



THOMAS WOLF<sup>1</sup>

# **The integration of systems of linear PDEs using conservation laws of syzygies**

---

<sup>1</sup>Department of Mathematics, Brock University, St.Catharines, Canada and ZIB Fellow



# The integration of systems of linear PDEs using conservation laws of syzygies

Thomas Wolf

Department of Mathematics  
Brock University, 500 Glenridge Avenue,  
St.Catharines, Ontario, Canada L2S 3A1  
email: twolf@brocku.ca

February 15, 2002

## 1 Introduction

The algorithm reported in this article is a qualitative improvement of the symbolic integration algorithm for exact partial differential equations (PDEs) discussed in [13], here specialized for the case of *linear systems* of PDEs. To motivate it we first look at an algebraic analogue.

Let us consider two polynomial equations

$$0 = y^2 + xy + y + x \quad (= e_1) \quad (1)$$

$$0 = xy - 5y + x^2 - 5x \quad (= e_2). \quad (2)$$

To check their compatibility one typically computes a Gröbner basis using, for example, a lexicographical ordering  $>_o$  of monomials with, say,  $y >_o x$ . Eliminating  $y^2$  first, we would get

$$0 = (x - 5)e_1 - ye_2 \quad (3)$$

$$= xy - 5y + x^2 - 5x \quad (= e_3). \quad (4)$$

In eliminating  $y$  between  $e_2$  and  $e_3$  the next step gives an identity  $0 \equiv e_3 - e_2$  and further after substituting  $e_3$  by  $(x - 5)e_1 - ye_2$  from (3)

$$0 = (x - 5)e_1 - (y + 1)e_2. \quad (5)$$

Equation (5) is a relation between the equations  $e_1, e_2$  and is satisfied identically in  $x, y$ . The Gröbner basis computation stops. It did not result in a lower degree equation, only in the identity (5), also called syzygy, which came as a by-product for free. In this syzygy the coefficients  $x - 5, y + 1$  are prime. Because  $e_1, e_2$  are polynomials consequently  $e_1$  must have  $y + 1$  as a factor and  $e_2$  must have  $x - 5$  as a factor and

$$e_4 := e_1/(y + 1) = e_2/(x - 5) \quad (= x + y)$$

must be a polynomial as well. The single equation  $0 = x + y$  is equivalent to the system of equations (1), (2). Therefore the Gröbner basis computation also gave us the greatest common divisor (GCD) of (1) and (2) for free. It would be computationally much more expensive to compute the GCD by

factorizing (1) or (2) and checking all factors. Also, case distinctions that follow could be intractable if many equations with many factors are involved.

The purpose of the present paper is to formulate and use syzygies for systems of linear PDEs. The computation of an equivalent of a GCD for linear partial differential operators will save us their factorization which is otherwise only possible algorithmically in special cases. Furthermore the factorization need not be unique [4, 5, 9]. But even if the differential operator of one equation can be fully factorized and the equation be integrated and used for a substitution of a function it still would be less advantageous to do so. In the algebraic analogue,  $e_1$  is of higher degree than  $e_4$  and the factorization of  $e_1$  leads to more factors (that have to be checked afterwards) than the factorization of the GCD of  $e_1$  with another equation. Similarly, the integration of a PDE gives more functions of integration or functions of more variables than the integration of the differential GCD of both PDEs. This will be demonstrated in an example below. Therefore, even if each of the equations can be integrated fully, the computation may blow up if the equations are of high order and have many independent variables. In such a case an excessive number of functions of integration result depending on many variables. With the method below fewer functions of integration depending on fewer variables arise.

After showing the computation with the new and the traditional method and comparing both in the next three sections, the algorithm is explained in general and an overview is given in section 5. A similar, but different version of the algorithm is presented in section 6. In the following 3 sections technical aspects are discussed: freedom within the new method, cases when it is not useful and some implementation issues. The final section provides an application that led to the development of syzygy based integrations. Comments are made about merging integration and integrability tests in one computer program.

## 2 An introductory example

With the following simple problem the qualitative differences between the new method to integrate equations of a differential Gröbner basis and the conventional method of integration are explained. A more complex example is given in section 10.

In applying integrability conditions for PDEs systematically, i.e. in computing a differential Gröbner basis, identities between equations  $0 = e_a$  will result that take the form of differential expressions with the  $e_a$  as dependent variables.

