Computer algebra algorithms and routines for the computation of conservation laws and fixing of gauge in differential expressions

Thomas Wolf
Queen Mary & Westfield College, University of London,
Mile End Road, London E1 4NS, UK
email: T.Wolf@maths.qmw.ac.uk

Andreas Brand
Institut für Informatik, Friedrich Schiller Universität Jena
email: maa@cnve.rz.uni-jena.de

Majid Mohammadzadeh
Faculty of Mathematics and Computer Engineering
University of Teacher Education
49 Mofateh Ave, PB 15614, Tehran
email: majid@saba.tmu.ac.ir

November 19, 2001

Abstract

In this paper we present three different approaches for the determination of conservation laws. For three corresponding `REDUCE` computer algebra programs `CONLAW1/2/3` the necessary subroutines are described. All three programs use subroutines which remove redundant functions and constants in the general solution of the conservation law conditions. The corresponding algorithm is explained. Such a program is not only applicable in the context of computing conservation laws but whenever redundacy in differential expressions is to be removed or gauge freedom to be fixed.

1 Introduction

The determination of conservation laws for single or systems of partial differential equations (PDEs) and of first integrals for ordinary differential equations (ODEs) is of interest for the exact solution of these DEs, for their understanding and classification and for supporting their numerical solution. In this paper we outline three computer algebra programs implemented in REDUCE for the computation of conservation laws. We also explain in detail a procedure that fixes redundancy of arbitrary functions and constants in differential expressions. This procedure is used to extract individual conservation laws from the general solution of conservation law determining equations.

In this paper we will not make the simplifying assumption that differential equations $\Delta = 0$ for which we want to find conservation laws, result from a variational principle. In that case any variational symmetry of the Lagrangian provides a conservation law as is known from Noether's Theorem. Instead, we will not make any restrictive assumptions which leaves us to solve $\mathrm{Div}\, P = 0$ either directly or to determine characteristic functions of conservation laws or to do both at once. A more application oriented comparison of these different approaches will be described elsewhere [11]. In this paper we concentrate on the computer algebra aspects.

The following section discusses the mathematical background of the three approaches. This is followed by a section on the computer algebra side of computing conservation laws. In this section the need for fixing redundancy in general solutions of conservation law conditions is motivated. A corresponding algorithm is explained in section 4 together with other procedures that

are more specific for the computation of conservation laws. In section 5 examples are given that demonstrate in which sense the programs `CONLAW1/2/3` are superior to other computer algebra programs that compute conservation laws. In two further sections the three methods are compared with each other and the syntax of the three programs is given.

2 The mathematical problem and the three approaches

In the following we adopt the notation of the book of Olver [7]. Independent variables will be denoted by $x = (x^1, x^2, \ldots, x^p)$. The differential equations for which we seek conservation laws are $\Delta(x, u^{(n)}) = 0$ (i.e. $\Delta_1 = 0, \ldots, \Delta_q = 0$), for q functions $u = (u^1, u^2, \ldots, u^q)$, $u^{(n)}$ denoting u - derivatives of order up to n. The conservation law that is to be fulfilled by solutions of $\Delta = 0$ is $\mathrm{Div}\, P = \sum_{i=1}^{p} D_i P^i = 0$ for a current $P = (P^1, \ldots, P^p)$ and total derivatives D_i. In this paper local conservation laws are considered where components P^i are expressions in $x^j, u^{(k)}$ and where $\mathrm{Div}\, P = 0$ has to be satisfied identically in $x^j, u^{(k)}$ modulo $\Delta = 0$. We will use $_J$ as a multiple index denoting partial derivatives, for example, u^α_J will stand for an arbitrary partial derivative, like $\partial^k u^\alpha / (\partial x^1 \partial x^2 \ldots)$.

A first approach to compute conservation laws is to solve

$$\mathrm{Div}\, P = 0 \quad \text{modulo} \;\; \Delta = 0 \tag{1}$$

directly. The components of the conserved current $P^1, \ldots, P^p$ that are to be calculated are allowed to depend on all independent variables x^i, the dependent variables u^α and their derivatives u^α_J up to some order.

We are not interested in trivial conservation laws $P = \mathtt{curl}\, V$ but rather in conservation laws that are obeyed by the solutions of $\Delta = 0$. Therefore we use $\Delta = 0$ to eliminate some of the so-called jet-variables u^α_J and substitute them in the determining conditions (1). This implies that the condition (1) has to be fulfilled identically in fewer variables. Thus the condition becomes less restrictive and so may have additional solutions apart from $P = \mathtt{curl}\, V$. These extra non-trivial conservation laws are the ones of interest. We therefore assume $\Delta = 0$ can be solved for leading derivatives u^α_J so that these and all their partial derivatives that occur in (1) can be substituted.

The conserved current P^i is defined only modulo $\Delta = 0$, because conservation laws are to be obeyed by solutions of $\Delta = 0$. We therefore assume, without loss of generality, that the P^i do not depend on the u-derivatives that we substitute. This fixes another kind of equivalence of conservation laws.

Other approaches calculate characteristic functions Q^ν. In ([7], p. 272) it is shown that, for a totally non-degenerate system $\Delta_\nu = 0$, each equivalence class of conservation laws $\mathrm{Div}\, P = 0$ (i.e. conserved currents differing only by a curl) is determined uniquely by characteristic functions Q^ν satisfying

$$\mathrm{Div}\, P = \sum_\nu Q^\nu \Delta_\nu \tag{2}$$

identically in *all* $x^i, u^\alpha, u^\alpha_J$. Equation (2) is not solved by simply eliminating Q^1 in terms of P and Δ and the other Q^ν as Q^1 would be singular for solutions of $\Delta = 0$. To avoid this problem and because the Q^ν are only unique modulo $\Delta = 0$, we can ignore dependencies of Q^ν on leading u-derivatives in $\Delta = 0$ and any of their derivatives. One way to calculate the Q^ν is to use the basic property of the Euler operators $E_\nu = \sum_J (-D)_J \partial/\partial(u^\nu_J)$, namely to give zero if and only if they act on an expression which is a divergence.

Applying the Euler operator on (2) gives

$$0 = \sum_{\mu,J} (-D)_J \left(\frac{\partial(Q^\mu \Delta_\mu)}{\partial(u^\nu_J)} \right) \quad \text{for all } \nu. \tag{3}$$

Using $\Delta_\nu = 0$ to simplefy the condition one obtains as necessary (but not sufficient) determining conditions for the Q^ν:

$$0 = \sum_{\mu,J} (-D)_J \left(Q^\mu \frac{\partial \Delta_\mu}{\partial(u^\nu_J)} \right) \quad \text{for all } \nu. \tag{4}$$

The three methods and programs discussed in this paper aim at solving (1), (2) and (4) identically in $x^i, u_\alpha, u^\alpha_J$. The program `CONLAW1` tries to solve (1) for P^i, `CONLAW2` tries to solve (4) for Q^ν and finally `CONLAW3` tries to solve (2) for P^i and Q^ν.

