$. The superderivative is defined as

$$\mathcal{D} = \partial_\theta + \theta \partial_x.$$

Its square power is the spatial derivative, $\mathcal{D}^2 = \partial_x$.

Fields $u(x, t)$ now become $N = 1$ superfields $u(x, t; \theta)$. Provided that $\theta^2 = 0$, they have a very simple Taylor expansions in θ :

$$u(x, t; \theta) = u^0(x, t) + \theta u^1(x, t),$$

here u^0 has the same parity as u and u^1 has the opposite parity. The bosonic fields (those commuting with everything) are denoted by $b(x, t; \theta)$, and the fermionic fields, which anti-commute between themselves and with θ , will be denoted by $f(x, t; \theta)$. Further, we write $u_{k/2} = \mathcal{D}^k u$ for the k^{th} order super-derivative of u . Note that the super-derivatives $\mathcal{D}^{2k+1} b$ are fermionic and $\mathcal{D}^{2k+1} f$ are bosonic for any $k \in \mathbb{N}$.

Computer input will be shown in text font, for example, $\mathbf{f}(i)$ for f^i , $\mathbf{b}(j)$ for b^j , $\mathbf{d}\mathbf{f}(\mathbf{b}(j), \mathbf{x})$ for the derivative of b^j with respect to x , $\mathbf{d}(1, \mathbf{f}(i))$ for the superderivative $\mathcal{D}_{\theta^1} f^i$ written as $\mathcal{D} f^i$, because we will have only one θ and one \mathcal{D} .

Let $k, r < \infty$ be fixed integers. Suppose that $F^\alpha(x, t, u, \mathcal{D}(u), \dots, u_k, u_t)$ is a smooth function for any integer $\alpha \leq r$. In what follows, we consider systems of differential equations,

$$\{F^\alpha(x, u, \mathcal{D}(u), u_1, \mathcal{D}(u_1), \dots, u_k, u_t) = 0\}, \quad \alpha = 1, \dots, r, \quad (1)$$

of order k and, especially, the autonomous translation-invariant evolutionary systems,

$$\mathcal{E} = \{F^\alpha = u_t^\alpha - \Phi^\alpha(u, \dots, u_k) = 0\}, \quad \alpha = 1, \dots, r,$$

which are resolved w.r.t. the time derivatives, and the systems obtained by extending the evolutionary systems with some further differential relations upon u 's and their (super-)derivatives.

The weight technique

Physically meaningful equations have often several symmetries, among them one or more scaling symmetries. Suppose that to each super-field u^j and to the derivative ∂_t one can assign a real number (the weight), which is denoted by $[A]$ for any object A . By definition, $[\partial_x] \equiv 1$. The weight of a product of two objects is the sum of the weights of the factors, whence $[\mathcal{D}] = \frac{1}{2}$. The weight of any nonzero constant equals 0, but the weight of a zero-valued constant can be arbitrary.

From now on, we consider differential equations (1) with differential-polynomials F^α that admit the introduction of weights for all variables and derivations such that the weights of all monomials in each equation $F^\alpha = 0$ coincide. These equations are scaling-invariant, or homogeneous.

Example 1. Consider the Burgers equation

$$b_t = b_{xx} - 2bb_x, \quad b = b(x, t). \quad (2)$$

The weights are uniquely defined,

$$[b] = 1, \quad [\partial_t] = 2 \Leftrightarrow [t] = -2, \quad [\partial_x] \equiv 1.$$

Indeed, equation (2) is homogeneous w.r.t. these weights,

$$[b_t] = 1 + 2 = [b_{xx}] = 1 + 1 + 1 = [2bb_x] = 0 + 1 + 1 + 1 = 3$$

and, clearly, this is the only way to choose the weights.

A system of differential equations could be homogeneous w.r.t. to different weight systems. For a given system of differential equations these weight systems can be found by using the `FindSSWeight` function from SsTOOLS:

```
FindSSWeights(N,nf,nb,exli,zerwei,verbose)
where
```

```
N ... the number of superfields  $\theta_i$ ;
nf ... number of fermion fields  $f(1), f(2), \dots, f(nf)$ ;
nb ... number of boson fields  $b(1), b(2), \dots, b(nb)$ ;
exli ... list of equations or expressions;
zerwei ... list of constants or other kernels that should have zero weight;
verbose ... (=t(true)/nil(false)) whether detailed comments shall be made.
```

The program returns a list of homogeneities, each homogeneity being a list `fh`, `bh`, `hi` of

```
fh ... a list of the weights of  $f(1), b(2), \dots, f(nf)$ ;
```

bh ... a list of the weights of $b(1), b(2), \dots, b(nb)$;

hi ... a list of the weights of equations/expressions in the input.

Weights are scaled such that weight of $[\partial_x]$ is 2, i.e. the weight of any \mathcal{D} is 1. So the computer weights will be twice the “usual” weight. Input expressions can be in field form or coordinate form.

Example 2. Consider the nonlinear Schrödinger equation

$$\begin{aligned} b_t^1 &= b_{xx}^1 + 2(b^1)^2 b^2, & b^1 &= b^1(x, t), \\ b_t^2 &= -b_{xx}^2 - 2(b^2)^2 b^1, & b^2 &= b^2(x, t). \end{aligned}$$

We compute all possible weight systems of this system of equations:

```
FindSSWeights(0,0,2,{df(b(1),t) = df(b(1),x,2) + 2*b(1)**2*b(2),
                    df(b(2),t) = - df(b(2),x,2) - 2*b(2)**2*b(1) },
               {},t)$
```

The output contains

This system has the following homogeneities:

$$W[t] = -4$$

$$W[b(2)] = -\text{arbcomplex}(1) + 4$$

$$W[b(1)] = \text{arbcomplex}(1)$$

$$W[x] = -2$$

which gives us the following family of weight systems for (3)

$$[\partial_t] = 2, \quad [b^2] = -c + 4, \quad [b^1] = c, \quad [\partial_x] = 1,$$

where c is an arbitrary constant.

2 Symmetries

Definition 1. An l^{th} order symmetry of an evolutionary system $\mathcal{E} = \{u_t = \Phi\}$ is another autonomous evolutionary system $\mathcal{E}' = \{u_s = \varphi(x, u, \mathcal{D}(u), \dots, u_l)\}$ upon $u(s, t, x; \theta)$ such that a solution of the Cauchy problem for \mathcal{E}' propagates solutions of \mathcal{E} to solutions of \mathcal{E} . A necessary and sufficient condition for a vector φ to be a symmetry of \mathcal{E} is that solutions of the system satisfy

$$D_t D_s u = \pm D_s D_t u \tag{4}$$

where the minus sign applies if both t and s are fermionic. Because (4) is to be satisfied by solutions u of the system \mathcal{E} , i.e. u_t is replaced by Φ giving the so-called linearization $\text{Lin } \mathcal{E}$ of the system \mathcal{E} :

$$\text{Lin}_\Phi(\varphi) := D_t \varphi = \pm D_s \Phi.$$

This is a linear system for φ . If s is a bosonic variable then the symmetry is a system $\mathcal{E}' = \{f_s = F, b_s = B\}$ and if s is fermionic then the symmetry is $\mathcal{E}' = \{f_s = B, b_s = F\}$.