Let us consider the simple system

$$0 = f_{,yzz} \quad (=: e_1) \quad (6)$$

$$0 = f_{,x} + f_{,z} \cdot \quad (=: e_2) \quad (7)$$

Assuming, for example, a total ordering  $>_o$  of derivatives that implies  $\partial_x >_o \partial_z$  and  $\partial_y >_o \partial_z$ , a differential Gröbner Basis computation would first eliminate  $f_{,xyz}$  through cross-differentiation:

$$0 = e_{2,yzz} - e_{1,x} = f_{,yzzz} \quad (=: e_3) \quad (8)$$

then a substitution of  $f_{,yzz}$  using  $e_1$  yields

$$0 = e_3 - e_{1,z}$$

and a substitution of  $e_3$  using (8) provides the identity

$$0 = e_{2,yzz} - (e_{1,x} + e_{1,z}). \quad (9)$$

In this paper we concentrate ourselves to the integration of syzygies, like (9), which either have the form of a divergence or can be combined linearly to give a divergence  $0 = D_i P^i$  with suitable vector components  $P^i(e_k)$  that are differential expressions in the  $e_k$ . Only in section 6 we outline a variation of this principle to deal with a curl of syzygies. In order to find a combination of syzygies that is a divergence one could apply computer algebra programs CONLAW as described in [11], [14] by regarding the syzygies as the equations and the  $e_a$  as unknown functions. A discussion why the computation of conservation laws for syzygies is a relatively simple task can be found in appendix A.

One way to write the identity (9) as a divergence (with only two terms) is

$$0 = -e_{1,x} + (e_{2,yz} - e_1)_{,z} \quad (10)$$

$$= P^x_{,x} + P^z_{,z} \quad (11)$$

$$= (-f_{,yzz})_{,x} + (f_{,xyz})_{,z}. \quad (12)$$

In the following we will use the vector  $P^i$  in two representations, first in terms of  $e_i$ , in our example from the syzygy (10):

$$P^x = -e_1, \quad P^z = e_{2,yz} - e_1 \quad (13)$$

and second the representation of  $P^i$  in terms of the function  $f$ , in our example from the identity (12):

$$P^x = -f_{,yzz}, \quad P^z = f_{,xyz}. \quad (14)$$

With  $P$  satisfying the conservation law condition (11) we can write  $P$  as a 2-dim. curl

$$P^x = -Q_{,z}, \quad P^z = Q_{,x} \quad (15)$$

for some potential  $Q$ . Using for  $P^i$  the representation (14) we identify

$$Q = f_{,yz}.$$

The existence of differential expressions in unknowns  $f^\alpha$  for the potential  $Q$  is guaranteed because all syzygies and all their consequences like  $0 = D_i P^i$  are satisfied identically for any  $f^\alpha$ . In appendix B an algorithm DIVINT is given that computes potentials  $Q^{ij}(f^\alpha)$  in general for an arbitrary number of independent variables.

To do the next step in this example, we are reminded that expressions  $P^i(e_j)$  are linear homogeneous in the  $e_j$  and that they therefore must be zero, i.e.  $P^x = P^z = Q_{,x} = Q_{,z} = 0$ . This means that  $Q$  is independent of  $x, z$ , giving  $Q = c_1(y)$  and the new equation

$$0 = Q - c_1 = f_{,yz} - c_1 \quad (= e_4) \quad (16)$$

with the new function of integration  $c_1 = c_1(y)$ .

Apart from the integral (16) we also get new syzygies. Having on one hand expressions for  $P^i$  in terms of  $e_1, e_2$  due to equations (13) and on the other hand  $P^i$  in terms of  $Q_{,j}$  from equations (15) and  $Q$  in terms of  $e_4$  from equation (16) we get two new identities

$$0 = P^x + Q_{,z} = -e_1 + e_{4,z} \quad (17)$$

$$0 = P^z - Q_{,x} = e_{2,yz} - e_1 - e_{4,x}. \quad (18)$$

As equation  $e_1$  turns up algebraically in at least one of the new identities, this equation  $0 = e_1$  is redundant and can be dropped. Redundancy of an original equation due to integration need not always be the case but it is the case in this example because at least one of  $P^x$  and  $P^z$  happens to

be algebraic in  $e_1$  (in this case both). Substituting  $e_1$  from identity (17) in (18) gives a new identity in the form of a divergence