Any two of the three approaches (1)-(4) differ either in the number of conditional equations to be solved, the order of these equations, the number of functions to be determined, the number of independent jet-variables, or

the degree of an ansatz for P, Q that is necessary in order to obtain the same conservation laws.

To obtain solutions of (1)-(4) we assume bounds on the order of u-derivatives on which the P^i and Q^ν may depend. For (1) we assume a bound for P^i and for (2),(4) we assume a bound for Q^ν. Bounds for the remaining unknown functions follow. Differentiations done in all three conditions (1)-(4) introduce jet-variables (u-derivatives) on which none of the unknown functions depend.

As the conditions have to be satisfied identically in these variables as well, over determined PDE-systems result. Those are investigated with the computer algebra package `CRACK` ([9],[10]).

3 The computer algebra problem

The main computer algebra problem is to solve the over determined conditions (1)-(4). Steps for solving that problem include separation, integration, application of integrability conditions (differential Gröbner Basis), solution of ODEs and other steps, all of which are described in [9],[10]. The success rate of CRACK in solving over determined systems is high if the system is linear (like (1)-(4) which are linear in P^i, Q^ν) and not too big. Examples of what is currently possible are given in section 5. Any unsolved equations are returned. For example, when conservation laws of the Burgers equation are investigated in section (5), the heat equations remains as an unsolved condition and is returned. Complexity issues are discussed in [11].

From the general solution of (1),(2) or (4) a single conservation law is extracted by collecting all terms involving one of the arbitrary constants or arbitrary functions in the solution. Any redundant constants would give conservation laws which are not independent of each other.

Redundant constants and functions may result because in the process of solving the over determined system, there is no general rule for deciding whether integrations or substitutions should have a higher priority. There are examples requiring a higher priority for each of them. It therefore may happen that two equations are integrated which are not independent of each other and therefore the constants or functions of integration are not independent of each other. As a consequence, the final general solution could have redundant arbitrary constants and functions. For example, in the expres-

sion $c_1(x)t + c_2xt + c_3$, with independent variables x, t and arbitrary function $c_1(x)$ and arbitrary constants c_2, c_3, the constant c_2 is redundant as it can be absorbed by $c_1(x)$ through $c_1(x) \to c_1(x) - c_2x$. In this paper we will refer to redundancies also as gauge freedom.

Another application of redundancy recognition is the solution of PDE systems with some gauge freedom where the problem is to eliminate any gauge freedom from the general solution of this system. This can be accomplished by including in the solution terms representing the complete gauge freedom. For example, in the case of conditions (1)

- the general solution could be augmented by `curl` V and V be added to the list of free constants and functions,
- redundant constants and functions could be dropped, i.e. constants and functions that could be absorbed by V,
- finally V could be dropped from the list of free functions.

In this way trivial conservation laws could be filtered out as the free constants and functions corresponding to such conservation laws would have been absorbed by V.

In the case of computing conservation laws, one could easily drop trivial conservation laws after they have been computed by checking $\mathrm{Div}\, P = 0$ identically in all jet-variables. Such a simple test to eliminate gauge freedom might not be available for other problems. Therefore a general algorithm is described below.

4 Subroutines

In the following subsections we describe an algorithm for fixing redundancy in the general solution of a PDE-problem (and by that extracting conservation laws from the general solution of conditions (1)-(4)), subroutines that compute Q^ν from P^i and P^i from Q^ν, and subroutines that simplify P^i.

4.1 Identifying redundant constants and functions in solutions

This and the following section 4.2 are self contained and could be read independently of the rest of the paper. In them we consider solutions of arbitrary differential equations, having in mind the conditions (1)-(4) (and less the equations $\Delta = 0$, which we can not solve and which therefore are investigated for conservation laws). To avoid confusion we call these equations $\Omega(f_a, x^i) = 0$ which are to be solved for the functions $f_a(x^i)$. In the case of computing conservation laws, the equations $\Omega = 0$ are the equations (1)-(4), the f_a are the P^i and/or Q^μ, and the x^i now include all jet-variables. This notation is only used in the sub-sections 4.1 and 4.2.

In this sub-section we want to discuss the problem of fixing gauge freedom, if only a PDE-solver is available that can produce solutions with possible redundancies. Before giving the steps in a summary, the algorithm is explained using simple examples.

Given is a system of DEs $\Omega(f_a, x^i) = 0$ that is to be solved for the functions $f_a(x^i)$. We assume

$$f_b = F_b(x^i, g_c) \tag{5}$$

to be a general solution where F_b are differential expressions in x^i, g_c where g_c are arbitrary constants and functions of x^i (which, for example, resulted through integration).

The question is how to specify as many as possible of the g_c to fix any redundancy among them but not to loose generality of the solution. The idea behind the algorithm is the following.

If two sets of constants and functions g_c and $\bar{g}_c$ in the solution (5) differ only by a gauge then for both sets the solution (5) would still be the same identically in the x^i. Therefore the determining conditions for the redundancy will be the invariance condition for the general solution, i.e.

$$F_b(x^i, g_c) - F_b(x^i, \bar{g}_c) = 0 \tag{6}$$

identically in x^i where $\bar{g}_c$ are new functions, each $\bar{g}_a$ having the same variable dependence as the g_a with the same lower index.

Example: We assume the general solution of some problem for a single function $f(x^1, x^2, x^3)$ being computed as

$$f = x^1 g_1(x^2) + g_2(x^1, x^2) + x^3 g_3 \tag{7}$$

with arbitrary functions $g_1(x^2), g_2(x^1, x^2)$ and arbitrary constant g_3.[1]

For the solution (7), the invariance condition (6) is

$$x^1 g_1(x^2) + g_2(x^1, x^2) + x^3 g_3 - x^1 \bar{g}_1(x^2) - \bar{g}_2(x^1, x^2) - x^3 \bar{g}_3 = 0. \qquad (8)$$

Seen as a condition for functions $g_1(x^2), \bar{g}_1(x^2), g_2(x^1, x^2), \bar{g}_2(x^1, x^2)$ and constants $g_3, \bar{g}_3$ it is solved by separation of variables (here x^3):

$$\begin{aligned} 0 &= g_3 - \bar{g}_3, \\ 0 &= x^1 g_1(x^2) + g_2(x^1, x^2) - x^1 \bar{g}_1(x^2) - \bar{g}_2(x^1, x^2) \end{aligned}$$

which has as a general solution

$$\begin{aligned} \bar{g}_3 &= g_3 \\ \bar{g}_2(x^1, x^2) &= g_2(x^1, x^2) + x g_1(x^2) - x \bar{g}_1(x^2), \end{aligned} \qquad (9)$$

$\bar{g}_1(x^2), g_1(x^2), g_2(x^1, x^2), g_3$ are free. If it is possible to compute the general solution of the invariance condition (6) then all g_c for which g_c itself or $\bar{g}_c$ are computed (occur on the left hand side of this general solution) are *essential* free constants and functions of the general solution (5). In the solution (7) the $g_2(x^1, x^2), g_3$ are essential.

