AN ALGORITHM OF CONSTRUCTING THE CONSERVATION LAWS OF NONLINEAR EVOLUTION EQUATIONS

RUO-XIA YAO*;†,‡ and ZHI-BIN LI*;§

*Department of Computer Science, East China Normal University, Shanghai, 200062, China
†Department of Computer Science, Weinan Teachers College, Shaanxi, 714000, China
‡yb02241008@student.ecnu.edu.cn
§lizb@cs.ecnu.edu.cn

Received 31 August 2003

An algorithm of constructing the polynomial type conservation laws of nonlinear evolution equations (either parameterized or non-parameterized) is described and a Maple package CONSLAW to automate the computation is presented. For parameterized equation(s), CONSLAW can determine the parameters constraints automatically such that a sequence of conservation laws exist.

Keywords: Symbolic computation; nonlinear evolution equation; scaling symmetry; conservation law; integrability.

PACS numbers: 02.30.Jr, 02.70.Wz, 02.90.+p, 03.40.Kf, 03.65.Fd

1. Introduction

Knowledge of conservation laws (first integrals, invariants, constants of motion) is of great importance in the study of dynamical systems. There is a deep belief that every conservation laws (CLaws) reflects some profound physical principle acting in the system. On the other hand, sufficient number of independent CLaws leads to a complete integrability of dynamical system. A very extensive study of CLaws and symmetries is found in Olver, which includes both Lagrangian and Hamiltonian formulations. Although some lower order CLaws can be obtained directly by hand, the higher order ones needs extremely long and tedious work. This sort of calculation is an ideal candidate for such computer algebraic manipulation systems as Macsyma, Mathematica and Maple, which allow one to find the higher order CLaws for nonlinear evolution equations. Researchers have done much work on how to automatically construct the CLaws for differential equations. In this paper, by using the polynomial ansatz, a systematic procedure for obtaining the polynomial type CLaws for given nonlinear evolution equations is described, which

§Corresponding author.
has been implemented in *Maple*. The package *CONSLAW* automates the computation of CLaws of complicated nonlinear evolution equations. More importantly, new integrable system can be found while constructing the CLaws of nonlinear parameterized evolution equations.

2. Algorithm

It is well known that the classical KdV equation

\[ u_t + uu_x + u_{3x} = 0, \] (1)

has infinitely many CLaws and is invariant under the scaling symmetry

\[ (t, x, u) \rightarrow (\lambda^{-3} t, \lambda^{-1} x, \lambda^2 u), \] (2)

where \( u_{3x} = \partial^3 / \partial x^3 \) and \( \lambda \) is an arbitrary parameter. Before proceeding the computation of CLaws, the following definitions are introduced. Equation (1) is 2-homogeneous of weight 3. Here, let \( \omega \) stand for the weight of a variable, then in view of (2), we have \( \omega(u) = 2 \), \( \omega(\partial / \partial t) = 3 \), and without loss of generality, we can set \( \omega(\partial / \partial x) = 1 \). Another definition of the rank of a monomial is useful, which is defined as the total weight of the monomial in terms of derivatives with respect to \( x \), denoted by \( \text{Rank}(M) \). It is easy to see that all monomials in Eq. (1) have the same rank 5. This property is called *uniformity in rank*.

Consider nonlinear evolution equation(s)

\[ u_t = H(u, u_x, u_{2x}, \ldots), \] (3)

where \( H \) is polynomial of \( u \) and \( u_{ix} \) (\( i = 1, 2, \ldots \)), \( u_{ix} = \partial^i u / \partial x^i \). For now, we assume Eq. (3) is uniform in rank, i.e. it is invariant under some scaling transformation.

A conservation law for Eq. (3) is a divergence expression

\[ \rho_t + J_x = 0, \] (4)

which vanishes for all solutions \( u = f(x, t) \) of Eq. (3). As a matter of fact, for most of the nonlinear evolution equation or equations, with few exceptions, the conserved density-flux pairs are polynomials in \( u \) and \( u_{ix} \), and do not depend explicitly on \( x \) and \( t \). Such CLaws are called polynomial type CLaws.

