Constants of motion for the KdV and mKdV equations *

Márcia T. Fontenelle and Jason A.C. Gallas

Laboratório de Óptica Quântica, Departamento de Física da UFSC, 88049 Florianópolis, Brazil

Received 15 May 1989; in revised form 6 December 1989

We present recurrence relations that generate the terms appearing in the known expressions for the densities and the fluxes of the Korteweg–de Vries equation, and in the densities of the modified Korteweg–de Vries equation. Our relations are based on the observation that for a given rank \( r \) the set \( \{ P_r \} \) of all partitions of the integer \( 2r \) contains all monomials of \( \{ X, _1 \} \) and \( \{ X, _r \} \). The relations are very easy to program in systems able to perform computer algebra. In addition, we report three new constants of motion for the modified Korteweg–de Vries equation.

It is well known that certain nonlinear partial differential equations arising in the study of a number of different physical systems ranging from nonlinear optics to hadron physics obey what are called conservation laws. A prominent example is the Korteweg–de Vries (KdV) equation which contains an infinite sequence of conservation laws [1]. The discovery of such an infinite sequence of conservation laws for the KdV equation motivated, in subsequent years, a lot of activity on evolution equations possessing infinitely many symmetries [2]. Conservation laws are equations of the generic type

\[
T_t + X_x = 0, \tag{1}
\]

where \( T \) (the conserved density) and \( -X \) (the corresponding flux of \( T \)) are polynomials in a “field” variable \( u(x, t) \) and its derivatives, i.e. are sums of monomials

\[
c(i_0, i_1, ..., i_n)u_0^{i_0}u_1^{i_1}...u_n^{i_n}, \tag{2}
\]

where \( u_0 = u(x, t) \), \( u_n = \partial^n u(x, t)/\partial x^n \), \( c(i_0, i_1, ..., i_n) \) is a constant and \( i_0, i_1, ... \) are nonnegative integers. A brute-force approach to obtain densities and fluxes consists of summing several monomials (2) and, using eq. (1), setting up a system of equations to be solved for the several (sometimes hundreds) of \( c(i_0, i_1, ..., i_n) \).

For systems with uniform rank [5] there are ways of knowing how many monomials are needed in the summation for every rank \( r \). In practice one starts with far more monomials than needed and from constraints imposed by eq. (1) determines those monomials really present in \( T \) and \( X \). This approach is, however, difficult to follow since one quickly ends up with systems of equations involving several hundreds of coefficients \( c(i_0, i_1, ..., i_n) \) to be determined. In this paper we present recurrence relations that provide directly the monomials that really appear in \( T \) and \( X \) and only these. This has the effect of reducing the number of unknowns and equations to a minimum, thereby rendering possible the investigation of conservation laws of the KdV and mKdV equations and, we hope, in the future, of other nonlinear evolution equations.

It was pointed out to us by the referee of this paper that recurrence relations have long been used in connection with conservation laws for soliton equations [3]. We see, however, no direct connection between our results and the already available body of results on recurrence relations for soliton equations.

In a recent paper, Torriani [4] showed how to use combinatorial methods to obtain constants of motion for the Korteweg–de Vries (KdV) and related equations. His very interesting procedure

* Work partially supported by the Brazilian agencies CAPES, CNPq and FINEP.