The understanding of linearized systems $\text{Lin } \mathcal{E}$ from a computational viewpoint is as follows; we consider the differential polynomial case since this is what SsTOOLS can be applied to. Let us first consider the case of bosonic s .

Given a system \mathcal{E} of super-equations, formally assign the new ‘linearized’ fields $F^i = \mathbf{f}(\mathbf{nf}+\mathbf{i})$ and $B^j = \mathbf{b}(\mathbf{nb}+\mathbf{j})$ to $f^i = \mathbf{f}(\mathbf{i})$ and $b^j = \mathbf{b}(\mathbf{j})$, respectively, with $i = 1, \dots, \mathbf{nf}$ and $j = 1, \dots, \mathbf{nb}$. Pass through all equations, and whenever a power of a derivative of a variable f^i or b^j is met, differentiate (in the usual sense) this power with respect to its base, multiply the result from the right by the same order derivative of F^i or B^j , respectively, and insert the product in the position where the power of the derivative was met. Now proceed by the Leibniz rule. The final result, when all equations in the system \mathcal{E} are processed, is the linearized system $\text{Lin } \mathcal{E}$.

If s is fermionic then proceed in the same way, except we get extra factors of -1 :

- An overall factor -1 appears in (4) if t is fermionic due to anticommuting D_t and D_s .
- When differentiating a factor in a product then a factor -1 appears for each fermionic factor to the left of the differentiated factor.
- When changing the order of D_s and \mathcal{D} in differentiating then a factor of -1 appears as well.

The second difference between bosonic and fermionic s is the number of new ‘linearized’ fields F^i, B^j that are introduced in the linearized equation. For bosonic s these are $F = \mathbf{f}(\mathbf{nf}+1) \dots \mathbf{f}(\mathbf{nf}+\mathbf{nf})$, $B = \mathbf{b}(\mathbf{nb}+1) \dots \mathbf{b}(\mathbf{nb}+\mathbf{nb})$ whereas for fermionic s these are $F = \mathbf{f}(\mathbf{nf}+1) \dots \mathbf{f}(\mathbf{nf}+\mathbf{nb})$, $B = \mathbf{b}(\mathbf{nb}+1) \dots \mathbf{b}(\mathbf{nb}+\mathbf{nf})$.

To summarize, the linearization is obtained by a complete differentiation $D_s \Phi$ applying the Leibniz rule and chain rule (in place) and substituting $u_s = \varphi$.

Example 3. The linearized counterpart of $b_t = b(\mathcal{D}f)^2$ is $B_t = B(\mathcal{D}f)^2 + 2b\mathcal{D}f\mathcal{D}F$ for bosonic symmetry parameter s and $F_t = F(\mathcal{D}f)^2 - 2b\mathcal{D}f\mathcal{D}B$ for fermionic \bar{s} . Likewise, for $b_t = bff_x(\mathcal{D}f)^2$ and parity-odd \bar{s} , the linearization is $F_t = Fff_x(\mathcal{D}f)^2 + bBf_x(\mathcal{D}f)^2 - bfB_x(\mathcal{D}f)^2 - 2bff_x\mathcal{D}f\mathcal{D}B$.

The scaling weights of the new fields are always set by $[F] = [f]$, $[B] = [b]$ for bosonic s and $[F] = [b]$, $[B] = [f]$ for fermionic \bar{s} .

The linearization $\text{Lin } \mathcal{E}$ for a system of evolution equations \mathcal{E} is obtained using the procedure `linearize`:

```
linearize(pdes, nf, nb, tpar, spar);
```

where

`pdes` ... list of equations $b_t^i = \Phi_b^i$, $f_t^j = \Phi_f^j$;

`nf` ... number of the fermion fields $\mathbf{f}(1), \mathbf{f}(2), \dots, \mathbf{f}(\mathbf{nf})$;

`nb` ... number of the boson fields $\mathbf{b}(1), \mathbf{b}(2), \dots, \mathbf{b}(\mathbf{nb})$;

`tpar` ... (=t(true)/nil(false)) whether t is parity changing or not

`spar` ... (=t(true)/nil(false)) whether s is parity changing or not

Example 4. The linearizations of the system with parity reversing time \bar{t} ,

$$f_{\bar{t}} = b^2 + \mathcal{D}f, \quad b_{\bar{t}} = fb + \mathcal{D}b, \quad (5)$$

are obtained as follows,

```
depend {b(1),f(1)},x,t;
linearize({df(f(1),t)=b(1)**2+d(1,f(1)),
          df(b(1),t)=f(1)*b(1)+d(1,b(1))},1,1,t,nil);
```

for bosonic s , and the same call with t as last parameter instead of nil for fermionic \bar{s} .

The result is the new system involving twice as many variables as the original equation. The linearization correspondence between the fields is

$$f \mapsto F \quad (f(1) \mapsto f(2)), \quad b \mapsto B \quad (b(1) \mapsto b(2))$$

for bosonic s and

$$f \mapsto B \quad (f(1) \mapsto b(2)), \quad b \mapsto F \quad (b(1) \mapsto f(2))$$

for fermionic \bar{s} .

The procedure to compute the linearization is the same for normal times t and for parity reversing times \bar{t} except of a factor (-1) of the rhs's B, F if both \bar{t}, \bar{s} are fermionic (because of the anticommutativity of $D_{\bar{t}}, D_{\bar{s}}$ in that case).

The linearized system incorporates

- the initial system:

$$\begin{aligned} df(f(1),t) &= b(1)**2 + d(1,f(1)), \\ df(b(1),t) &= f(1)*b(1) + d(1,b(1)), \end{aligned}$$
- and its linearizations:

$$\begin{aligned} df(f(2),t) &= 2*b(2)*b(1) + d(1,f(2)), \\ df(b(2),t) &= d(1,b(2)) + f(2)*b(1) + f(1)*b(2) \end{aligned}$$
 for s ; respectively,

$$\begin{aligned} df(b(2),t) &= d(1,b(2)) - 2*f(2)*b(1), \\ df(f(2),t) &= - b(2)*b(1) + d(1,f(2)) - f(2)*f(1) \end{aligned}$$
 for \bar{s} .

One does not need to compute the linearizations in order to obtain a symmetry of a differential equation. However, the explicit computation of the linearizations will be required for finding the recursions, which are “symmetries of symmetries.”