The remaining g_c which occur together with their counterpart $\bar{g}_c$ on right hand sides of the general solution of the invariance condition are *redundant*. They can be chosen freely (like $g_1(x^2)$ in solution (7) of the example). Any change in them can be compensated by a change in the essential functions, as (9) shows how $\bar{g}_2(x^1, x^2)$ has to be adapted if $g_1(x^2), \bar{g}_1(x^2)$ would be changed.

Although the solvability of (6) for $g_a, \bar{g}_b$ cannot be guaranteed, this should in practice not be a problem for the following reasons.

- Under normal circumstances when (5) is a solution to some PDE(s), functions g_c appearing on the right hand side have to be functions of less variables than f_b on the left hand side. The invariance condition (6) which is to be satisfied in *all* independent variables x^i identically is therefore a highly over determined problem for functions $g_c, \bar{g}_c$ of only a subset of variables of (6). Even if the problem to which (5) is

[1] The general solution to conservation law problems usually involves only constants, like in examples 1,3 in section 5. Example 2 in section 5 only involves functions because this is a rare case of a linearizable equation - the Burgers equation.

the solution is underdetermined, and so consequently some g_c have the same variable dependence as the f_b, still their number must be smaller than the number of f_b. In any case, the invariance condition (6) is a over determined set of conditions for the $g_c, \bar{g}_c$ and therefore well suited to be solved by the package `CRACK`.

- Although it is over determined, the invariance condition (6) has always at least one trivial solution $\bar{g}_c = g_c$, for all c. This is the general solution of the invariance condition if no redundancy exists.

- If the equations $\Omega = 0$ are linear in f_a, then their solution is linear in the arbitrary functions g_c, i.e. equation (6) is linear in the $g_c, \bar{g}_c$. This is the case for the computation of conservation laws.[2]

- If the equations $\Omega = 0$ are non-linear in f_a, then solving (6) should still be easier than the solving $\Omega = 0$, an equation which we assumed was possible to solve.

In the following example we show how new redundancy can result from solving the invariance condition. The example also illustrates that it may happen that one is able to compute gauge freedom but that it is impossible to remove it.

Example: As mentioned above, the more underdetermined a problem is, the more difficult it is in general to fix redundancy in its general solution. In this respect the following well known simple problem (one condition for 3 unknowns) is comparatively hard. Equation

$$0 = \mathrm{Div} f = \partial_1 f_1 + \partial_2 f_2 + \partial_3 f_3 \tag{10}$$

for functions f_1, f_2, f_3 of variables x^1, x^2, x^3 has the general solution

$$\begin{aligned} f_1 &= \partial_3 g_2 - \partial_2 g_3 \\ f_2 &= \partial_1 g_3 - \partial_3 g_1 \\ f_3 &= \partial_2 g_1 - \partial_1 g_2 \end{aligned} \tag{11}$$

where g_1, g_2, g_3 are arbitrary functions of x^1, x^2, x^3. (Equation (10) is *not* a conservation law problem. It is not to be solved modulo some equation, but describes a static source free field.)

[2]Conditions become non-linear if $\Delta = 0$ does contain parameters and we want to calculate their value such that conservation laws exist.

The general solution (11) must have gauge freedom as it has the same number of free functions g_i of all variables as there are functions f_i to be determined initially. To solve the invariance conditions

$$\begin{aligned} 0 &= \partial_3 g_2 - \partial_2 g_3 - \partial_3 \bar{g}_2 + \partial_2 \bar{g}_3 && (12)\\ 0 &= \partial_1 g_3 - \partial_3 g_1 - \partial_1 \bar{g}_3 + \partial_3 \bar{g}_1 && (13)\\ 0 &= \partial_2 g_1 - \partial_1 g_2 - \partial_2 \bar{g}_1 + \partial_1 \bar{g}_2 && (14) \end{aligned}$$

a computer algebra program could do the following steps:

- Solve (12) by introducing a new function $k_1(x^1, x^2, x^3)$ with

$$\bar{g}_2 - g_2 = \partial_2 k_1, \quad \bar{g}_3 - g_3 = \partial_3 k_1. \qquad (15)$$

- Solve (13) by substituting g_3 from (15) and integrate it with respect to x^3 with a new function $k_2(x^1, x^2)$ to get

$$\bar{g}_1 - g_1 = \partial_1 k_1 + k_2.$$

- equation (14) finally requires $\partial_2 k_2 = 0$, giving the general solution

$$\begin{aligned} \bar{g}_1 &= g_1 + \partial_1 k_1 + k_2 && (16)\\ \bar{g}_2 &= g_2 + \partial_2 k_1 && (17)\\ \bar{g}_3 &= g_3 + \partial_3 k_1 && (18) \end{aligned}$$

  with $k_1 = k_1(x^1, x^2, x^3),\ k_2 = k_2(x^1)$.

This example shows how solving the invariance condition may require solving differential equations, therefore the solution may involve extra redundant functions, here k_2. The free function k_1 is a function of all variables. Therefore, if k_1 occurs in one of equation (16)-(18) only algebraically then at least locally this equation could be solved for k_1, and k_1 could be substituted so that only two of the g_c are essential.

The situation in solution (16)-(18) is different. k_1 can not be solved for and the redundancy can not be fixed. This deficiency is not only a technical aspect. If a derivative of k_1 could be renamed to $\tilde{k_1}$ in order to have $\tilde{k_1}$ appearing algebraically in at least one equation and everywhere else only $\tilde{k_1}$ (not k_1) or derivatives of $\tilde{k_1}$ then this would be done in the corresponding

routine in CRACK. However, in (16)-(18) k_1 comes with different derivatives, therefore it is not a deficiency of the algorithm or computer program that gauge freedom can not be fixed explicitly in a general solution like (11).

The following summary is a list of steps as they are performed in the program CRACK to fix gauge freedom in a given solution (5)

$$f_b = F_b(x^i, g_c)$$

to some differential problem:

- Formulate a set of conditions (6)

  $$F_b(x^i, g_c) - F_b(x^i, \bar{g}_c) = 0$$

  where $\bar{g}_c$ are new functions, each $\bar{g}_c$ having the same variable dependence as g_c. Regard equation (6) as a system of equations for the set of unknown functions $\{g_c, \bar{g}_c\}$, to be satisfied identically in the x^i.