$$0 = (-e_4)_{,x} + (e_{2,y} - e_4)_{,z}$$

allowing the same integration step again *without having to do a differential reduction or cross differentiation step*. Proceeding as above we now identify as a new potential  $\bar{Q}$

$$\bar{Q} = -f_{,y} + (z - x)c_1$$

giving the new equation

$$0 = -\bar{Q} - c_2 = f_{,y} - (z - x)c_1 - c_2 \quad (=: e_5) \quad (19)$$

with the new function of integration  $c_2 = c_2(y)$ . Now, equation  $e_4$  is redundant and substitution gives the new identity

$$0 = e_{2,y} - e_{5,x} - e_{5,z} \quad (20)$$

which is a divergence as well but now in three variables. In general, the solution of  $0 = D_i P^i$  for  $P^i$  is given in terms of potentials  $Q^{ij} = -Q^{ji}$  through

$$P^i = D_j Q^{ij}$$

using the summation convention for identical indices. If the divergence involves only two differentiation variables like in equation (11) then there is only one component  $Q = Q^{zx} = -Q^{xz}$ . If the divergence involves three differentiation variables like in equation (20) then there are three potentials  $Q^{xy}, Q^{yz}, Q^{zx}$ . In that case the requirement  $0 = P^i = D_j Q^{ij}$  (based on the  $P^i$  being homogeneous in the  $e_k \rightarrow P^i = 0$ ) is solved through

$$Q^{ij} = R^{ijk}_{,k}$$

with free functions  $R^{ijk}(x^n) = R^{[ijk]}(x^n)$ . In three dimensions this introduces one new function  $R^{xyz}$  of all variables through  $Q^{xy} = R^{xyz}_{,z}$ ,  $Q^{yz} = R^{xyz}_{,x}$ ,  $Q^{zx} = R^{xyz}_{,y}$ .

By performing syzygy based integration again we would solve the remaining equations  $0 = e_2 = e_5$  (i.e. (7),(19)) for one function  $f$  but also introduce one new function  $R^{xyz}$  of all variables and therefore not make real progress. This is demonstrated in the first example in section 8. The remaining equation  $0 = e_5$  (i.e. (19)) is now better solved directly by introducing two new functions  $c_3(y), c_4(y)$  through  $c_1 = c_{3,y}$ ,  $c_2 = c_{4,y}$  and one new function  $c_5 = c_5(x, z)$  to obtain

$$f = (z - x)c_3 + c_4 + c_5 \quad (21)$$

with the remaining equation (7) now taking the shape

$$0 = c_{5,x} + c_{5,z} \quad (22)$$

Although this equation (22) is easily solved, in general this needs not to be the case. Hence, for the purpose of our example, we leave it as it is.

### 3 The same example in a conventional treatment

For comparison, we solve the system (6), (7) again, this time in the conventional direct way. After integrating (6) to

$$f = c_1(x, y) + zc_2(x, y) + c_3(x, z) \quad (23)$$

and substitution of  $f$  the equation (7) reads

$$0 = c_1(x, y)_{,x} + z c_2(x, y)_{,x} + c_3(x, z)_{,x} + c_2(x, y) + c_3(x, z)_{,z}. \quad (24)$$

In equation (24) there is no function that does depend on all variables and each variable does occur in at least one function. An algorithm for such ‘indirectly separable equations’ (ISEs) is contained in the package `CRACK` (see [10] and sub-section 10.2). These equations undergo a series of differentiations and divisions (producing a list of divisors)

- to eliminate all functions of some variable,
- to do a direct separation with respect to this variable, and
- to use the same list of divisors now in reverse order as integrating factors to back-integrate the equations which resulted from direct separation.

In the case of equation (24) a single  $y$ -differentiation eliminates  $c_3$  and allows a direct  $z$  separation (as  $c_1, c_2$  are independent of  $z$ ) giving  $0 = c_2(x, y)_{,xy}$ ,  $0 = c_1(x, y)_{,xy} + c_2(x, y)_{,y}$  and through back-integration with respect to  $y$  further

$$0 = c_2(x, y)_{,x} + c_4(x) \quad (25)$$

$$0 = c_1(x, y)_{,x} + c_2(x, y) + c_5(x) \quad (26)$$

$$0 = c_3(x, z)_{,x} + c_3(x, z)_{,z} - z c_4(x) - c_5(x) \quad (27)$$

with new functions of integration  $c_4, c_5$ . Renaming  $c_4 = c_6(x)_{,xx}$ ,  $c_5 = c_7(x)_{,x}$  and integrating equations (25), (26) gives