An algorithm to constructing the polynomial type CLaws for Eq. (3) is described as follows.

- **Determining the weights (scaling properties) of dependent variables and parameters.** Requiring uniformity in rank leads to a linear system of equations for the unknown weights, which on solving determines the weights of the dependent variables and parameters. For example, the ranks of the three terms in Eq. (1) are \( \omega(\partial / \partial t) + \omega(u), 1 + 2 \omega(u) \) and \( 3 + \omega(u) \) respectively. The requirement of uniformity in rank leads to \( \omega(\partial / \partial t) + \omega(u) = 1 + 2 \omega(u) = 3 + \omega(u) \), which on solving yields \( \omega(u) = 2, \omega(\partial / \partial t) = 3 \). However, there indeed exists such a case that
maybe the equation is not uniformity in rank. For instance, the terms \( \alpha uu_x \) and \( \beta u^2 u_x \) appearing in KdV-MKdV equation
\[
    u_t + \alpha uu_x + \beta u^2 u_x + u_{3x} = 0, \tag{5}
\]
do not allow uniform rank. To process such case congruously, we treat \( \alpha \) and \( \beta \) as extra variables with (unknown) weights. That is, we have
\[
    \omega(\partial/\partial t) + \omega(u) = 1 + \omega(\alpha) + 2\omega(u) = 1 + \omega(\beta) + 3\omega(u) = 3 + \omega(u),
\]
which on solving yields
\[
    \{\omega(\alpha) = \omega(\alpha), \omega(\partial/\partial t) = 3, \omega(u) = -\omega(\alpha) + 2, \omega(\beta) = -2 + 2\omega(\alpha)\}.
\]
Since all the weights should be nonnegative, we choose \( \omega(\alpha) = 1 \), then \( \omega(u) = \omega(\alpha) = 1, \omega(\partial/\partial t) = 3 \). That means only \( \alpha \) is a weighted parameter.

- **Determining the form of the conserved density \( \rho \), i.e. finding the components in \( u \) and their \( x \)-derivatives of the conserved density with prescribed rank \( R \).** In detail, the procedure proceeds as follows:
  1. Collect the dependent variables and the weighted parameters in set \( V \).
  2. Form the basis set \( B \). It consists of such elements as \([M, rank(M)]\), where \( M \) is one of the monomials of rank \( R \) or less by taking all appropriate combinations of different powers of the elements in set \( V \), and \( rank(M) \) is the rank of the monomials \( M \).
  3. Form the set \( Q \) of all monomials in \( u \) and the \( x \)-derivatives of \( u \) with rank \( R \). To do this, for each element in \( B \), we compute \( l_i = R - rank(M_i) \), which form a list \( L \). Then compute the \( x \)-derivative of \( M_i \) up to \( l_i \) such that the new generating monomials exactly have rank \( R \). Meanwhile, we gather the new generating monomials in set \( Q \).
  4. Remove the redundant monomials in set \( Q \). If some monomials in set \( Q \) belong to the same equivalence class, i.e. their conserved densities are equivalent if they only differ by a total \( x \)-derivative, then identify them. Denote the simplified set as \( P \).
  5. Linear combination of the elements in set \( P \) with constant coefficients \( c_i \)'s yields the form of the polynomial conserved density of rank \( R \).