- Find the general solution of the system (6) as

  $$\tilde{g}_c = G_c(x^i, h_d) \tag{19}$$

  where $\tilde{g}_c$ is a subset of $\{g_a, \bar{g}_b\}$, and G_c are algebraic or differential expressions in functions h_d which are the remaining $\{g_a, \bar{g}_b\}$ and extra constants and functions of integration. (In the example above the G_c are the right hand sides of (16)-(18), the $\tilde{g}_c$ are $\bar{g}_1, \bar{g}_2, \bar{g}_3$ and the h_d are g_1, g_2, g_3, k_1, k_2.) The h_d are arbitrary. Any function g_a or $\bar{g}_a$ appears only once on a left hand side of (19) or only on right hand sides.

- If for any index c, both g_c and $\bar{g}_c$ appear only on right hand sides of (19) then g_c is redundant and can be set to zero in all F_b in (5) and all G_c in (19).

- The system (19) will now be solved for the $\bar{g}_c$: if for any index c, both g_c and $\bar{g}_c$ appear only on left hand sides of (19) in the equations $g_c = G_c$ and $\bar{g}_c = \bar{G}_c$ then these two equations are replaced by $\bar{g}_c = g_c - G_c + \bar{G}_c$ in (19).

- If for any index c, g_c appears on a left hand side of (19) and $\bar{g}_c$ appears only on right hand sides then the equation with left hand side g_c is solved for $\bar{g}_c$ and replaced by the new equation $\bar{g}_c = \bar{G}_c(g_c, \ldots)$. With this new equation $\bar{g}_c$ is substituted on any right hand side of (19).

- There remains only the case of $\bar{g}_c$ being on the left hand side of an equation and g_c being on right hand sides. The system (19) now has the form

$$\bar{g}_c = \bar{G}_c(x^i, g_a, \bar{h}_b) \tag{20}$$

  where $\bar{h}_b$ are arbitrary constants and functions of integration which arose during the solution of (6). $\bar{g}_c$ do not occur on right hand sides as they would be redundant and would have been set to zero otherwise.

- Free constants and functions $\bar{h}_b$ that occur only algebraically on the right hand side of at least one equation and that depend on all independent variables in that equation, will be chosen to make as many $\bar{G}_c$ as possible zero. In such a case also the redundant g_c is set to zero in (5) and (20). (In equation (16) k_2 could not be used as k_2 does not depend on all variables and k_1 could not be used as it does not occur algebraically in at least one equation, also not through the substitution of a derivative of k_1 by k_1 everywhere.) As we do not have to know $\bar{h}_b$ explicitly, it is enough to find equations in (20) which include such a $\bar{h}_b$. Assuming local solvability of $\bar{G}_c = 0$ for $\bar{h}_b$ we conclude redundancy of g_c.

- Finally, all remaining $\bar{h}_b$ which cannot be used to make a right hand side zero are set to zero themselves (in example (16)-(17) these are k_1, k_2). The final form of (20)

$$\bar{g}_c = \bar{G}_c(x^i, g_a) \tag{21}$$

  provides substitutions which turn the old solution (5) ($f_b = F_b(x^i, \bar{g}_c)$ after g_c have been renamed $\bar{g}_c$) into a new solution $f_b = \tilde{F}_b(x^i, g_c)$ for which as much redundancy as possible is fixed.

4.2 Identifying redundant constants and functions in solutions *and* remaining unsolved equations

The above steps for fixing redundancy are not only applicable once a general solution of a PDE(-system) $\Omega(f_a, x^i) = 0$ has already been found. Also, if a preliminary solution is known but some equations remain to be solved, then fixing at least some redundancy is possible. For example, the computation

of conservation laws for the Burgers equation in section 5 returns a solution together with the heat equation which remains unsolved.

What could the fixing of redundancies mean under these circumstances? Substitutions that find redundancies in g_c in a preliminary solution $f_b = F_b(x^i, g_c)$ and in remaining differential equations $D(x^i, g_c) = 0$ which have to be satisfied by g_c, should leave the solution completely unchanged but should be allowed to change the remaining equations $D = 0$ into a set of equivalent equations. This is in general a much harder problem than we had before as it involves equivalence of PDE-systems. A simpler question is to find redundancy fixing substitutions that leave the solution *and* equations unchanged. This is done by extending redundancy conditions (6) by $D(x^i, g_c) - D(x^i, \bar{g}_c) = 0$.

Although this simplified version of fixing redundancies seems to potentially miss redundancies, we have not encountered such a case yet, given that our remaining equations are at most polynomially non-linear (for conservation law calculations linear), unknown functions are not nested, or depending on expressions (like $f(x+y)$), and in each equation there is at least one function of all variables.

Example: Using the above steps to delete redundancies in the preliminary solution

$$f = g_1(x^1, x^3) - g_2(x^1, x^2) - g_3(x^2) \tag{22}$$

and the remaining condition

$$g_2(x^1, x^2) + g_3(x^2) + g_4(x^3) = 0 \tag{23}$$

would show that g_3 is redundant but g_4 not. The reason for not recognizing that after substituting g_2 from (23) in (22), g_4 can be absorbed by g_1, is that equation (23) does not involve a function of all variables x^1, x^2, x^3.

A situation like this can not occur in practice, as the program `CRACK` has routines to handle equations in which there is no function of all variables. An equation like (23) can therefore not remain unsolved. `CRACK` would find $g_4 =$const. (and $g_2 + g_3 =$const.) and then g_4 would be absorbed.

To look at the problem in terms of redundancy fixing substitutions that transform (23) into an equivalent condition, a computer program would have to be able to see that equation (23) is equivalent to

$$g_2(x^1, x^2) + g_3(x^2) + g_5 = 0, \quad \text{and} \quad g_4(x^3) = g_5 = \text{ const.}$$

and then g_5 would be found to be redundant.

The possibility of fixing gauge freedom even in the presence of yet unsolved equations opens the possibility of running a gauge-fixing step during the process of solving over determined PDE-systems (which is currently not done). By that the number of unknown functions could be reduced and the remaining equations simplified.

4.3 Computing characteristic functions from conserved currents

The remainder of the paper is concerned with the computation of conservation laws. In this sub-section we explain how to compute the characteristic functions Q from the conserved current P.

The first approach (1) to compute conservation laws is attractive compared with (2),(4) as it generates only one PDE to be solved which is of first order and involves fewer jet-variables than approach (2) because it is computed modulo $\Delta = 0$. Also, it has fewer functions to compute than approach (2). A negative aspect is that it provides only the conserved current P and not the characteristic functions Q.