$$c_2 = -c_6(x)_{,x} - c_8(y) \quad (28)$$

$$c_1 = c_6(x) + x c_8(y) - c_7(x) + c_9(y) \quad (29)$$

$$0 = c_3(x, z)_{,x} + c_3(x, z)_{,z} - z c_6(x)_{,xx} - c_7(x)_{,x} \quad (30)$$

$$f = (x - z) c_8(y) + c_9(y) + c_3(x, z) + c_6(x) - z c_6(x)_{,x} - c_7(x). \quad (31)$$

The solution (31) is identical to (21) and the remaining condition (30) is identical to (22) if we drop the redundant functions  $c_6, c_7$  which can be absorbed by  $c_3$ . A method to recognize redundancy is described in [14]. It involves the solution of an over-determined system of equations which involves even more effort.

## 4 Advantages of the integration using syzygies

Comparing both treatments above we make the following observations.

- Syzygies that are utilized by the new method come for free when computing a differential Gröbner basis.
- In the new method the computed integrals are necessary and sufficient not only for one integrated equation but for all  $P^i = 0$  which occur in  $0 = D_i P^i$ . This means all the conditions  $P^i = 0$  are integrated *at once* which has the consequence that functions of integration have less variables to depend on (because more conditions are satisfied by the integral) compared to the case that just one equation is integrated at a time. For example, the function of integration  $c_1(y)$  in (16) does depend only on one variable whereas integrating equation (6) conventionally with the integral (23), all *three* functions of integration  $c_1, c_2, c_3$  depend on *two* variables.

- Integrations based on syzygies are motivated through the structure of the system of equations and not just by the shape of a single equation. As a consequence the final solution contains either no redundant free functions or at least less redundant free functions compared to the conventional integration. For example, the solution (22),(23) has no redundant functions but the solution (30), (42) has. This difference is significant. Identifying redundant functions and constants involves the solution of an over-determined system of PDEs and may be computationally expensive (see [14]).
- New syzygies such as (17),(18) satisfied by the new equations, like (16), are instantly available and do not require further checking of integrability conditions. It may be possible that these new syzygies can be solved for one of the  $e_k$  that may occur algebraically in one of the  $P^i$  and that  $e_k$  is substituted in other syzygies which then obtain a divergence form. This was demonstrated in the example above.

## 5 The algorithm in general

In our notation  $x^i$ ,  $i = 1, \dots, p$  are the independent variables and  $f^\alpha$  are the unknown functions which do not need to depend on all  $x^i$ . These functions satisfy equations  $0 = e_a(x^n, f_J^\alpha)$  where  $J$  is a multi-index (standing, for example, for  $_{112}$ , i.e.  $\partial_{x_1}^2 \partial_{x_2}$ ) and where  $f_J^\alpha$  stands for a possible dependence on  $f^\alpha$  and any partial derivatives of  $f^\alpha$ . Total derivatives appear as  $D_i$ . Summation is performed over identical indices. In the following description we refer to individual steps shown in the overview underneath.

For a given system of differential equations (32) the investigation of integrability conditions (e.g. Gröbner basis computation) yields identities (33), called syzygies. In these syzygies the  $e_k$  take the role of dependent variables. The program CRACK has been used to compute syzygies for examples presented in this paper but many other computer algebra programs are available (for example, RIF [8], diffalg [1],[2],[3], diffgrob2 [6]) although only few generate syzygies automatically.

Either by hand, or by using the package CONLAW [11] or other computer algebra software one can try to find a conservation law (34) for the syzygies. In the conservation laws as in the syzygies the dependent variables are the  $e_k$ . Most often syzygies are very simple expressions and already have a conservation law form. Computing conservation laws is not fully algorithmic but it is argued in appendix A that this task is relatively simple for under-determined systems of syzygies.

In the computed conserved currents  $P^i(x, e_k)$  we replace the equation names  $e_k$  by their expressions (32) in terms of  $x, f^\alpha$ . The resulting  $P^i(x, f^\alpha)$  in (35) are the input to the algorithm DIVINT (given in appendix B) to compute a special solution for the potentials  $Q^{ij} = Q^{[ij]}(x, f^\alpha)$  satisfying  $P^i = D_j Q^{ij}$  shown in (36). This works because the kernel of a divergence  $D_i P^i$  is a curl  $D_j Q^{ij}$  with  $Q^{ij} = -Q^{ji}$ .