0010-4655/90/$03.50 © 1990 – Elsevier Science Publishers B.V. (North-Holland)
consists of associating partitions of integers and
their Ferrers graphs to the first density T and the
first flux X and then, through simple rules, to
generate all subsequent T and X obeying eq. (1).
For example, let us take \( \cdots \) as the Ferrers graph
associated with \( u_0 \). His first conjecture [4] gives
then \( \cdots \) as the only possible graph. This graph is
associated to the partition \( 2^2 \) and to the monomial
\( u_0^2 \), which is the only one present in \( T_2 \). Applying
Conjecture 1 to the graph \( \cdots \) gives
\[ \cdots \] and \( \cdots \),
which are associated to the partitions \( 2^3 \) and \( 3^2 \),
corresponding to \( u_0^3 \) and \( u_1^2 \), respectively. A fu-
ter application of the conjecture generates three
different graphs, corresponding to the partitions
\( 2^4, 23^2 \) and \( 2^4 \), producing \( u_4^4, u_0u_1^2 \) and \( u_2^2 \), respecti-
vely. In this clever way Torriani was able to
generate all known densities and fluxes of the
KdV equation as well as the known densities of
the modified KdV equation (mKdV).
The main purpose of this paper is to present an
alternative method to generate the monomials pre-
sent in the densities and fluxes of the KdV and
mKdV equations. Our approach has much to do with
the combinatorial approach proposed by
Torriani [4], but the overlap of both procedures is
difficult to assess. The biggest advantage of the
method being proposed here is that it provides
recurrence relations from which the monomials can
be obtained. These recurrence relations are easy to pro-
gram in systems able to perform algebraic computa-
tions like, for example, REDUCE.
Let us start with the KdV equation. We use the
word monomial as in Torriani [4], but denote the
set of monomials in \( T_r \) by \( \{ T_r \} \), in \( X_r \) by \( \{ X_r \} \),
etc. Our method is based on the following em-
pirical observations made of the set of monomials
presently available in the literature [1,5]:
(a) for a given rank \( r \), the set \( \{ P_{2r} \} \) of all parti-
tions of the integer \( 2r \) contains all monomials
of \( \{ X_{r-1} \} \) and \( \{ T_r \} \);
(b) those monomials belonging to \( \{ P_{2r} \} \) but not
to \( \{ X_{r-1} \} \) define a set \( \{ Q_{2r} \} \); those in \( \{ P_{2r} \} \)
but not in \( \{ T_r \} \) define a set \( \{ R_{2r} \} \);
(c) the set \( \{ P_r \} \) may be easily generated recur-
svively from integrations over \( u \) and in the
derivatives of \( u \);
(d) all three sets are obtained from identical re-
currence relations, but with different initial
conditions, i.e. the generation of \( \{ Q_r \} \) and
\( \{ R_r \} \) is identical to that of \( \{ P_r \} \).
Since the method generates only monomials
(and not the numerical coefficients in the densities
and fluxes), all integrations over \( u_i \) may be per-
formed as though they were simple multiplications
by \( u_j \). Explicitly, for \( r \geq 2 \) we obtain
\[ \{ T_r \} = \{ P_{2r} \} - \{ R_{2r} \}, \] \[ \{ X_r \} = \{ P_{2r+2} \} - \{ Q_{2r+2} \}, \] where
\[ P_i = u_{i-2} + \sum_{k=0}^{(i-4)/2} u_k P_{i-k-2}, \quad i \geq 4, \] \[ Q_i = u_{i-2} + \sum_{k=0}^{(i-7)/2} u_k Q_{i-k-2}, \quad i \geq 7, \] \[ R_i = u_{i-2} + \sum_{k=0}^{(i-5)/2} u_k R_{i-k-2}, \quad i \geq 5, \] where the symbol \([x]\) means the largest integer not
greater than \( x \). The following initial values are
required by the recurrence relations:
\( P_2 = Q_2 = R_2 = u_0, \) \( P_3 = Q_3 = R_3 = u_1, \) \( Q_4 = R_4 = u_2, \) \( Q_5 = u_3, \)
and \( Q_6 = u_4 \). Although numerical coefficients of
the monomials in eqs. (3-4) are totally meaning-
less, we found it convenient to avoid summation
of repeated terms. This can be achieved by drop-
ing from the summations all products of \( u_k \)
with monomials containing \( u_j \) with \( j < k \). This obvi-
ously constrains all numerical coefficients to be
unity. In appendix A we give a REDUCE pro-
gram that was used here to generate the monomi-
als in the densities and fluxes of the invariants
of the KdV equation. Procedure PROD in this
program is used to avoid summation of repeated
terms. The monomials will also be correctly gener-
ated if calls to PROD(J, F(K)) are simply replaced
by U(J) \* F(K). However, in this case the numeri-
cal coefficients of the monomials will not be unity
anymore.
The above recurrence relations for \( T_r \) and \( X_r \)
define our method to generate all and only those
monomials contained in the densities and fluxes,
respectively, for the KdV equation. In an analo-
M. T. Fontenelle, J. A. C. Gallas / Constants of motion for the KdV and mKdV equations

In this context, the densities for the modified KdV terms of $c(r, 0, \ldots, 0)$ and arbitrarily set $c(r, 0, \ldots, 0) = 1/r$. Accordingly, the total number of unknowns is given by the sum of monomials in $T_r$ and in $X_r$, minus 1.