If expressions $Q^{\nu J}$ in a relation (24) below are known then partial integrations (25) yield the characteristic functions Q^ν and the corresponding conserved current $P - R$:

$$\begin{aligned} \mathrm{Div}\, P = 0 \quad & \text{modulo } \Delta_\nu = 0 \quad \leftrightarrow \\ \exists\, Q^{\nu J} : \mathrm{Div}\, P \;&=\; \sum_{\nu,J} Q^{\nu J} \Delta_\nu^{(J)} \qquad \text{(identically in } \textit{all } x, u_J^\alpha) \quad (24) \\ &=\; \sum_{\nu,J} D_J(Q^{\nu J}\Delta_\nu) - D_J(Q^{\nu J})\Delta_\nu \qquad \text{(repeatedly)} (25) \\ &=\; \mathrm{Div}\, R + \sum_\nu Q^\nu \Delta_\nu \end{aligned}$$

Equation (24) cannot be regarded as a linear algebraic equation to determine $Q^{\nu J}$ as there is the additional requirement that the $Q^{\nu J}$ are non-singular for solutions of $\Delta = 0$. Instead, $\mathrm{Div}\, P$ is calculated and substitutions of a different form from before are made. For example, if conservation laws for the Harry Dym equation $0 = \Delta = u_t - u^3 u_{xxx}$ are investigated and if for the derivation of (1) there had been done substitutions $u_t = u^3 u_{xxx}$, $u_{tx} =$

$(u^3 u_{xxx})_x, \ldots$ before then now the substitutions would be $u_t = \Delta + u^3 u_{xxx}$, $u_{tx} = \Delta_x + (u^3 u_{xxx})_x, \ldots$ which provide the right hand side of (24). The computation of Q^ν and $P^i - R^i$ from P^i is part of `CONLAW1`.

4.4 Computing conserved currents from characteristic functions

The inverse computation is necessary in `CONLAW2` where the conserved current P^i has to be computed from Q^μ by integrating $\mathrm{Div}\, P = \sum_\nu Q^\nu \Delta_\nu$.

A direct way is based on a formula given by Anco & Bluman in [1]:

$$
\begin{aligned}
P^i &= \int_0^1 \frac{d\lambda}{\lambda} \left(S^i(u) + N^i_\mu(u) u^\mu + N^{ij}_\mu(u) D_j u^\mu + \ldots \right) |_{u \to \lambda u} && (26)\\
S^i(u) &= Q^\nu \frac{\partial \Delta_\nu}{\partial u^\mu_i} u^\mu + Q^\nu \frac{\partial \Delta_\nu}{\partial u^\mu_{ij}} u^\mu_j - \left(Q^\nu \frac{\partial \Delta_\nu}{\partial u^\mu_{ij}} \right)_j u^\mu + \ldots && (27)\\
N^i_\mu(u) &= \frac{\partial Q^\nu}{\partial u^\mu_i} \Delta_\nu - \left(\frac{\partial Q^\nu}{\partial u^\mu_{ij}} \Delta_\nu \right)_j + \left(\frac{\partial Q^\nu}{\partial u^\mu_{ijk}} \Delta_\nu \right)_{jk} - \ldots && (28)\\
N^{ij}_\mu(u) &= \frac{\partial Q^\nu}{\partial u^\mu_{ij}} \Delta_\nu - \left(\frac{\partial Q^\nu}{\partial u^\mu_{ijk}} \Delta_\nu \right)_k + \left(\frac{\partial Q^\nu}{\partial u^\mu_{ijkl}} \Delta_\nu \right)_{kl} - \ldots && (29)
\end{aligned}
$$

where summation is done over double indices.

These formulas are programmed in the more compact form

$$
\begin{aligned}
V &= Q^\nu \Delta_\nu,\\
W^i &= n(i) u^\mu \frac{\partial V}{\partial u^\mu_i} +\\
&\quad n(ij) \left(u^\mu_j - u^\mu D_j \right) \frac{\partial V}{\partial u^\mu_{ij}} +\\
&\quad n(ijk) \left(u^\mu_{jk} - u^\mu_j D_k + u^\mu D_j D_k \right) \frac{\partial V}{\partial u^\mu_{ijk}} +\\
&\quad \vdots\\
T^i &= x^i \int_0^1 d\lambda\, \lambda^{p-1} V|_{u \to 0, x \to \lambda x}\\
P^i &= T^i + \int_0^1 \frac{d\lambda}{\lambda} W^i|_{u \to \lambda u} && (30)
\end{aligned}
$$

where in W^i it is summed over equal indices (not the $i, j, k, \ldots$ in $n(i, j, \ldots)$) with $i \leq j \leq k \leq \ldots$ and $n(i, j, \ldots) = \prod_a r_a!/(\sum_b r_b)!$ with r_a being the multiplicities of different arguments $i, j, \ldots$ of n (e.g. $n(i) = 1, n(i, i) = 1, n(1, 2) = 1/2$) which also occur in (27) - (29). p is the number of variables x^i and T^i are non-zero only if $u \equiv 0$ does not solve $\Delta = 0$ (T^i have to enter (26) in that case as well).

Despite being an elegant formula there may be problems in computing the integral analytically. More seriously, the integral may be singular for $\lambda = 0, 1$. That is the case, for example, for the non-polynomial characteristic functions of the Harry-Dym equations in the next section.

If taking the limit $u \to 0$ in $V|_{u\to 0, x\to \lambda x}$ or taking the limit $u \to \lambda u$ in $W^i|_{u\to\lambda u}$ produces a singularity then this can be circumvented by choosing a $\tilde{u}(x)$ (e.g. $\tilde{u} = 1$ or $\tilde{u} = x$) such that after replacing $u \to \lambda u + (1 - \lambda)\tilde{u}$ the limits $V|_{u\to\tilde{u}, x\to\lambda x}$ and $W^i|_{u\to\lambda u+(1-\lambda)\tilde{u}}$ become non-singular [2].

Because of these potential difficulties, the default procedure to compute P^i is to use the integration module of `CRACK` to x^1-integrate $\sum_\nu Q^\nu \Delta_\nu$, to x^2-integrate the remaining unintegrated terms, and so on. In case terms remain after the last x^p-integration, the process is restarted on the remaining terms until all terms are integrated or at most a fixed number of times. If this method does not work because not all determining conditions can be solved as, for example, for the Burgers equation below then (30) is used.

If both methods fail then the program concludes that solutions Q of (4) represent adjoint symmetries and not characteristic functions of conservation laws.

4.5 The simplification of P in two variables

After deleting trivial conservation laws and identifying equivalent conservation laws through the computation of characteristic functions Q it remains to simplify the conserved current P through the addition of some curl: $P \to P + \text{curl } V$. This is done if there are only two independent variables, say x^1, x^2. The aim is to lower the order of x^2-derivatives in P^1 through changes $P^1 \to P^1 - D_2 R$, $P^2 \to P^2 + D_1 R$. R is found by repeated partial integration of terms in P^1 with highest x^2-derivatives of u. For that, partial integration routines of `CRACK` are used which are limited in applicability to expressions which are at most polynomially non-linear in u and derivatives of u.

5 Examples

Computation times refer to a 24 MB REDUCE 3.6 session under LINUX on a 133 MHz Pentium PC with the Jan. 1998 version of `CRACK`.