Carrying on with Eq. (1), we compute the form of the density of rank \( R = 6 \). From \( V = \{u\} \) we get \( B = \{[u, 2], [u^2, 4], [u^3, 6]\} \). Easily, we get the list \( L = [4, 2, 0] \), then compute the various \( x \)-derivatives of \( u, u^2 \) and \( u^3 \) up to 4, 2 and 0 respectively such that the new generating monomials exactly have rank 6. Therefore, we have
\[
    \frac{d^4}{dx^4} u = u_{4x}, \quad \frac{d^2}{dx^2} (u^2) = 2uu_{2x} + 2u_x^2, \quad \frac{d^0}{dx^0} (u^3) = u_3. \tag{6}
\]
Then the set \( Q = \{u_{4x}, uu_{2x}, u_x^2, u^3\} \). Since \( u_{4x} = \frac{d^4}{dx^4}(u_{3x}) \), and \( uu_{2x} = \frac{d^2}{dx^2}(uu_x) - u_x^2 \), therefore, \( uu_{2x} \) is replaced by \( u_x^2 \), and \( u_{4x} \) is cancelled. Hence, we obtain the set \( P = \{u_x^2, u^3\} \), and the form of the conserved density of rank 6 is
\[
    \rho = c_1 u_x^2 + c_2 u^3. \tag{7}
\]
- **Determining the unknown coefficients in the conserved density \( \rho \).**
(1) Substitute $\rho$ into Eq. (4) and use Eq. (3) to eliminate all $t$-derivatives. Note that, the resulting expression $E$ must be a total $x$-derivative, which means that the Euler–Lagrange equation must vanish identically, i.e. $L_u(E) = 0$, where $L_u$ is Euler operator

$$L_u = \frac{\partial}{\partial u} - D_x \left( \frac{\partial}{\partial u_x} \right) + D_x^2 \left( \frac{\partial}{\partial u_{2x}} \right) + \cdots + (-1)^n D_x^n \left( \frac{\partial}{\partial u_{nx}} \right).$$ (8)

(2) Group the remained terms that including the ones in $V$ and the $x$-derivatives of the dependent variables in $V$ up to $N$, where $N$ is the highest order of the derivative of the expression $E$. Then let them be zero. This yields a linear system for the $c_i$’s, which on solving can determine the various relations among the parameters and the undetermined constants $c_i$’s.

For example, computing the $t$-derivative of (7) yields

$$\rho_t = 3c_2 u^2 u_t + 2c_1 u_x u_{xt}.$$ (9)

Replacing all the $t$-derivatives in Equ. (9) by $u_t = -uu_x - u_{3x}$ yields

$$E = -3c_2 u^3 u_x - 3c_2 u^2 u_{3x} - 2c_1 u_x^3 - 2c_1 u_x u_{2x}^2 - 2c_1 u_x u_{4x}.$$ Applying the Euler operator $L_u$ on $E$ leads to $6(c_1 + 3c_2) u_x u_{2x} = 0$, which on solving gives $c_1 = -3c_2$, $c_2 = c_2$. Letting $c_2 = 1$, we obtain the conserved density as

$$\rho = -3u_x^2 + u^3.$$ (10)

### Determining the corresponding conserved flux

Once the conserved density $\rho$ determined, then, the corresponding conserved flux $J$ can be obtained from Eq. (4) by using the idea of integrating by parts. For example, the conserved flux corresponding to Eq. (10) is

$$J = -6u_x^2 u - 6u_x u_{3x} + 3u_{2x}^2 + \frac{3}{4} u^4 + 3u^2 u_{2x}.$$ (11)

Equations (10) and (11) is the third conserved density-flux pair of KdV equation.

We have implemented the algorithm and developed a package `CONSLAW` in Maple, which automates the computation and delivers, if exist, the possible conserved density-flux pairs for a prescribed rank of Eq. (3). The effectiveness of the Maple package `CONSLAW` is illustrated by a lot of nonlinear evolution equations. Next, we give three examples.

**Example 1.** Working with Eq. (1), as mentioned before, it has infinitely many polynomial CLaws. Theoretically, our package `CONSLAW` can compute them one by one, provided a conserved density of that rank exists. Here, we give two CLaws for which the rank of density is 4 and 12 respectively.
First, we read in the package CONSLAW, and then start the program. After inputting Eq. (1) and the rank 4 of the conserved density, \textsc{conslaw} outputs the following CLaw immediately:

\[(u^2)_t + \left(\frac{2}{3}u^3 + 2uu_{2x} - u_x^2\right)_x = 0\, ,
\]

which represents the conservation of energy.