For computing symmetries of any system \mathcal{E} , use the procedure `ssym` with the call `ssym(N,tw,sw,afwlist,abwlist,eqnlist,fl,inelist,flags);`

N ... the number of superfields θ_i ;

tw ... $2 \times$ the weight of ∂_i ;

sw ... $2 \times$ the weight of ∂_s ;

afwlist ... list of $2 \times$ weights of the fermion fields $f(1), f(2), \dots, f(n_f)$;
abwlist ... list of $2 \times$ weights of the boson fields $b(1), b(2), \dots, b(n_b)$;
eqnlist ... list of extra conditions on the undetermined coefficients;
fl ... extra unknowns in **eqnlist** to be determined;
inelist ... a list, each element of it is a non-zero expression or a list with at least one of its elements being non-zero;
flags ... list of flags:
 init: only initialization of global data,
 zerocoeff: all coefficients = 0 which do not appear in **inelist**,
 tpar: if the time variable t changes parity,
 spar: if the symmetry variable s changes parity,
 lin: if symmetries of a linearization are to be computed,
 filter: if a symmetry should satisfy homogeneity weights defined in **hom_wei**

Note that the computer representation of the weights is twice the standard notation; thus we avoid half-integers values for convenience. For more details on other flags run the command **sshelp()**.

3 Recursions

Definition 2. A recursion operator for the symmetry algebra of an evolutionary system

$$\mathcal{E} = \{f_t^i = \Phi_f^i, b_t^j = \Phi_b^j\}, \quad i = 1, \dots, n_f, \quad j = 1, \dots, n_b$$

is the vector expression

$$\mathcal{R} = (\mathcal{R}_f^1(\varphi), \dots, \mathcal{R}_f^{n_f}(\varphi), \mathcal{R}_b^1(\varphi), \dots, \mathcal{R}_b^{n_b}(\varphi))^T,$$

which is linear w.r.t. the new fields φ , which for bosonic s are $\varphi = (F^1, \dots, F^{n_f}, B^1, \dots, B^{n_b})^T$ and for fermionic \bar{s} are $\varphi = (B^1, \dots, B^{n_f}, F^1, \dots, F^{n_b})^T$, and their derivatives and which is a (right-hand side of a) symmetry of \mathcal{E} whenever

$$\varphi = (F^1, \dots, F^{n_f}, B^1, \dots, B^{n_b})^T \quad \text{for } s, \text{ or}$$

$$\varphi = (B^1, \dots, B^{n_f}, F^1, \dots, F^{n_b})^T \quad \text{for } \bar{s}$$

is a symmetry of \mathcal{E} . In other words, $\mathcal{R}: \text{sym } \mathcal{E} \rightarrow \text{sym } \mathcal{E}$ is a linear operator that generates a symmetry of \mathcal{E} when applied to a symmetry $\varphi \in \text{sym } \mathcal{E}$.

The weight $[\mathcal{R}]$ of recursion \mathcal{R} is the difference $[\mathcal{R}(\varphi)] - [\varphi]$ of the weights of the (time derivatives $\partial_{s^t}, \partial_s$ of u , i.e. of the) resulting and the initial symmetries, here $\varphi \in \text{sym } \mathcal{E}$.

Different recursions can have the same weight. Experiments show that in this case operators may have different properties, e.g., a majority of them is nilpotent, several zero-order operators act through multiplication by a differential-functional

expression and do not increase the differential orders of the flows, and only few recursions construct higher-order symmetries and reveal the integrability.

If $[\mathcal{R}]$ is the weight of a recursion \mathcal{R} , then, clearly, at least one recursion is found with weight $2 \times [\mathcal{R}]$. Indeed, this is \mathcal{R}^2 . At the same time, other recursions may appear with weight $\mathbb{N} \times [\mathcal{R}]$. If \mathcal{R} is nonlocal (see below), then its powers are also nonlocal, but it remains a very delicate matter to predict the form of their nonlocalities, and strong theoretical assertions can be formulated for some particular integrable system.

Remark 1. The derivation of the weight of a recursion is constructive in the following sense. To attempt finding a recursion for an evolutionary system, it is beneficial to know already many symmetries φ_{s_i} of different weights $[s_i]$. Then one tries first the weights $[\mathcal{R}] := [s_i] - [s_j]$ for various i, j . However, the recursions obtained this way can be nilpotent, i.e. $\mathcal{R}^q(\varphi) \equiv 0$ for some q and any φ . More promising are weights for which there exist i, j, k with $[\mathcal{R}] = [s_i] - [s_j] = [s_j] - [s_k]$ and s_i, s_j, s_k being elements of the infinite hierarchies of symmetries with low weights. Still, the actual weights of unknown recursions can turn out to be larger than the weight differences of the lowest order symmetries.

Finally, non-trivial recursions may only appear in nonlocal settings. We discuss this in section 4.

The crucial point is that \mathcal{R} is a symmetry of the linearized system $\text{Lin } \mathcal{E}$. The original system \mathcal{E} is only used for substitutions. Hence we use `ssym` for finding recursions of the linearizations, which are previously calculated by `linearize`.

Example 5. Let us construct a recursion for equation (2). We obtain the linearization using the procedure `linearize`,

```
linearize({df(b(1),t)=df(b(1),x,2)-2*b(1)*df(b(1),x)},0,1,nil,nil);
```

The new system depends on the two fields $b = b(1)$ and $B = b(2)$.

```
df(b(2),t) = -2*b(2)*df(b(1),x)-2*b(1)*df(b(2),x)+df(b(2),x,2),
df(b(1),t) = df(b(1),x,2)-2*b(1)*df(b(1),x).
```

The recursion of weight 0 is obtained as follows (thus $\text{sw} = 2[\mathcal{R}] = 0$).

```
ssym(1, 4, 0, {}, {2, 2},
      {df(b(2),t)= -2*b(2)*df(b(1),x)-2*b(1)*df(b(2),x)+df(b(2),x,2),
        df(b(1),t)=> df(b(1),x,2)-2*b(1)*df(b(1),x)}, {}, {}, {lin});
```

By writing `df(b(1),t)=> ...` we require that the original system is only used for substitutions.

The output contains

```
df(b(2),s)=b(2)
1 solution was found.
```


This is the identity transformation

$$\varphi \mapsto \mathcal{R} = \text{id}(\varphi) \equiv \varphi;$$

it maps symmetries to themselves. Clearly, the identity is a recursion for any system!