Because the syzygies  $0 = \Omega_m(x, e_k)$  are linear homogeneous expressions in the  $e_k$ , therefore  $D_i P^i$  as a linear homogeneous expression in the  $\Omega_m$  is a linear homogeneous expression in the  $e_k$  as well. Hence the  $P^i$  are linear homogeneous expressions in the  $e_a$ . Consequently, we have  $0 = P^i$  in the space of solutions of the original equations. On the other hand, the algorithm DIVINT computes expressions  $Q^{ij}$  satisfying  $P^i = D_j Q^{ij}$  identically and therefore  $0 = D_j Q^{ij}$  in the space of solutions of the original equations. The general solution of this condition for the  $Q^{ij}$  is shown in (38) and is the result of the whole computation. Its form depends on the number  $p$  of variables  $x^i$ : for  $p = 2$  a single constant of integration  $R$  is introduced for  $p > 2$  one or more functions  $R^{ijk}(x)$  are introduced.

The formal integration of  $0 = D_j Q^{ij}$  gives new equations whose right hand sides are abbreviated by  $e_{ij}$  in (39). We are instantly able to formulate syzygies which these new equations  $0 = e_{ij}$  satisfy, as shown in (40). If any one of them can be solved for one  $e_m$  (as indicated in (41)) then  $e_m = \omega$



can be substituted in other syzygies and the original equation  $0 = e_m(x, f^\alpha)$  can be deleted as it is a consequence of the equations  $e_k, e_{ij}$  in  $\omega(x, e_k, e_{ij,j})$ .

$$\text{Given system:} \quad 0 = e_k(x, f^\alpha) \quad (32)$$

$$\text{CRACK} \rightarrow \text{Syzygies:} \quad 0 = \Omega_m(x, e_k) \quad (33)$$

$$\text{CONLAW} \rightarrow \text{Cons. law form:} \quad 0 = D_i P^i(x, e_k), \quad (34)$$

$$\text{Conserved current:} \quad P^i = P^i(x, e_k)|_{e_k \rightarrow e_k(x, f^\alpha)} = P^i(x, f^\alpha) \quad (35)$$

$$\text{DIVINT} \rightarrow \text{New potentials:} \quad P^i(x, f^\alpha) = D_j Q^{ij} \quad \text{with} \quad Q^{ij} = Q^{[ij]}(x, f^\alpha) \quad (36)$$

$$\text{Integration of:} \quad 0 = P^i = D_j Q^{ij} \quad (37)$$

$$\text{to new integral(s):} \quad Q^{ij}(x, f^\alpha) = \begin{cases} R = \text{const} & \text{in } 2 \text{ dim} \\ R^{ijk},{}_k \text{ with } R^{ijk} = R^{[ijk]}(x) & \text{in } >2 \text{ dim} \end{cases} \quad (38)$$

$$\text{New equation names:} \quad 0 = \left\{ \begin{array}{l} Q^{ij}(x, f^\alpha) - R \\ Q^{ij}(x, f^\alpha) - R^{ijk},{}_k \end{array} \right\} =: e_{ij} \quad (39)$$

$$\text{New syzygies:} \quad \rightarrow 0 = P^i(x, e_k) - e_{ij,j} \quad (40)$$

$$\text{Redundancies?} \quad e_m = \omega(x, e_k, e_{ij,j}) \rightarrow \quad (41)$$

– substitution of  $e_m = \omega$  in any syzygy

– deleting equation  $0 = e_m$

return to start

## 6 An integration based on curls of syzygies

The described ansatz of extracting information out of syzygies in order to do integrations is not the only possible way. In this section we want to provide a different integration method, this time based on curls of syzygies. We will see that it is even more effective than divergence based integration but the required structure of the system of syzygies is more special which is the reason why it has not been implemented in CRACK yet. Also, the computation of conservation laws for syzygies was implemented so far because computer programs, like CONLAW, are available to compute conservation laws and because the existence of conservation laws is a relative weak condition for syzygies. The method based on curls is shown in the following overview.