As part of the present work we also recalculated for the KdV equation all constants of motion up to $r \leq 10$, and for the mKdV up to $r \leq 8$. The constants of motion were obtained from the algebraic procedures given in the appendices A and B. After determining the monomials for every rank, we calculated all numerical coefficients by solving the linear system of equations resulting from eq. (1). Such a calculation guarantees that every function generated from the monomials obtained by our method was indeed a constant of motion and that our method generates the correct number of monomials. We found a misprint in one of the constants of motion already available in the literature: in eq. (10b) of ref. [5] the coefficient of $u_0 u_2^2$ should be $+144/7$ instead of $-144/7$.

For the mKdV we obtained the following three new constants of motion:

$T_6 = \frac{1}{12} v_0^{12} - \frac{55}{2} v_0 v_1^2 + 66 v_2^2 - 319 v_0 v_1^4 - v_0^4 - 394 v_1^2 v_2^2$  
$+ \frac{260}{7} v_0 v_3^2 + \frac{12276}{7} v_0 v_1 v_2^2 + v_0^2 v_1^2 v_2^2 - \frac{396}{7} v_0 v_2^2 v_3^2$  
$- \frac{5214}{7} v_1^4 v_2^2 + \frac{2173}{7} v_2^4 - 108 v_2^6$,  

$T_7 = \frac{1}{14} v_0^{14} - 39 v_0^2 v_1^2 + 117 v_0^2 v_2^2 - 1014 v_0^4 v_1^2$  
$- \frac{14004}{7} v_0 v_3^2 + 9260 v_0 v_1 v_2^2 + 63180 v_0 v_1 v_2^2 v_3^2 + 1404 v_0^4 v_3^2$  
$- \frac{28080}{7} v_0^3 v_2^2 - 22740 v_0^2 v_2^2 - 46120 v_0^2 v_2^2 v_3^2$  
$+ 35100 v_0 v_2^4 + \frac{8424}{7} v_0^2 v_2^4 v_3^2 + \frac{17912}{7} v_0^2 v_2^4 v_3^2$  
$+ \frac{16948}{11} v_0 v_2^2 v_3^2 + \frac{9878}{7} v_0 v_2^2 v_3^2 + \frac{109912}{7} v_0 v_2^2 v_3^2$  
$- 39312 v_1^4 v_2^2 - \frac{958104}{77} v_0 v_2^2 v_3^2 + \frac{1944}{77} v_0^2 v_2^2 v_3^2$.  

Table 1: Number of monomials (unknowns) in $T_r$ and $X_r$ with rank $r \leq 10$. BF represents the number of all monomials that exist for a given rank while M gives the effective number of monomials present in $T_r$ or $X_r$ as generated by our method. The last two lines give the total number of unknowns and of equations in the system obtained by substituting $T_M$ and $X_M$ into eq. (1). For $r > 5$ the system is overdetermined.

<table>
<thead>
<tr>
<th>Rank $r$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coef. in $T_M$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>21</td>
<td>34</td>
<td>55</td>
<td>88</td>
<td>137</td>
</tr>
<tr>
<td>Coef. in $X_M$</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>14</td>
<td>22</td>
<td>32</td>
</tr>
<tr>
<td>Coef. in $T_M$</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>21</td>
<td>34</td>
<td>55</td>
<td>88</td>
<td>137</td>
<td>210</td>
</tr>
<tr>
<td>Coef. in $X_M$</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>20</td>
<td>31</td>
<td>47</td>
<td>71</td>
<td>105</td>
<td>136</td>
</tr>
<tr>
<td>No. of unknowns</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>16</td>
<td>27</td>
<td>44</td>
<td>60</td>
<td>92</td>
<td>136</td>
</tr>
<tr>
<td>No. of equations</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>21</td>
<td>34</td>
<td>55</td>
<td>88</td>
<td>137</td>
</tr>
</tbody>
</table>
Constants of motion for the KdV and mKdV equations