Example 6 (A recursion for the Korteweg–de Vries equation). The KdV equation upon the bosonic field $u(x, t)$ is

$$u_t = -u_{xxx} + uu_x. \tag{6}$$

Equation (6) is homogeneous w.r.t. the weights

$$[u] = 2, \quad [t] = -3, \quad [x] = -1.$$

The linearization of (6) is constructed using the procedure `linearize`,

```
linearize({df(b(1),t)= -df(b(1),x,3) + b(1)*df(b(1),x)},0,1,nil,nil)
```

Thus we obtain the new system that depends on the fields $u = b(1)$, $U = b(2)$:

$$\begin{aligned} \text{df}(b(1),t) &= -\text{df}(b(1),x,3) + b(1)*\text{df}(b(1),x) \\ \text{df}(b(2),t) &= b(2)*\text{df}(b(1),x) + b(1)*\text{df}(b(2),x) - \text{df}(b(2),x,3). \end{aligned}$$

As a first guess we are looking for the recursion operator of weight 0. The recursion of weight 0 (hence $\text{sw} = -2[\mathcal{R}] = 0$) is constructed as follows.

```
ssym(1, 6, 0, {}, {4, 4},
      {df(b(2),t) = b(2)*df(b(1),x)+b(1)*df(b(2),x)-df(b(2),x,3),
        df(b(1),t) => -df(b(1),x,3)+b(1)*df(b(1),x)}, {}, {}, {lin});
```

The output contains

```
df(b(2),s)=b(2)
1 solution was found.
```

Again, this operator \mathcal{R} of weight 0 is the identity,

$$\varphi \mapsto \mathcal{R}(\varphi) = (\text{id})(\varphi).$$

The well-known explanation for this result is that, as a rule, one needs to introduce nonlocalities first and only then obtains nontrivial recursions in the nonlocal setting.

4 Nonlocalities

The nonlocal variables for $N = 1$ super-systems are constructed by trivializing [7, 9] conservation laws

$$\partial_t(\text{density}) \doteq \mathcal{D}(\text{super-flux}),$$

that is, in each case the above equality holds by virtue (\doteq) of the system at hand and all possible differential consequences from it. The standard procedure [9] suggests that every conserved current determines the new nonlocal variable, say v , whose derivatives are set to

$$v_t = \text{super-flux}, \quad \mathcal{D}v = \text{density} \tag{7a}$$

if the time t preserves the parities and

$$v_{\bar{t}} = -\text{super-flux}, \quad \mathcal{D}v = \text{density} \tag{7b}$$

if the time \bar{t} is parity-reversing. Note that in the classical case the nonlocality v can be specified through

$$v_t = \text{flux}, \quad v_x = \text{density} \tag{7c}$$

for the conservation law $\partial_t(\text{density}) \doteq \partial_x(\text{flux})$. Each nonlocality thus makes the conserved current trivial because the cross derivatives of v coincide in this case, $[D_t, \mathcal{D}](v) = 0$, where $[,]$ stands for the commutator if t is parity-preserving and for the *anti*commutator whenever the time \bar{t} is parity-reversing. The new variables can be bosonic or fermionic; the parities are immediately clear from the formulae for their derivatives.

Hence, starting with an equation \mathcal{E} , one calculates several conserved currents for it and *trivializes* them by introducing a *layer* of nonlocalities whose derivatives are still local differential functions. This way the number of fields is increased and the system is extended by new substitution rules. Moreover, it may acquire new conserved currents that depend on the nonlocalities and thus specify the second layer of nonlocal variables with nonlocal derivatives. At each step the number of variables will increase by 2 compared with the previous layer (a new nonlocal variable plus the corresponding linearized field). Clearly, the procedure is self-reproducing.

So, one keeps computing conserved currents and adding the layers of nonlinearities until an extended system $\tilde{\mathcal{E}}$ is achieved such that its linearization $\text{Lin } \tilde{\mathcal{E}}$ has a symmetry \mathcal{R} ; this symmetry of $\text{Lin } \tilde{\mathcal{E}}$ is a recursion for the extended system $\tilde{\mathcal{E}}$.

The calculation of conservation laws for evolutionary super-systems with homogeneous polynomial right-hand sides is performed by using the procedure `ssconl`:

```
ssconl(N,tw,mincw,maxcw,afwlist,abwlist,pdes);
```

where

N ... the number of superfields θ^i ;

`tw` ... 2×the weight $[\partial_t]$;
`mincw` ... minimal weight of the conservation law;
`maxcw` ... maximal weight of the conservation law;
`afwlist` ... list of weights of the fermionic fields $f(1), \dots, f(nf)$;
`abwlist` ... list of weights of the bosonic fields $b(1), \dots, b(nb)$;
`pdes` ... list of the equations for which a conservation law must be found.

For positive weights of bosonic variables, the ansatz is fully determined through the weight `mincw`, ..., `maxcw` of the conservation law. If a boson weight is non-positive then the global variable `max_deg` must have a positive integer value which is the highest degree of such a variable or any of its derivatives in any ansatz. The conservation law condition leads to an algebraic system for the undetermined coefficients, which is further solved automatically by `CRACK`.

Having obtained a conserved current, one defines the new bosonic or fermionic dependent variable (the nonlocality) using the standard rules (7).

We illustrate the general scheme of fixing the derivatives of a nonlocal variable by several examples. Further information on the `SSTOOLS` environment is contained in [8] and the `sshelp()` function in `SSTOOLS`. The algebraic structures that describe the geometry of recursion operators for super-PDE are described in detail in [9]. Some more examples and their applications are also found in [7].

Example 7 (A nonlocal recursion for the KdV equation). Consider the Korteweg–de Vries equation (6) again,

$$\partial_t(u) \doteq \partial_x \left(-u_{xx} + \frac{1}{2}u^2 \right).$$

We declare that the conserved density u is the spatial derivative $w_x = u$ of a new nonlinear variable w and the flux is its derivative w.r.t. the time, $w_t = -u_{xx} + \frac{1}{2}u^2$. Then $w_{xt} = w_{tx}$ by virtue of (6). Thus we introduce the bosonic nonlocality w by trivializing the conserved current. Let us remember that

$$\begin{aligned} w_x &= u, \\ w_t &= -w_{xxx} + \frac{1}{2}w_x^2 \end{aligned}$$

and the weight of w is $[w] = 1$ because $[w] + [\partial_x] = [u] = 2$.

Next, we compute the linearization of equation (6) and of the relations that specify the new variable,

```
linearize({df(b(1),t)= -df(b(1),x,3) + b(1)*df(b(1),x),
           df(b(2),x)= b(1),
           df(b(2),t)= -df(b(2),x,3) + df(b(2),x)**2}, 0, 2);
```

The linearization correspondence between the fields is

$$u \mapsto U \quad (b(1) \mapsto b(3)), \quad w \mapsto W \quad (b(2) \mapsto b(4)).$$

The linearized system is

$$\begin{aligned} \text{df}(b(3), t) &= b(3) * \text{df}(b(1), x) + b(1) * \text{df}(b(3), x) - \text{df}(b(3), x, 3), \\ \text{df}(b(4), x) &= b(3), \\ \text{df}(b(4), t) &= -\text{df}(b(4), x, 3) + \text{df}(b(2), x) * \text{df}(b(4), x) \end{aligned}$$

In this nonlocal setting, we obtain the nonlocal recursion of weight $\text{sw} = 2[\mathcal{R}] = 4$ as follows,

```
ssym(1, 6, 4, {}, {4, 2, 4, 2}, {
    df(b(3), t) = b(3) * df(b(1), x) + b(1) * df(b(3), x) - df(b(3), x, 3),
    df(b(1), t) => b(1) * df(b(1), x) - df(b(1), x, 3),
    df(b(2), x) => b(1),
    df(b(2), t) => -df(b(2), x, 3) + 1/2 * df(b(2), x) ** 2,
    df(b(4), x) => b(3),
    df(b(4), t) => -df(b(4), x, 3) + df(b(2), x) * df(b(4), x)
}, {}, {}, {lin});
```

We recall here that only the linearized system should be written as equations, and all other relations, including the nonlocalities, should be written as substitutions.