Example 1:

The advantage of using the package `CRACK` for solving determining equations is that they can be PDEs and do not have to be restricted to algebraic equations for coefficients of a polynomial ansatz for the conservation law. By that it is possible to find non-polynomial conservation laws and conservation laws that have an explicit x^i dependence. An example is the Harry Dym equation

$$\Delta = u_t - u^3 u_{xxx}, \quad u = u(t,x)$$

which was used below to substitute u_t and derivatives of u_t. These calculations were done with `CONLAW1`. In the following the calculation to find conservation laws with P^t of order 0 is described. The results for this case and the higher order cases are given afterwards.

If P^t is of order 0, i.e. $P^t = P^t(x,t,u)$, the condition (1): $\mathrm{Div}\, P = 0$ will involve u_t from $D_t P^t$. After substitution of u_t from the Harry Dym equation it will have u_{xxx} as the highest u-derivative which has to match the highest u-derivative in $D_x P^x$. Therefore, `CONLAW1` makes the ansatz $P^t = P^t(x,t,u), P^x = P^x(x,t,u,u_x,u_{xx})$ for conservation laws with P^t of order 0, then formulates the condition $\mathrm{Div}\, P = 0$ as

$$u_{xxx}(u^3 P^t_u + P^x_{u_{xx}}) + P^t_t + P^x_x + P^x_u u_x + P^x_{u_x} u_{xx} = 0 \tag{31}$$

and calls `CRACK` to solve it. As P^t, P^x do not depend on u_{xxx}, a separation performed by `CRACK` gives

$$0 = u^3 P^t_u + P^x_{u_{xx}} \tag{32}$$

$$0 = P^t_t + P^x_x + P^x_u u_x + P^x_{u_x} u_{xx}. \tag{33}$$

P^t does not depend on u_{xx}, so integration of (32) with respect to u_{xx} introduces a function of integration $g_1(x,t,u,u_x)$ and substitution of P^x in (33) gives an equation that can be separated with respect to u_{xx}. Two equations for $P^t(x,t,u)$ and $g_1(x,t,u,u_x)$ result. One of the equations can be integrated with respect to u_x with a function of integration $g_2(x,t,u)$ such that g_1 can be substituted in the other equation which then becomes separable with respect

to u_x, giving 4 equations. One of the four equations $3P^t_{xu} + uP^t_{xuu} = 0$ can be integrated 3 times to get $u^2P^t + g_3(t,u) + g_4(x,t) + u^2g_5(x,t)/2 = 0$. Substituting P^t in the other 3 equations yields among 2 other equations $g_{3,uuu} = 0$ which integrated and g_3 being substituted provides 2 equations. One of these equations can be u-integrated such that g_2 can be substituted in the other equation, which in turn becomes u-separable, giving 4 equations.

The computation so far needed 14 steps in `CRACK`. Continuing this process of integrations, substitutions, separations for another 14 steps gives the solution

$$\begin{aligned} P^t &= -(g_{17,x}\,u^2 + 2ug_{10} + 2xg_{14} + x^2g_{15} + 2g_{16} - g_7u^2)/(2u^2) \quad (34)\\ P^x &= (g_{17,t} - xg_{7,t} - 2uu_{xx}g_{10} + u_x^2g_{10} + 4u_xg_{14} - 4xu_{xx}g_{14} \\ &\quad -4ug_{15} + 4xu_xg_{15} - 2x^2u_{xx}g_{15} - 4u_{xx}g_{16})/2 \quad (35)\end{aligned}$$

with free $g_{17}(x,t), g_7(t), g_{10}, g_{14}, g_{15}, g_{16}$. The algorithm of section 4.1 finds g_7 to be redundant and sets it to zero which concludes the run of `CRACK`. In `CONLAW1`, the terms in the general solution returned by `CRACK` which involve g_7 are found to fulfill $\mathrm{Div}\,P = 0$ identically. Therefore this conservation law is trivial and is dropped. The coefficients of the remaining 4 constants give 4 non-trivial conservation laws. As described in section 4.3 the characteristic functions Q are computed which finally gives:
P^t of order 0: time to formulate (1): 0.32 sec, to solve (1): 1.34 sec, 28 steps:

$$\begin{aligned} 2u^{-2}\cdot\Delta &= D_t(-2u^{-1}) &+ D_x(u_x{}^2 - 2uu_{xx})\\ 2u^{-3}\cdot\Delta &= D_t(-u^{-2}) &+ D_x(-2u_{xx})\\ 2xu^{-3}\cdot\Delta &= D_t(-xu^{-2}) &+ D_x(2u_x - 2xu_{xx})\\ 2x^2u^{-3}\cdot\Delta &= D_t(-x^2u^{-2}) &+ D_x(4xu_x - 2x^2u_{xx} - 4u)\end{aligned}$$

Similar computations for higher orders give:
P^t of order 1: time to formulate (1): 0.32 sec, to solve (1): 2.6 sec, 49 steps:

$$(2uu_{xx} - u_x{}^2)u^{-2}\cdot\Delta =$$
$$D_t(-u_x{}^2u^{-1}) + D_x((2u_tu_x - u_{xx}{}^2u^3 + u_{xx}u_x{}^2u^2 - u_x{}^4u/4)u^{-1})$$

P^t of order 2: time to formulate (1): 1.7 sec, to solve (1): 110 sec, 193 steps[3]:

$$(-8u_{xxxx}u^3 - 16u_{xxx}u_xu^2 - 12u_{xx}{}^2u^2 + 12u_{xx}u_x{}^2u - 3u_x{}^4)u^{-2}\cdot\Delta =$$

[3]The times and the number of steps depend strongly on the relative priorities of the individual steps done (integrations, substitutions,...) which is especially the case for longer computations and therefore these measures characterize the complexity only roughly.

$$D_t((-4{u_{xx}}^2u^2 - 3u_{xx}{u_x}^5tu - {u_x}^4)u^{-1}) +$$
$$D_x((8u_{tx}u_{xx}u^2 + 3u_{tx}{u_x}^5tu - 8u_tu_{xxx}u^2 - 8u_tu_{xx}u_xu + 4u_t{u_x}^3 +$$
$$4{u_{xx}}^2u^5 + 4{u_{xx}}^3u^4 - 6{u_{xx}}^2{u_x}^2u^3 + 3u_{xx}{u_x}^4u^2)u^{-1})$$

Computations for conservation laws of higher order of equations $\Delta = 0$ of higher order in more variables x^i and more functions u^α are similar to the one described above. However, the application of integrability conditions may be necessary (like in a computation for a differential Gröbner basis) and a more general separation method may be necessary if no function depends on all variables and each variable occurs in at least one function. More difficult computations may involve several thousand steps.