\[ T_8 = \frac{1}{16} v_0^{16} - \frac{105}{2} v_0^{12} v_1^2 + 189 v_0^{10} v_2^2 - \frac{5145}{2} v_0^8 v_3^2 \\
- 405 v_0^6 v_4^2 + 3600 v_0^4 v_5^2 + 31860 v_0^2 v_6^2 \\
+ 540 v_0^2 v_7^2 - 16200 v_0 v_8^2 v_1^2 - 30474 v_0 v_9^2 v_1^2 \\
- 34020 v_0 v_1 v_2 v_3^2 + 31590 v_0 v_1 v_3 v_4^2 - \frac{4860}{11} v_0 v_2 v_3 v_4^2 \\
+ 286848 v_0 v_1 v_2 v_3 v_4 - 136080 v_0 v_1 v_2 v_3 v_4 v_5 + 330804 v_0 v_1 v_2 v_3 v_5^2 \\
+ 127640 v_0 v_1 v_2 v_3 v_5^2 - \frac{57108}{11} v_0 v_1 v_2 v_3 v_5^2 - \frac{1507986}{11} v_0 v_2 v_3^2 v_4^2 \\
+ \frac{29160}{143} v_0 v_2 v_3^2 v_4^2 - \frac{367136}{11} v_0 v_2 v_3^2 v_4^2 + \frac{712980}{11} v_0 v_2 v_3^2 v_4^2 \\
- \frac{544320}{143} v_0 v_2 v_3^2 v_4^2 + \frac{483888}{143} v_0 v_2 v_3^2 v_4^2 - \frac{83153}{7} v_1^2 v_3^2 \\
- \frac{712746}{143} v_1^2 v_3^2 + \frac{3083508}{11} v_1^2 v_3^2 - \frac{513160}{11} v_1^2 v_3^2 \\
+ \frac{5334336}{143} v_1^2 v_3^2 v_4^2 + \frac{3938332}{143} v_1^2 v_3^2 v_4^2 - \frac{227028}{143} v_1^2 v_3^2 \\
- \frac{58333}{143} v_1^2 v_3^2. \quad (9) \]

The corresponding fluxes contain many more terms than the densities. For example, \( X_6 \) contains 42 terms, \( X_7 \) contains 69 and \( X_8 \) contains 110, while \( T_6, T_7 \) and \( T_8 \) above contain 13, 20 and 32 terms, respectively.

In summary, we report simple recurrence relations able to generate all monomials in the known expressions for the densities and fluxes of the KdV equation, as well as those in the densities of the mKdV equation. The great advantage of our recurrence relations is that they are easy to implement in systems able to perform computer algebra. A REDUCE implementation of them is given in appendices A and B. Therefore, besides allowing direct construction of constants of motion, one is able to investigate peculiar properties of complicated nonlinear evolution equations as well as to demonstrate the strong impact of combinatorial analysis in this field. The new constants of motion reported in this paper support the conjectures of Torriani [4]. In a subsequent paper we intend to present a recurrence relation for the fluxes of the mKdV equation and to investigate the structure of the next few constants of motion for both KdV and mKdV equations. All this work is part of a preliminary effort the ultimate goal of which is the study of the integrability of much more complicated nonlinear evolution equations of the generic type

\[ u_t - a u_{xxx} = F(u, u_x, u_{xx}, \ldots), \quad (10) \]

like, for example, the BBM equation [6] and the equations discussed by Caldas and Tasso [7]. Before concluding we would like to observe that for the much simpler case discussed in this paper (of equations having uniform rank and \( a = 0 \)) there is already in the literature [8] an algorithm in RLISP to generate the densities (not the fluxes). It would be interesting now to write a REDUCE code able of generating not only the monomials but also the corresponding numerical coefficients.

Acknowledgements

MTF is a predoctoral fellow of the Brazilian Research Council (CNPq). JACG is a research fellow of the CNPq. He thanks Professor H.H. Torriani for helpful correspondence. The work of our group is supported by the Brazilian agencies CNPq, FINEP and CAPES.