Given system:	$0 = e_k(x, f^\alpha)$
Syzygies:	$0 = \Omega_m(x, e_k)$
Vanishing curl cond.:	$0 = D_j P^{ij}$ with $P^{ij} = P^{[ij]}(x, e_k)$ ,
Curl free tensor:	$P^{ij} = P^{ij}(x, e_k) _{e_k \rightarrow e_k(x, f^\alpha)} = P^{ij}(x, f^\alpha)$
New potentials:	$P^{ij}(x, f^\alpha) = D_k Q^{ijk}$ with $Q^{ijk} = Q^{[ijk]}(x, f^\alpha)$
Integration of:	$0 = P^{ij} = D_k Q^{ijk}$
to new integral(s):	$Q^{ijk}(x, f^\alpha) = \begin{cases} R = \text{const} & \text{in 3 dim} \\ R^{ijkl}_{,l} \text{ with } R^{ijkl} = R^{[ijkl]} & \text{in } >3 \text{ dim} \end{cases}$
New equation names:	$0 = \left\{ \begin{array}{l} Q^{ijk}(x, f^\alpha) - R \\ Q^{ijk}(x, f^\alpha) - R^{ijkl}_{,l} \end{array} \right\} =: e_{ijk}$
New syzygies:	$\rightarrow 0 = P^{ij}(x, e_a) - e_{ijk,k}$
Redundancies?	$e_m = \omega(x, e_k, e_{ijk,k}) \rightarrow$ substitution of $e_m$ – substitution of $e_m = \omega$ in any syzygy – deleting equation $0 = e_m$
	return to start

The superficial difference between divergence and curl based integration is that  $P, Q, R$  have one extra index for the curl based method. This method also needs at least 3 independent variables. The following two examples involve each 4 independent variables and allow a closer comparison of both methods.

*A typical example:*

For 4 unknown functions  $a, b, c, d$  depending on  $x, y, z, t$  a system of 6 equations

$$\begin{array}{lll} 0 = d_{,z} - c_{,t} \quad (= e_{xy}), & 0 = b_{,t} - d_{,y} \quad (= e_{xz}), & 0 = c_{,y} - b_{,z} \quad (= e_{xt}) \\ 0 = d_{,x} - a_{,t} \quad (= e_{yz}), & 0 = a_{,z} - c_{,x} \quad (= e_{yt}), & 0 = b_{,x} - a_{,y} \quad (= e_{zt}) \end{array}$$

is given. It has syzygies

$$\begin{array}{l} 0 = e_{xy,y} + e_{xz,z} + e_{xt,t} \\ 0 = -e_{xy,x} + e_{yz,z} + e_{yt,t} \\ 0 = -e_{xz,x} - e_{yz,y} + e_{zt,t} \\ 0 = -e_{xt,x} - e_{yt,y} - e_{zt,z} \end{array}$$

which take the form of a vanishing curl:  $0 = D_j P^{ij}$  for  $P^{ij} = e_{ij}$  leading to potentials  $Q^{ijk}$

$$Q^{xyz} = d, \quad Q^{txy} = c, \quad Q^{xzt} = b, \quad Q^{ytz} = a$$

and a new free function of integration  $R^{xyzt} = g(x, y, z, t)$ . The resulting integrals are

$$a = g_{,x}, \quad b = g_{,y}, \quad c = g_{,z}, \quad d = g_{,t}.$$

*A related example for a conservation law syzygy:*

In comparison, the typical example using a conservation law syzygy in 4 independent variables would involve 6 unknown functions  $a, b, c, d, f, g$  and 4 equations, so a less over-determined system:

$$\begin{array}{ll} 0 = a_{,y} + b_{,z} + c_{,t} \quad (= e_1), & 0 = -a_{,x} + d_{,z} + f_{,t} \quad (= e_2) \\ 0 = -b_{,x} - d_{,y} + g_{,t} \quad (= e_3), & 0 = -c_{,x} - f_{,y} - g_{,z} \quad (= e_4). \end{array}$$

The conservation law  $0 = e_{1,x} + e_{2,y} + e_{3,z} + e_{4,t}$  gives  $P^i = e_i$  and potentials

$$Q^{xy} = a, \quad Q^{xz} = b, \quad Q^{xt} = c, \quad Q^{yz} = d, \quad Q^{yt} = f, \quad Q^{zt} = g.$$

The resulting integrals are

$$a = r_{,z} - s_{,t} , \quad b = u_{,t} - r_{,y} , \quad c = s_{,y} - u_{,z} , \quad d = r_{,x} - w_{,t} , \quad f = w_{,z} - s_{,x} , \quad d = u_{,x} - w_{,y}$$

with new arbitrary functions  $r, s, u, w$ .