This yields the solution

$$\text{df}(b(3), s) = -3 * \text{df}(b(3), x, 2) + 2 * b(1) * b(3) + \text{df}(b(1), x) * b(4),$$

which is the well-known nonlocal recursion operator for KdV,

$$\varphi \mapsto \mathcal{R} = (-3D_x^2 + 2u + u_x \cdot D_x^{-1})(\varphi). \quad (8)$$

This recursion generates the hierarchy of *local* symmetries starting from the translation $\varphi_0 = u_x$. The powers $\mathcal{R}^2, \mathcal{R}^3, \dots$ of the recursion operator are also nonlocal.

Example 8. Consider the Burgers equation (2) and introduce the bosonic non-locality w of weight $[w] = [b] - [\partial_x] = 0$ by trivializing the conserved current $\partial_t(b) \doteq \partial_x(b_x - b^2)$. We therefore, set

$$w_x = b, \quad w_t = w_{xx} - w_x^2. \quad (9)$$

The linearization of the extended system is obtained through

```
linearize({df(b(1), t) = df(b(1), x, 2) - 2 * b(1) * df(b(1), x),
    df(b(2), x) = b(1),
    df(b(2), t) = df(b(2), x, 2) - df(b(2), x) ** 2}, 0, 2);
```

The correspondence between the bosonic fields is

$$b \mapsto B \quad (b(1) \mapsto b(3)), \quad w \mapsto W \quad (b(2) \mapsto b(4)).$$

The entire linearized system is (2) and (9) together with the relations

$$\begin{aligned} \text{df}(b(3), t) &= -2 * b(3) * \text{df}(b(1), x) - 2 * b(1) * \text{df}(b(3), x) + \text{df}(b(3), x, 2); \\ \text{df}(b(4), x) &= b(3); \\ \text{df}(b(4), t) &= \text{df}(b(4), x, 2) - 2 * \text{df}(b(4), x) * \text{df}(b(2), x). \end{aligned}$$

The difference between weights of the first-order and the second-order symmetries is 1. Hence, the recursion operator could have weight 1, see Remark 1. The nonlocal recursion of weight 1, $\mathbf{sw} = 2[\mathcal{R}] = 2$, is obtained by

```
max_deg:=1;
ssym(1, 4, 2, {}, {2, 0, 2, 0},{
  df(b(3),t) = -2*b(3)*df(b(1),x)-2*b(1)*df(b(3),x)+df(b(3),x,2),
  df(b(4),x) => b(3),
  df(b(4),t) => df(b(4),x,2)-2*df(b(4),x)*df(b(2),x),
  df(b(1),t) => -2*b(1)*df(b(1),x)+df(b(1),x,2),
  df(b(2),x) => b(1),
  df(b(2),t) => df(b(2),x,2)-df(b(2),x)**2}, {}, {}, {lin});
```

We finally get the recursion

$$\mathbf{df}(b(3),s) = -\mathbf{df}(b(3),x) + b(1)*b(3) + \mathbf{df}(b(1),x)*b(4),$$

which is nonlocal,

$$\varphi \mapsto \mathcal{R} = (-D_x + b + b_x D_x^{-1})(\varphi).$$

Example 9. Consider the super-field representation [7] of the Burgers equation, see (2),

$$f_t = \mathcal{D}b, \quad b_t = \mathcal{D}f + b^2;$$

its weights are $|f| = |b| = \frac{1}{2}$, $|\partial_x| = 1$, and $|\partial_t| = \frac{1}{2}$.

We introduce the nonlocal bosonic field $w(x, t; \theta)$ of weight $[w] = [f] - [\mathcal{D}] = 1 - 1 = 0$ such that

$$\mathcal{D}w = -f, \quad w_t = -b.$$

We get the linearized system by

```
linearize({df(f(1),t)= d(1, b(1)),
  df(b(1),t)= df(f(1),x) + b(1)**2,
  d(1,b(2)) = -f(1),
  df(b(2),t)= -b(1)}, 1, 2);
```

For the linearization correspondence between the fields is $f \rightarrow F$, $b \rightarrow B$, $w \rightarrow W$, and we have

$$F_t = \mathcal{D}B, \quad B_t = 2bB + \mathcal{D}F, \quad \mathcal{D}W = -F, \quad W_t = -B,$$

that is,

```
df(f(2),t)= d(1, b(3)),
df(b(3),t)= 2*b(1)*b(3)+d(1,f(2)),
d(1,b(4)) = -f(2),
df(b(4),t)= -b(3).
```

In this setting, we obtain the nonlocal recursion of weight $\frac{1}{2}$, $\mathbf{sw} = 2[\mathcal{R}] = 1$: the input is

```
max_deg:=1;
ssym(1, 1, 1, {1, 1}, {1, 0, 1, 0},
      {df(f(2),t) = d(1, b(3)),
       df(b(3),t) = 2*b(3)*b(1) + d(1, f(2)),
       d(1, b(4)) => - f(2),
       df(b(4),t) => - b(3),
       df(f(1),t) => d(1, b(1)),
       df(b(1),t) => d(1, f(1)) + b(1)**2,
       d(1, b(2)) => -f(1),
       df(b(2),t) => -b(1)}, {}, {}, {lin});
```

The recursion is

$$\begin{aligned} df(f(2),s) &= d(1,b(3)) + d(1,b(1))*b(4) - f(2)*b(1), \\ df(b(3),s) &= b(4)*b(1)**2 + b(3)*b(1) + d(1,f(2)) + d(1,f(1))*b(4), \end{aligned}$$

in other words,

$$\begin{pmatrix} F \\ B \end{pmatrix} \mapsto \mathcal{R} \begin{pmatrix} F \\ B \end{pmatrix} = \begin{pmatrix} DB - Db\mathcal{D}^{-1}F - Fb \\ -b^2\mathcal{D}^{-1}F + Bb + \mathcal{D}F - \mathcal{D}f\mathcal{D}^{-1}F \end{pmatrix}.$$

Example 10. Consider the fifth order evolution superequation found by Tian and Liu (Case F in [14], see also [6, 15]):

$$f_t = f_{5x} + 10(f_{xx}\mathcal{D}f)_x + 5(f_x\mathcal{D}f_x)_x + 15f_x(\mathcal{D}f)^2 + 15f(\mathcal{D}f_x)(\mathcal{D}f). \quad (10)$$

In what follows, we are considering this equations in components. Substitution $\xi + \theta u$ for f in (10), for example, using SSTOOLS, we obtain

$$u_t = u_{5x} + 10uu_{xxx} + 20u_xu_{xx} + 30u^2u_x - 5\xi_{xxx}\xi_x + 15u\xi_{xx}\xi + 15u_x\xi_x\xi, \quad (11a)$$

$$\xi_t = \xi_{5x} + 10u\xi_{xxx} + 15u_x\xi_{xx} + 5u_{xx}\xi_x + 15u^2\xi_x + 15uu_x\xi, \quad (11b)$$

where u is a bosonic field and ξ is a fermionic field.