Example 2:
The Burgers equation in the form

$$\Delta = u_t - u_{xx} - \frac{1}{2}u_x^2 = 0, \quad u = u(t,x) \tag{36}$$

is an example for the case that the determining equations cannot be solved completely. It has 0^{th} order conservation laws

$$fe^{u/2}\Delta = D_t(2fe^{u/2}) + D_x(e^{u/2}(2f_x - fu_x)) \tag{37}$$

with $f = f(t,x)$ satisfying the linear reverse heat equation $f_t + f_{xx} = 0$. The above law is also an example that `CONLAW` allows the computation of conservation laws with non-rational terms which is not possible with approaches based on a polynomial ansatz. A remaining linear PDE and the occurrence of free functions in the conservation law indicates linearizability of $\Delta = 0$, which is the case with the Burgers equation.

Example 3:
The MVDNLS equations (Modified Vector Derivative Nonlinear Schrödinger equations) describe oblique propagation of magneto-hydrodynamic waves in warm plasmas [8]. For functions $u = u(t,x)$, $v = v(t,x)$ and $b = \text{const.}$ they are

$$\Delta_1 = u_t + [u(u^2+v^2) + bu - v_x]_x \tag{38}$$
$$\Delta_2 = v_t + [v(u^2+v^2) + u_x]_x. \tag{39}$$

Both equations have the form of conservation laws. Using the abbreviations (introduced by hand afterwards)

$$\begin{aligned} E &= -v_x + u(u^2+v^2) \\ F &= u_x + v(u^2+v^2-b) \\ G &= 2u_{xx} + 6v_x(u^2+v^2) - 3u(u^2+v^2)^2 - 2bu^3 \\ H &= 2v_{xx} - 6u_x(u^2+v^2) - 3v(u^2+v^2)^2 + 2bv^3 \\ I &= b(u^4-v^4) + (u^2+v^2)^3 - 2u_x^2 - 2v_x^2 \end{aligned}$$

and using equations (38), (39) to substitute for u_t, v_t, further conservation laws calculated by `CONLAW2/3` have the characteristics $\{Q^1, Q^2\}$:

$$\{u, v\}, \quad \{E, F\}, \quad \{G, H\}, \tag{40}$$

$$\{(bt-2x)E - 2tG + b(bt-x)u + v, \ \ (bt-2x)F - 2tH + b(bt-x)v - u\}, \tag{41}$$

$$\{-H_x + 2uvH + (b+2u^2)G + uI, \ \ G_x + 2uvG + 2v^2H + vI\}. \tag{42}$$

`CONLAW2` can compute one more conservation law with Q^1, Q^2 of 4^{th} order and 36 terms each. Run times are listed in table 1.[4]

Apart from (41) these conservation laws are given in [8] where also a bi-Hamiltonian structure is provided. Although an infinite sequence of conserved densities can be calculated from the resulting recursion operator, the conservation law (41) is not contained in that sequence and is new - it has an explicit t, x-dependence.

In the scope of `CONLAW1` to find conservation laws with P^1 of order 1 are conservation laws (40),(41).

6 Comparison of the three methods

The determining equations (1)-(4) differ in the number of functions, number of variables and their order. For example, for the MVDNLS equations (38),(39) the condition (1) for conservation laws with P^1 of order 2 and the conditions (2),(4) for conservation laws with Q^μ of order 3 have the following characteristics:

[4] With the improved (Sep. 1998) version of `CRACK` times are reduced. For example, the 1977 sec are now 76 sec and the `CONLAW1` second order case can now be solved in 880 sec.

CONLAW	order of P^t for CONLAW1, order of Q for CONLAW2/3									
	0		1		2		3		4	
	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2
1	0.15	2.9	0.15	1977						
2	1.7	2.0	2.7	16	4.5	194	8.5	722	17	2784
3	0.17	4.5	0.18	11.7	0.3	28.5	0.6	377	1.9	low memory

Table 1: Run times t_1 to formulate and t_2 to solve determining conditions of conservation laws of the MVDNLS equations

(1): 1 condition in 12 variables $(t, x, u, v, u_x, v_x, \ldots, u_{4x}, v_{4x})$, 2 of which occur only explicitly (u_{4x}, v_{4x}), with 55 terms linear in functions P^t of 8 variables $(t, x, u, v, u_x, v_x, u_{xx}, v_{xx})$ and P^x of 10 variables $(t, x, u, v, \ldots, u_{xxx}, v_{xxx})$ and their 1st order derivatives. The difference in the dependencies of P^t, P^x at the beginning of CONLAW1 is a consequence of using (38), (39) for substituting a first order t-derivative of u by a second order x-derivative.

(2): 1 condition in 22 variables $(t, x, u, v, \ldots, u^{(3)}, v^{(3)})$, 6 of which occur only explicitly (2nd order derivatives of u_t, v_t), with 37 terms linear in functions P^t, P^x of 14 variables $(t, x, u, v, \ldots, u^{(2)}, v^{(2)})$ and their 1st order derivatives, and furthermore functions Q^1, Q^2 of 10 variables $(t, x, u, v, \ldots, u_{xxx}, v_{xxx})$.

(4): 2 coupled conditions in 14 variables $(t, x, u, v, u_x, v_x, \ldots, u_{5x}, v_{5x})$, 4 of which occur only explicitly $(u_{4x}, v_{4x}, u_{5x}, v_{5x})$, with 131 and 132 terms linear in functions Q^1, Q^2 of 10 variables $(t, x, u, v, \ldots, u_{xxx}, v_{xxx})$ and their 1st and 2nd order derivatives.

The following are general features of equations (1)-(4).
Equation (1) is of first order and therefore only highest order u-derivatives which are not substituted due to $\Delta = 0$ are not variables of the P^i and can be used for direct separation. Equation (1) therefore is only weakly over determined with the application of integrability conditions playing an important role. A general problem with computing a differential Gröbner Basis is that the complexity of these calculations depends heavily on the total ordering of derivatives of functions P, Q chosen for which there is currently no complete theory available. Choices made by the program can be particularly good or bad for the problem at hand.

In contrast, equations (4) are of higher order with more jet-variables that occur only explicitly and that can be used for direct separation. Although these equations are of higher order they are highly over determined and simpler to solve in general. An efficient way of doing direct separations and handling large equations is of importance for this approach.

Finally, in equations (2) the P^i depend initially on all jet-variables (apart from highest order u-derivatives), and also those substituted through $\Delta = 0$ on which the Q^μ do not depend. On the other hand, the Q^μ do depend on highest order u-derivatives initially. The efficiency in solving (2) therefore depends on the efficiency of a module for indirect separation, i.e. on a module for handling equations which have no function depending on all variables but which have also no variable occurring only explicitly so that no direct separation with respect to any variable is possible. Such a module is described in [9].

All techniques are used to solve the over determined system of all three approaches, only some are used more often in one approach than in the other.