In the case of Rank(\(\rho\)) = 12, likewise, \textsc{conslaw} delivers the conserved density-flux pair within 3.545s. Because the number of the terms of the conserved flux is 20, which is omitted here. The conserved density is

\[\rho = \frac{216}{7}u_{1x}^2 - 60u^3u_x^2 + 8u_{3x}^2 - \frac{648}{7}uu_{3x} + \frac{720}{7}u_x^3 + u^6 + 108u_{2x}u_x^2\, ,
\]

**Example 2.** Let us consider KdV-MKdV Eq. (5) again, which also has infinitely many CLaws. For this parameterized equation, \textsc{conslaw} can automatically determines whether the parameters with weight or not, and further gives the parameters constraint such that a sequence of CLaws exist. For Eq. (5), \textsc{conslaw} gives \(\omega(u) = \omega(\alpha) = 1\) and \(\omega(\alpha^2) = 3\). Here we also only list two CLaws, for which the rank of density is 2 and 6, respectively.

In the case of Rank(\(\rho\)) = 2, \textsc{conslaw} outputs the following conserved density-flux

\[\rho = \alpha u + u^2, \quad J = \alpha uu_{2x} + \frac{2\alpha}{3}u^3 + \frac{\beta}{2}u^4 + 2uu_{2x} - u_x^2 + \frac{\alpha^2}{2}u^2 + \frac{\alpha\beta}{3}u^3,\]

for which without any constraints on the parameters.

In the case of Rank(\(\rho\)) = 6, only the conserved density is listed as

\[\rho = -3\alpha^2u_x^2 + \beta u^2u_x^2 - \frac{\beta^2}{30}u^6 + \alpha\beta uu - \frac{3}{5}u_{2x}^2 + \alpha^4u^2 + \alpha^3u^3 + \alpha uu_x^2
\]

\[\quad - \frac{\beta\alpha}{10}u^5 + \frac{\beta\alpha^2}{2}u^4 - \frac{\alpha^2}{12}u^4,\]

for which without any constraints on the parameters.

**Example 3.** The generalized seventh order KdV equation

\[u_t + \tilde{a}u^3u_x + \tilde{b}u_x^3 + \tilde{c}uu_xu_{2x} + \tilde{d}u^2u_{3x} + \tilde{e}u_{2x}u_{3x} + \tilde{f}u_xu_{4x} + 5u_{5x} + u_{7x} = 0\, ,
\]

is reduced to

\[u_t + au^3u_x + bu_x^3 + cuu_xu_{2x} + du^2u_{3x} + eu_{2x}u_{3x} + fu_xu_{4x} + uu_{5x} + u_{7x} = 0\, ,
\]

(12)

by introducing a simple transformation \(u = u/\tilde{g}\), where \(a = \tilde{a}/\tilde{g}^3, b = \tilde{b}/\tilde{g}^2, c = \tilde{c}/\tilde{g}, d = \tilde{d}/\tilde{g}^2, e = \tilde{e}/\tilde{g}, f = \tilde{f}/\tilde{g}\). For Eq. (12), there are three cases that are of particular interest: (i) \((a, b, c, d, e, f) = (5/98, 5/14, 10/7, 5/14, 5, 3)\) (Lax equation); (ii) \((a, b, c, d, e, f) = (4/147, 1/7, 6/7, 7/2, 3, 2)\) (SK-Ito equation); (iii) \((a, b, c, d, e, f) = (4/147, 5/14, 9/7, 2/7, 6, 7/2)\) (unnamed 7th-order equation).
For Eq. (12), **CONSLAW** gives $\omega(u) = 2, \omega(\partial/\partial t) = 7$ first. Then, rich higher order polynomial type CLaws are obtained. Here, two conserved densities are listed as:

In the case of $\text{Rank}(\rho) = 8$, **CONSLAW** reports only one branch of parameters constraints as

$$
\begin{aligned}
\left\{ 
 a &= \frac{(4 - f)(10f^2 - 45f + 49c + 20)}{882} , \\
 b &= \frac{-9f + 2f^2 + 4 + 14c}{42} , \\
 d &= \frac{7c - 4 + 9f - 2f^2}{42} , \\
 e &= 2f - 1 , \\
 c, f : & \text{ free}
\right. 
\end{aligned}
$$

(13)

and gives one set of conserved density-flux pairs for Eq. (12). Due to the number of the terms included in the conserved flux is more then one hundred, we only list the conserved density as

$$
\rho = \frac{10f^2 - 45f + 49c + 20}{1764} u^4 + \frac{1 - 2f}{7} uu_x^2 + u_x^2 .
$$

(14)

In the case of $\text{Rank}(\rho) = 12$, **CONSLAW** reports three branches of parameters constraints and the corresponding conserved density-flux pairs one by one. There are 74 terms in each of the three conserved fluxes, which are too long to write down here. The conserved densities are listed as follows.

(A) The first branch of parameters constraints is

$$
\left\{ 
 a &= \frac{5}{98} , \\
b &= \frac{5}{14} , \\
c &= \frac{10}{7} , \\
d &= \frac{5}{14} , \\
e &= 5 , \\
f &= 3
\right. 
$$

(15)

and the associated conserved density is

$$
\rho = \frac{3}{2744} u^6 + u_x^2 + \frac{5}{28} u_x^4 + \frac{10}{7} u_x^2 u_x^2 - \frac{9}{7} uu_x^2 - \frac{15}{98} u^3 u_x^2 + \frac{9}{14} u^2 u_x^2 ,
$$

(16)

which, obviously, is only admitted by the seventh order Lax equation.

(B) The second branch of parameters constraints is

$$
\left\{ 
 a &= \frac{4}{147} , \\
b &= \frac{1}{7} , \\
c &= \frac{6}{7} , \\
d &= \frac{2}{7} , \\
e &= 3 , \\
f &= 2
\right. 
$$

(17)

and the conserved density is

$$
\rho = \frac{4}{21609} u^6 + u_x^2 + \frac{17}{147} u_x^4 + \frac{16}{21} u_x^2 u_x^2 - \frac{50}{1029} u^3 u_x^2 + \frac{16}{49} u^2 u_x^2 ,
$$

(18)

which is only admitted by the well known SK-Ito equation.

(C) The third branch of parameters constraints is

$$
\left\{ 
 a &= \frac{4}{147} , \\
b &= \frac{5}{14} , \\
c &= \frac{9}{7} , \\
d &= \frac{2}{7} , \\
e &= 6 , \\
f &= \frac{7}{2}
\right. 
$$

(19)

and the conserved density is

$$
\rho = \frac{16}{21609} u^6 + u_x^2 + \frac{31}{294} u_x^4 + \frac{37}{21} u_x^2 u_x^2 - \frac{10}{7} uu_x^2 - \frac{20}{147} u^3 u_x^2 + \frac{34}{49} u^2 u_x^2 ,
$$

(20)
which is only admitted by the following equation
\begin{align*}
u_t &= \frac{4}{147} u^3 u_x + \frac{5}{14} u^3_x + \frac{9}{7} u u_x u_{2x} + \frac{2}{7} u^2 u_{3x} + 6 u u_{2x} u_{3x} \\
&\quad + \frac{7}{2} u_x u_{4x} + uu_{5x} + u_{7x}.
\end{align*}

(21)

3. Summary

Up to now, the package CONSLAW has automatically delivered the polynomial type CLaws of as many ranks as possible for more than forty nonlinear parameterized or non-parameterized evolution equations. Here, we mention that although the package CONSLAW can not construct CLaws depending explicitly on t or x directly, for some PDEs, such CLaws can be obtained indirectly with the help of CONSLAW.\textsuperscript{6}

Acknowledgment

Project supported by the Research Found for the Doctoral program of the Higher Education of China (Grant No. 20020269003).

References