Appendix A

LINELENGTH 60$ OFF NAT$ ON LIST$ OPERATOR US$ ARRAY P(50), Q(50), R(50), TT(20), XX(20)$

COMMENT PROGRAM TO GENERATE DENSITIES TT AND FLUXES XX FOR THE KDV EQUATION UP TO RANK R (= RNK) USING Eqs. (2–5) OF THE TEXTS

RNK: = 15$ KLIN := 2 * (RNK + 1)$
Q(2) := U(0)$ Q(3) := U(1)$ Q(4) := U(2)$ Q(5) := U(3)$ Q(6) := U(4)$
P(2) := U(0)$ P(3) := U(1)$
R(2) := U(0)$ R(3) := U(1)$ R(4) := U(2)$
PROCEDURE PROD(J,POLIN)$
BEGIN SCALAR TERMO,PROV$ ARRAY G(25), V(25, 25), VV(25)$
    TERMO := POLIN$
    IF J EQ 0 THEN PROV := U(0)*TERMO$
    IF J NEQ 0 THEN BEGIN$
        G(0) := COEFF(TERMO, U(0), VV)$
        FOR I := 0: G(0) DO V(0, I) := VV(I)$
        IF J GEQ 2 THEN BEGIN$
            FOR I := 1: J — 1 DO BEGIN$
                G(I) := COEFF(V(I — 1, 0), U(I), VV)$
                FOR K := 0: G(I) DO V(I, K) := VV(K)$
            END$
        END$
        PROV := U(J)*V(J — 1, 0)$
    END$ RETURN PROV$
END$

COMMENT XX(K) = P(2*(K + 1)) — Q(2*(K + 1))
TT(K) = P(2*K) — R(2*K)$

FACTOR U(0), U(1), U(2), U(3), U(4), U(5), U(6), U(7), U(8), U(9), U(10), U(11), U(12), U(13), U(14), U(15)$

FOR K := 4: KLIM DO BEGIN$
    IF K GEQ 4 THEN P(K) := U(K — 2) + FOR J := 0: ((K — 4)/2) SUM PROD (J, P(K — J — 2))$
    IF K GEQ 5 THEN R(K) := U(K — 2) + FOR J := 0: ((K — 5)/2) SUM PROD (J, R(K — J — 2))$
    IF K GEQ 7 THEN Q(K) := U(K — 2) + FOR J := 0: ((K — 7)/2) SUM PROD (J, Q(K — J — 2))$
    IF FIXP(K/2 — 1) THEN BEGIN$
        XX(K/2 — 1) := P(K) — Q(K)$
        WRITE "XX(", K/2 — 1, ") := \ xx(K/2 — 1)$
    END$
    IF FIXP(K/2) THEN BEGIN$
        TT(K/2) := P(K) — R(K)$
        WRITE "TT(", K/2, ") := \ tt(K/2) END$
    END$
END$}

APPENDIX B

LINELENGTH 60$ OFF NATS ON LISTS OPERATOR VS ARRAY MT(50), MR(50)$

COMMENT PROGRAM TO GENERATE THE DENSITIES MT FOR THE MODIFIED KDV EQUATION UP TO RANK R (= RNK) USING EQ. (6) OF THE TEXT$
RNK := 8$ MT(1) := V(0) • • 2$
MR(1) := 0$ MR(2) := 0$ MR(3) := 0$ MR(4) := 0$ MR(5) := V(0) • V(2) • • 3$
MR(6) := V(0) • V(2) • V(3) • • 2$ MR(7) := V(0) • V(2) • V(4) • • 2 + V(1) • V(3) • • 3$
FACTOR V(0), V(1), V(2), V(3), V(4), V(5), V(6), V(7), V(8), V(9), V(10),
V(11), V(12), V(13), V(14), V(15)$
FOR K := 2: RNK DO BEGIN$
  IF K GEQ 8 THEN
    MR(K) := V(K — 3) • • 2 • V(0) • V(2) + V(K — 4) • • 2 • (V(1) • V(3) + V(0) • V(4))$
  IF K GEQ 2 THEN
    MT(K) := V(K — 1) • • 2 + FOR J := 0: (K — 2) SUM (V(J) • • 2 • MT(K — J — 1)) + MR(K)$
  WRITE "MT(", K," : " := ", MT(K)$
END$
END$

References