If both methods would be applicable, i.e. if the system of syzygies would provide a vanishing divergence and a vanishing curl then one would prefer the curl based integration because it makes use of more syzygies.

The last two examples look very artificial but one could exchange the unknown functions  $a, b, c, \dots$  by any functionally independent expressions, each of them involving at least one different function, and the computations and results would be unchanged.

The remainder of the paper is concerned with divergence based integration.

## 7 Choosing between different syzygy conservation laws

The integration of a syzygy  $0 = D_i P^i$  with two derivatives  $0 = D_x P^x + D_y P^y$  is always useful but not necessarily the integration of a syzygy with more than 2 derivatives (see the example in section 8). Sometimes there is a choice allowing to write a syzygy in different forms, for example

$$0 = e_{1,x} + (e_{2,x})_y + e_{3,z}$$

can also be written as

$$0 = (e_1 + e_{2,y})_{,x} + e_{3,z} .$$

To find out whether a conservation law with fewer derivatives exists one has two options. First, one can make an ansatz for the conservation law with fewer derivatives and solves the resulting conditions (for example, with the programs CONLAW1 or CONLAW3). Alternatively, one computes the most general conservation law involving arbitrary functions. If a conservation law exists which does *not* contain derivatives  $D_j P^j$ ,  $j = m, \dots, p$  then  $0 = D_j (C P^j)$ ,  $j = 1, \dots, m - 1$  is a conservation law with an arbitrary function  $C = C(x^m, \dots, x^p)$ . Reversely, finding a conservation law involving an arbitrary function  $C(x^m, \dots, x^p)$  can be exploited to derive a conservation law involving no derivatives with respect to  $x^m, \dots, x^p$  as it is described in [12].

## 8 Cases when a syzygy integration is not useful

To apply the new integration method it not only matters whether all steps are algorithmic but also whether its execution is beneficial. Syzygy information is useful if it provides a factorization of differential operators. If they do not factorize (for example, if they are of first order) then a syzygy based integration can still be useful if more functions are solved for than new functions are introduced. If the divergence  $D_i P^i$  contains more than two derivatives, i.e. the conserved current  $P^i$  has more than 2 components, then the integral equations (38) contain at least one new function  $R^{ijk}$  of all variables and we may not gain new information from the integration if we can not solve for at least 2 functions. This is demonstrated in the following series of 3 examples with successively more functions to be solve for and an increasing usefulness of the integration.

*Example:*

When computing the Gröbner basis of the two equations

$$0 = f_{,x} + f_{,y} \quad (=: e_1) \tag{42}$$

$$0 = f_{,z} \quad (=: e_2) \tag{43}$$

for a function  $f = f(x, y)$  (and in doing that confirming that they are already a Gröbner basis) one will generate the identity

$$0 = e_{2,x} + e_{2,y} - e_{1,z}. \quad (44)$$

From identifying  $P^x = e_2$  from (44) and the general formula  $P^x = D_y Q^{xy} + D_z Q^{xz}$  together with (43) we identify  $Q^{xy} = 0$ ,  $Q^{xz} = f$ ,  $Q^{yz} = f$ . With the new function  $R^{xyz} = c(x, y, z)$  substituted into the formula  $Q^{ij} = R^{ijk},_k$  the new equations are

$$0 = c_{,z} \quad (45)$$

$$0 = f - c_{,x} \quad (46)$$

$$0 = f + c_{,y}. \quad (47)$$

After a substitution of  $f$  from (46) into (47) they are identical to the original set (42), (43), only now for a function  $c$  instead of  $f$ . No progress was made. In contrast, for the following two similar examples the integration of syzygies is advantageous.

*Example:*

For the equations

$$0 = f_{,x} + g_{,y} \quad (= e_1) \quad (48)$$

$$0 = f_{,z} \quad (= e_2) \quad (49)$$

$$0 = g_{,z} \quad (= e_3) \quad (50)$$

the identity

$$0 = e_{2,x} + e_{3,y} - e_{1,z} \quad (51)$$

results. Integrated in the above manner it gives

$$0 = c_{,x} + g \quad (52)$$

$$0 = -c_{,y} + f \quad (53)$$

$$0 = c_{,z} \quad (54)$$

leaving only equation (54) for  $c = c(x, y, z)$  to be solved, an improvement compared to the original system (48) – (50). In the next example no equations remain to be solved.