Observe that the bosonic limit ($\xi := 0$) is the fifth order symmetry of Korteweg–de Vries equation (6). However, a direct computation shows that the equation (11) has local symmetries of the orders $1+6k$ and $5+6k$, where $k \in \mathbb{N}$, and does not have any local symmetries of order $3+6k$, where $k \in \mathbb{N}$. Therefore, the recursion operator for (11) should be at least of order 6. Let us also assume that the bosonic limit of the recursion operator for (11) is the third power \mathcal{R}^3 of the recursion operator (8) for the Korteweg–de Vries equation.

It is easy to check that for the construction of the 3rd power of the recursion operator (8) we should “trivialise” the following conserved densities of the Korteweg–de Vries equation: u , $u^2 + u_{xx}$ and $u_{4x} + 6uu_{xx} + 5u_x^2 + 2u^3$.

Let S and Q satisfy the linearized equation for (11). The correspondence between fields is the following $u \mapsto S$, $\xi \mapsto Q$. The linearized system of nonlocalities for the generalisation of those conservation laws for the supersymmetric equation (11) is the following:

1) the layer of nonlocalities corresponding to the generalisation of the conserved density u of the Korteweg–de Vries equation

$$\begin{aligned} W_{1;x} &= S, \\ W_{1;t} &= -15Qu\xi_x + S(30u^2 + 10u_{xx} - 15\xi\xi_x) - 5Q_{xx}\xi_x + Q_x(5\xi_{xx} + 15\xi u) + S_{4x} \\ &\quad + 10S_{xx}u + 10S_xu_x, \end{aligned}$$

2) the layer of nonlocalities corresponding to the generalisation of the conserved density $u^2 + u_{xx}$ of the Korteweg–de Vries equation

$$\begin{aligned} W_{2;x} &= 2Su + 2Q_x\xi, \\ Q_{2;x} &= Qu + S\xi, \\ W_{2;t} &= Q(-30u^2\xi_x + 30\xi uu_x) + S(60u^3 + 40uu_{xx} + 2u_{4x} - 30\xi_{xx}\xi_x - 120\xi u\xi_x \\ &\quad - 20\xi\xi_{xxx}) + 2Q_{5x}\xi - 2Q_{4x}\xi_x + Q_{xxx}(2\xi_{xx} + 20\xi u) + Q_{xx}(-30u\xi_x - 2\xi_{xxx} \\ &\quad + 30\xi u_x) + Q_x(30u\xi_{xx} + 2\xi_{4x} + 60\xi u^2 + 10\xi u_{xx}) + 2S_{4x}u - 2S_{xxx}u_x \\ &\quad + S_{xx}(20u^2 + 2u_{xx} - 10\xi\xi_x) + S_x(-2u_{xxx} - 30\xi\xi_{xx}), \\ Q_{2;t} &= Q(15u^3 + 10uu_{xx} + u_{4x} + 5u_x^2 - 5\xi_{xx}\xi_x) + S(20u\xi_{xx} + \xi_{4x} - 5\xi_x u_x + 45\xi u^2 \\ &\quad + 10\xi u_{xx}) + Q_{4x}u - Q_{xxx}u_x + Q_{xx}(10u^2 + u_{xx} + 5\xi\xi_x) + Q_x(-5uu_x - u_{xxx} \\ &\quad - 5\xi\xi_{xx}) + S_{4x}\xi - S_{xxx}\xi_x + S_{xx}(\xi_{xx} + 10\xi u) + S_x(-5u\xi_x - \xi_{xxx} + 10\xi u_x), \end{aligned}$$

3) the layer of nonlocalities corresponding to the generalisation of the conserved densities $u_{4x} + 6uu_{xx} + 5u_x^2 + 2u^3$ of the Korteweg–de Vries equation

$$\begin{aligned} W_{3;x} &= -9Q_2\xi u + 3Qu\xi_x + S(6u^2 + 3\xi\xi_x) - Q_{xxx}\xi - 3Q_x\xi u + 2S_{xx}u, \\ Q_{3;x} &= Q_2(6u^2 - 6\xi\xi_x) - 6W_2\xi u + 14Su\xi_x + 2Q_{xxx}u + 7Q_xu^2 - 2S_{xxx}\xi, \\ W_{3;t} &= Q_2(-90u^2\xi_{xx} - 9u\xi_{4x} + 45u\xi_x u_x + 9\xi_{xxx}u_x - 9\xi_{xx}u_{xx} + 9\xi_x u_{xxx} - 135\xi u^3 \\ &\quad - 90\xi uu_{xx} - 9\xi u_{4x} - 45\xi u_x^2 + 45\xi\xi_{xx}\xi_x) + Q(45u^3\xi_x + 9u^2\xi_{xxx} + 3u\xi_{5x} \\ &\quad - 42u\xi_{xx}u_x + 21u\xi_x u_{xx} - 3\xi_{4x}u_x + 3\xi_{xxx}u_{xx} - 3\xi_{xx}u_{xxx} + 3\xi_x u_{4x} + 42\xi_x u_x^2 \\ &\quad - 15\xi uu_{xxx} - 45\xi u_{xx}u_x) + S(180u^4 + 300u^2u_{xx} + 32uu_{4x} + 138u\xi_{xx}\xi_x \\ &\quad - 32u_{xxx}u_x + 16u_{xx}^2 + 16\xi_{4x}\xi_x - 16\xi_{xxx}\xi_{xx} + 135\xi u^2\xi_x + 78\xi u\xi_{xxx} + 13\xi\xi_{5x} \\ &\quad + 93\xi\xi_{xx}u_x + 111\xi\xi_x u_{xx}) + Q_{6x}\xi_x + Q_{5x}(-\xi_{xx} - 13\xi u) + Q_{4x}(6u\xi_x + \xi_{xxx} \\ &\quad - 32\xi u_x) + Q_{xxx}(-26u\xi_{xx} - \xi_{4x} + 26\xi_x u_x - 24\xi u^2 - 48\xi u_{xx}) + Q_{xx}(69u^2\xi_x \\ &\quad + 16u\xi_{xxx} - 15\xi_{xx}u_x + 17\xi_x u_{xx} - 48\xi uu_x - 22\xi u_{xxx}) + Q_x(-69u^2\xi_{xx} \\ &\quad + 4u\xi_{4x} + 45u\xi_x u_x - \xi_{xxx}u_x - 2\xi_{xx}u_{xx} + 5\xi_x u_{xxx} - 45\xi u^3 - 66\xi uu_{xx} \\ &\quad - 8\xi u_{4x} - 87\xi u_x^2) + 2S_{6x}u - 2S_{5x}u_x + S_{4x}(26u^2 + 2u_{xx} + 8\xi\xi_x) \\ &\quad + S_{xxx}(28uu_x - 2u_{xxx} + 22\xi\xi_{xx}) + S_{xx}(120u^3 + 112uu_{xx} + 2u_{4x} - 28u_x^2 \\ &\quad + 22\xi_{xx}\xi_x + 81\xi u\xi_x + 48\xi\xi_{xxx}) + S_x(180u^2u_x + 48uu_{xxx} - 16u_{xx}u_x \\ &\quad + 16\xi_{xxx}\xi_x + 3\xi u\xi_{xx} + 32\xi\xi_{4x} + 174\xi\xi_x u_x) + \xi Q_{7x}, \end{aligned}$$