There is another issue. If the order of derivatives with respect to different variables differs, such as for example the Harry Dym equation $u_t - u^3 u_{xxx} = 0$, then it matters whether this equation is used to do substitutions $u_t = u^3 u_{xxx}$ or $u_{xxx} = u_t/u^3$. Substituting u_t gives a lower increase in complexity when successively higher order ansätze for P or Q are made. On the other hand, one has to go to higher orders of P and Q to cover the same equivalence classes of conservation laws compared to substituting u_{xxx}. As equations (4) involve already higher order u-derivatives, a further increase could explode the size of (4) even more.

Another relation between (2) and (4) is that one could look at (4) as resulting from a differential-Gröbner-Basis calculation done with (2), with the aim to eliminate the P^i first. It is of course more efficient to exploit knowledge of the structure of (2) and to apply the Euler operator to write down (4) directly rather than to do the differential Gröbner Basis calculation step by step with (2). On the other hand, `CRACK` includes a number of modules to take advantage of special situations (e.g. to integrate exact PDEs or to recognize and solve PDEs that are ODEs for some partial derivatives and to solve them using `ODESOLVE` [6]). For a concrete problem it is very likely that there exists a quicker way to solve (2) than to first eliminate all P^i. The question of which of the `CONLAW` programs is best depends on the effectiveness of different submodules of the program `CRACK` which is used

to solve conditions (1)-(4). With the current version of CRACK (Jan. 1998), programs CONLAW1/3 are better for simpler conservation law problems while CONLAW2 is better for larger problems.

7 Syntax of CONLAW

Example: The input to find conservation laws with Q of order 0-4 for the MVDNLS equations (38),(39) is

```
depend u,x,t;
depend v,x,t;
conlaw2({{df(u,t) = - df( u*(u**2+v**2) + b*u - df(v,x) ,x),
          df(v,t) = - df( v*(u**2+v**2) + df(u,x)       ,x) },
         {u,v}, {t,x}
        },
        {0, 4, t, {}, {}});
```

In REDUCE lists are enclosed in { }. The input of CONLAWi (i=1,2,3) consists of two lists, of which the first encodes the PDE problem. It contains a list of equations with the derivative to be substituted on the left hand side, a list of functions and a list of independent variables. The second parameter to CONLAWi is a list that specifies the conservation laws to be computed. Its first two elements are the minimum and maximum order of P^1 in the case of CONLAW1 and the order of Q^μ in the case of CONLAW2/3. The third element is t or nil and specifies whether the conservation law may depend explicitly on the x^i or not. The fourth element is a list of functions to be determined in an ansatz made for P^i or Q^μ and the last element is a list of inequalities to be satisfied.

More details about investigating an ansatz is given in a manual file that comes with the three CONLAWi.RED files.

8 Summary

Supplied with subroutines to fix gauge freedom in differential expressions, the programs CONLAW1/2/3 proved to be an efficient tool for the computation of conservation laws of differential equations. Compared with other programs, (see for instance [3]) the programs CONLAWi show the following new features:

- By solving systems of over determined differential equations, it is possible to find conservation laws with non-polynomial or even non-rational P, Q.
- It is possible to find conservation laws with an explicit dependence of P, Q on the independent variables.
- There is no limit to the number of DEs nor the number of independent variables to be investigated for conservation laws, other than a limit through the complexity of computations.
- It is possible to determine values of parameters in the DE such that conservation laws exist (examples in [11]).
- For each of the programs `CONLAWi`, an ansatz for P^i and/or Q^μ can be input to specify conservation laws to be calculated.

Compared with the program of Göktaş and Hereman, `CONLAW` is able to find more general conservation laws and to make a definitive statement if local conservation laws do not exist and if the order is not too high to complete the computations.

The strength of the program described in [3] is to sometimes get a higher order that still can be handled by concentrating on polynomial conservation laws because the program only has to solve algebraic systems for coefficients of a polynomial ansatz. They were also able to extend applicability to differential-difference systems [4], [5].

The comparison of the three approaches (1)-(4) showed that each of them has advantages in special circumstances. It also serves as a comparison between using a general purpose program to find the quickest way of solving over determined PDE systems directly (`CONLAW1/3`) and an approach to derive integrability conditions by applying extra information about the structure of the PDE system (`CONLAW2`).

The programs including a manual and a test file are available via ftp from `ftp://ftp.maths.qmw.ac.uk/pub/tw`. The package will be submitted to the REDUCE network library.

9 Acknowledgement

The authors want to thank Malcolm MacCallum, George Labahn and the referees for comments on an earlier version of the manuscript. Also, Jan Sanders and Willy Hereman are thanked for discussions along a research visit of TW at CAN Netherlands/Amsterdam. Discussions with Winfried Neun during multiple visits at the Konrad Zuse Institut/Berlin are gratefully acknowledged. Finally, Frank Verheest, Willy Sarlet, Micheline Musette (Belgium) and the Relativity group at Hall University (England) provided PDEs to test the code.

References

[1] Anco, S.C. and Bluman, G. (1997). *Direct Construction of Conservation Laws from Field Equations* Phys. Rev. Let. **78**, no 15, 2869-2873.

[2] Anco, S.C. and Bluman, G. (1998). *Direct construction method for conservation laws of differential equations*, preprint.

[3] Göktaş, Ü. and Hereman, W. (1996) *Symbolic Computation of Conserved Densities for Systems of Nonlinear Evolution Equations.* Journal of Symbolic Computation, vol. 24, pp. 591-621 (1997).

[4] Göktaş, Ü., Hereman, W. and Erdmann, G. (1997) *Computation of conserved densities for systems of nonlinear differential-difference equations.*, Physics Letters A, vol. 236, pp. 30-38 (1997).

[5] Göktaş, Ü. and Hereman, W. (1998) *Computation of conserved densities for nonlinear lattices*, Physica D (1998) in press.

[6] MacCallum, MAH. (1988) *An Ordinary Differential Equation Solver for REDUCE* Proc. ISSAC'88, Springer Lect. Notes in Comp. Sc., 358, 196-205.

[7] Olver, P.J. (1986). *Applications of Lie Groups to Differential Equations.* Grad. Texts in Math. Berlin: Springer-Verlag.

[8] Willox, R., Hereman, W. and Verheest, F. (1995). *Complete Integrability of a Modified Vector Derivative Nonlinear Schrödinger Equation.* Physica Scripta **52**, 21-26.

[9] Wolf, T. and Brand, A. (1992) *The Computer Algebra Package CRACK for Investigating PDEs*, manual + software in the `REDUCE` network library.

[10] Wolf, T. (1996) *The program CRACK for solving PDEs in General Relativity* in *Relativity and Scientific Computing* Eds. F.W.Hehl, R.A.Puntigam, H.Ruder, Springer, 241-257.

[11] Wolf, T. *A comparison of four approaches to the calculation of conservation laws*, in preparation.