$$\begin{aligned}
 Q_{3;t} = & Q_2(90u^4 + 120u^2u_{xx} + 12uu_{4x} - 180u\xi_{xx}\xi_x - 12u_{xxx}u_x + 6u_{xx}^2 - 12\xi_{4x}\xi_x \\
 & + 12\xi_{xxx}\xi_{xx} - 270\xi u^2\xi_x - 60\xi u\xi_{xxx} - 6\xi\xi_{5x} - 90\xi\xi_{xx}u_x - 30\xi\xi_x u_{xx}) \\
 & + Q(45u^3u_x + 24u^2u_{xxx} + 84uu_{xx}u_x - 12u\xi_{xxx}\xi_x - 28u_x^3 - 84\xi_{xx}\xi_x u_x \\
 & + 60\xi u^2\xi_{xx} - 24\xi u\xi_{4x} - 30\xi u\xi_x u_x - 96\xi\xi_{xxx}u_x - 84\xi\xi_{xx}u_{xx} - 36\xi\xi_x u_{xxx}) \\
 & + W_2(-60u^2\xi_{xx} - 6u\xi_{4x} + 30u\xi_x u_x + 6\xi_{xxx}u_x - 6\xi_{xx}u_{xx} + 6\xi_x u_{xxx} - 90\xi u^3 \\
 & - 60\xi uu_{xx} - 6\xi u_{4x} - 30\xi u_x^2 + 30\xi\xi_{xx}\xi_x) + S(660u^3\xi_x + 288u^2\xi_{xxx} + 34u\xi_{5x} \\
 & - 28u\xi_{xx}u_x + 590u\xi_x u_{xx} - 34\xi_{4x}u_x + 34\xi_{xxx}u_{xx} - 34\xi_{xx}u_{xxx} + 34\xi_x u_{4x} \\
 & + 28\xi_x u_x^2 + 135\xi u^2u_x - 102\xi uu_{xxx} - 20\xi u_{5x} - 366\xi u_{xx}u_x - 72\xi\xi_{xxx}\xi_x) \\
 & - 2Q_{6x}u_x + Q_{5x}(27u^2 + 2u_{xx} - 10\xi\xi_x) + Q_{4x}(36uu_x - 2u_{xxx} - 20\xi\xi_x) \\
 & + Q_{xxx}(106u^3 + 124uu_{xx} + 2u_{4x} - 36u_x^2 - 20\xi_{xx}\xi_x + 72\xi u\xi_x) \\
 & + Q_{xx}(121u^2u_x + 36uu_{xxx} - 12u_{xx}u_x + 10\xi_{xxx}\xi_x - 30\xi u\xi_{xx} + 20\xi\xi_{4x} \\
 & + 144\xi\xi_x u_x) + Q_x(165u^4 + 295u^2u_{xx} + 24uu_{4x} + 28u_x^2 - 30u\xi_{xx}\xi_x \\
 & - 24u_{xxx}u_x + 12u_{xx}^2 + 10\xi_{4x}\xi_x - 10\xi_{xxx}\xi_{xx} - 42\xi u\xi_{xxx} + 10\xi\xi_{5x} \\
 & - 114\xi\xi_{xx}u_x + 30\xi\xi_x u_{xx}) + 2S_{6x}\xi_x + S_{5x}(-2\xi_{xx} - 20\xi u) + S_{4x}(44u\xi_x \\
 & + 2\xi_{xxx} - 80\xi u_x) + S_{xxx}(16u\xi_{xx} - 2\xi_{4x} + 36\xi_x u_x - 36\xi u^2 - 140\xi u_{xx}) \\
 & + S_{xx}(325u^2\xi_x + 104u\xi_{xxx} - 82\xi_{xx}u_x + 94\xi_x u_{xx} - 306\xi uu_x - 140\xi u_{xxx}) \\
 & + S_x(91u^2\xi_{xx} + 56u\xi_{4x} + 386u\xi_x u_x - 22\xi_{xxx}u_x - 12\xi_{xx}u_{xx} + 46\xi_x u_{xxx} \\
 & + 45\xi u^3 - 246\xi uu_{xx} - 80\xi u_{4x} - 414\xi u_x^2 - 174\xi\xi_{xx}\xi_x) + 2uQ_{7x} - 2\xi S_{7x}.
 \end{aligned}$$

Here W_1, W_2, W_3 are bosonic fields and Q_2, Q_3 are fermionic fields. Let us note that the generalisation of the conservation law of (6) with the density $u_{4x} + 6uu_{xx} + 5u_x^2 + 2u^3$ is no longer a local conservation law for (11).

The weights of fields are as follows

$$\begin{aligned}
 |x| = -1, \quad |t| = -5, \quad |u| = |S| = 2, \quad |\xi| = |Q| = \frac{3}{2}, \\
 |W_1| = 1, \quad |W_2| = 3, \quad |W_3| = 5, \quad |Q_2| = \frac{5}{2}, \quad |Q_3| = \frac{11}{2}.
 \end{aligned}$$

Using the technique described above we obtain the following recursion operator (cf. [14])

$$\mathcal{R} \left(\begin{pmatrix} S \\ Q \end{pmatrix} \right) = \begin{pmatrix} S' \\ Q' \end{pmatrix},$$

where

$$\begin{aligned}
 S' = & 3Q_3\xi_x + Q_2(-42u\xi_{xx} - 6\xi_{4x} - 42\xi_x u_x - 18\xi u^2) + Q(-51u^2\xi_x - 12u\xi_{xxx} \\
 & + 2\xi_{5x} - 26\xi_{xx}u_x - 22\xi_x u_{xx} + 18\xi uu_x - 2\xi u_{xxx}) + 4W_3u_x + W_2(24uu_x \\
 & + 4u_{xxx} + 6\xi\xi_{xx}) + W_1(120u^2u_x + 40uu_{xxx} + 4u_{5x} + 80u_{xx}u_x - 20\xi_{xxx}\xi_x \\
 & - 60\xi u\xi_{xx} - 60\xi\xi_x u_x) + S(128u^3 + 192uu_{xx} + 24u_{4x} + 144u_x^2 - 16\xi_{xx}\xi_x \\
 & - 180\xi u\xi_x - 30\xi\xi_{xxx}) - 12Q_{4x}\xi_x + Q_{xxx}(-6\xi_{xx} + 36\xi u) + Q_{xx}(-12u\xi_x \\
 & + 8\xi_{xxx} + 58\xi u_x) + Q_x(-12u\xi_{xx} + 8\xi_{4x} - 22\xi_x u_x + 72\xi u^2 + 30\xi u_{xx}) + 2S_{6x} \\
 & + 24S_{4x}u + 60S_{xxx}u_x + S_{xx}(96u^2 + 80u_{xx} - 30\xi\xi_x) \\
 & + S_x(280uu_x + 60u_{xxx} - 54\xi\xi_{xx}),
 \end{aligned}$$

$$\begin{aligned}
 Q' = & 3Q_3u + Q_2(42uu_x + 6u_{xxx}) + Q(35u^3 + 48uu_{xx} + 2u_{4x} + 40u_x^2 - 6\xi_{xx}\xi_x \\
 & + 4\xi_{xxx}) + 4W_3\xi_x + W_2(-12u\xi_x - 2\xi_{xxx} - 6\xi u_x) + W_1(60u^2\xi_x + 40u\xi_{xxx} \\
 & + 4\xi_{5x} + 60\xi_{xx}u_x + 20\xi_x u_{xx} + 60\xi uu_x) + S(114u\xi_{xx} + 22\xi_{4x} + 104\xi_x u_x \\
 & + 72\xi u^2 + 30\xi u_{xx}) + 2Q_{6x} + 24Q_{4x}u + 48Q_{xxx}u_x + Q_{xx}(54u^2 + 38u_{xx} + 4\xi\xi_x) \\
 & + Q_x(126uu_x + 14u_{xxx}) + 12S_{xxx}\xi_x + S_{xx}(42\xi_{xx} + 42\xi u) \\
 & + S_x(100u\xi_x + 46\xi_{xxx} + 54\xi u_x).
 \end{aligned}$$

In [6] this system of nonlocalities was used to construct a zero-curvature representation of (11) to prove its integrability.

Acknowledgements

This work was supported in part by an NSERC grant to T. Wolf who is thanked by A. V. K. for warm hospitality. This research was done in part while A. O. K. was visiting at New York University Abu Dhabi; the hospitality and warm atmosphere of this institution are gratefully acknowledged.

References

- [1] Berezin F A, *Introduction to superanalysis*. Mathematical Physics and Applied Mathematics, D. Reidel Publishing Co., Dordrecht–Boston, MA, 1987.
- [2] Bocharov A V, Chetverikov V N, Duzhin S V, Khor’kova N G, Krasil’shchik I S, Samokhin A V, Torkhov Y N, Verbovetsky A M, Vinogradov A M, *Symmetries and conservation laws for differential equations of mathematical physics, Translations of Mathematical Monographs*, volume 182. American Mathematical Society, Providence, RI, 1999, xiv+333 pp.
- [3] Delduc F, Gallot L, Ivanov E, New super KdV system with the $N = 4$ SCA as the Hamiltonian structure. *Phys. Lett. B*, **396**:1-4(1997), 122–132.
- [4] Deligne P, Etingof P, Freed D S, Jeffrey L C, Kazhdan D, Morgan J W, Morrison D R, Witten E, *Quantum fields and strings: a course for mathematicians*, volume 1-2. AMS, Providence, RI; Institute for Advanced Study (IAS), Princeton, NJ, 1999, Vol. 1: xxii+723 pp.; Vol. 2: pp. i–xxiv and 727–1501 pp.
- [5] Kiselev A V, The twelve lectures in the (non)commutative geometry of differential equations. Preprint IHÉS/M/12/13 (Bures-sur-Yvette, France), 2012, <http://preprints.ihes.fr/2012/M/M-12-13.pdf>.
- [6] Kiselev A V, Krutov A O, On the (non)removability of spectral parameters in \mathbb{Z}_2 -graded zero-curvature representations and its applications, 2018, [arXiv:math.DG/1301.7143v6](https://arxiv.org/abs/math/1801.07143) [math.DG].

- [7] Kiselev A V, Wolf T, Supersymmetric representations and integrable fermionic extensions of the Burgers and Boussinesq equations. *SIGMA Symmetry Integrability Geom. Methods Appl.*, **2** (2006), Paper 030, 19 pp., [arXiv:math-ph/0511071](#).
- [8] Kiselev A V, Wolf T, Classification of integrable super-systems using the SsTools environment. *Comput. Phys. Comm.*, **177**:3(2007), 315–328, [arXiv:nlin/0609065 \[nlin.SI\]](#).
- [9] Krasil’shchik I S, Kersten P H M, *Symmetries and recursion operators for classical and supersymmetric differential equations, Mathematics and its Applications*, volume 507. Kluwer Academic Publishers, Dordrecht, 2000, xvi+384 pp.
- [10] Krasil’shchik J, Verbovetsky A, Vitolo R, *Symbolic Computation of Integrability Structures for Partial Differential Equations, Texts & Monographs in Symbolic Computation*, Springer, Cham, 2017, xv+263 pp.
- [11] Krivonos S, Pashnev A, Popowicz Z, Lax pairs for $N = 2, 3$ supersymmetric KdV equations and their extensions. *Modern Phys. Lett. A*, **13**:18(1998), 1435–1443.
- [12] Kupershmidt B A, Mathieu P, Quantum Kortevog-de Vries like equations and perturbed conformal field theories. *Phys. Lett. B*, **227**:2(1989), 245–250.
- [13] Olver P J, *Applications of Lie groups to differential equations, Graduate Texts in Mathematics*, volume 107. Springer-Verlag, New York, second edition, 1993, xxviii+513 pp.
- [14] Tian K, Liu Q P, Supersymmetric fifth order evolution equations. *AIP Conf. Proc.*, **1212**(2010), pp. 81–88. Nonlinear and modern mathematical physics. (July 15–21, 2009; Beijing, China).
- [15] Tian K, Wang J P, Symbolic representation and classification of $N = 1$ supersymmetric evolutionary equations. *Stud. Appl. Math.*, **138**:4(2017), pp. 467–498. [arXiv:1607.03947 \[nlin.SI\]](#).
- [16] Wolf T, Schrüfer E, SsTOOLS: Computations with supersymmetric algebraic and differential expressions. <http://lie.math.brocku.ca/crack/susy/sstools.red>.