*Example:*

For the equations

$$0 = h_{,y} - g_{,z} \quad (= e_1) \quad (55)$$

$$0 = f_{,z} - h_{,x} \quad (= e_2) \quad (56)$$

$$0 = g_{,x} - f_{,y} \quad (= e_3) \quad (57)$$

the identity

$$0 = e_{1,x} + e_{2,y} + e_{3,z} \quad (58)$$

leads to

$$0 = f + c_{,x} \quad (59)$$

$$0 = g + c_{,y} \quad (60)$$

$$0 = h + c_{,z} \quad (61)$$

with an arbitrary function  $c = c(x, y, z)$  and no remaining equation.

In order to incorporate this method of integration into a general program for solving over-determined systems the usefulness of integration has to be judged automatically based on the number of derivatives in the divergence and the number of functions solved for. But also other adjustments to the whole program have to be made. These are discussed in the following short section.

## 9 Implementation

Apart from the implementation of the algorithm DIVINT as shown in appendix B, also changes to the package CRACK were needed in order to automate syzygy based integrations. When checking integrability conditions in a Gröbner basis computation the program had to keep track of any resulting identities (syzygies). This was done in the following way.

To each equation, for example  $e_3$  in (4), we will assign not only a value, like  $xy - 5y + x^2 - 5x$ , but also, what we will call a ‘history-value’ or short ‘history’, i.e.  $(x - 5)e_1 - ye_2$ . This history of an equation expresses one equation in terms of other equations, i.e. how it was historically computed doing the algebraic or differential Gröbner basis computation. At the beginning the history of each equation  $e_a$  is  $e_a$  itself. Whenever a new equation is computed then not only its value but also its history is calculated. For example, when in the initial example  $y$  is eliminated from equations (2) and (4) then a new equation  $0 = e_4$  is generated where  $e_4$  has the value 0 (as all terms cancel) and has the history value  $e_3 - e_2$  where  $e_3$  and  $e_2$  are replaced by their history values. The history of  $e_2$  is  $e_2$  whereas the history of  $e_3$  is  $(x - 5)e_1 - ye_2$  giving for  $e_4$  the history  $(x - 5)e_1 - (y + 1)e_2$  as is shown in (5).

The paper concludes with a larger application.

## 10 The application that led to the development of the syzygy based integration

### 10.1 The problem

A problem introduced to the author by Stephen Anco concerns the computation of all conservation laws of the radial SU(2) chiral equation in 2 spatial dimensions where the integrating factors are of at most 2<sup>nd</sup> order. The equation can be written as a first order system for two 3-component vectors  $\mathbf{j}(r,t)$ ,  $\mathbf{k}(r,t)$ :

$$\mathbf{k}_{,t} = \mathbf{j}_{,r} + \mathbf{j} \times \mathbf{k} \quad (62)$$

$$\mathbf{j}_{,t} = (r\mathbf{k})_{,r} / r. \quad (63)$$

Equation (63) is already in conservation law form:

$$(r\mathbf{j})_{,t} + (-r\mathbf{k})_{,r} = 0$$

and the only other known conservation law (of energy) has zeroth order integrating factors:

$$r\mathbf{k} \cdot [\mathbf{k}_{,t} - \mathbf{j}_{,r} - \mathbf{j} \times \mathbf{k}] + \mathbf{j} \cdot [\mathbf{j}_{,t} - (r\mathbf{k})_{,r} / r] = \left( \frac{r}{2} (\mathbf{j} \cdot \mathbf{j} + \mathbf{k} \cdot \mathbf{k}) \right)_{,t} + (-r\mathbf{j} \cdot \mathbf{k})_{,r} = 0 \quad (64)$$

The existence conditions for conservation laws below were generated with the program CONLAW2 described in [11]. It generates conditions for 6 integrating factors  $Q_1, \dots, Q_6$  (like the multipliers  $rk_1, rk_2, rk_3, j_1, j_2, j_3$  on the left hand side of (64)). Each of the  $Q_i$  is an unknown function of 20 independent variables  $t, r, \mathbf{j}, \mathbf{k}, \mathbf{l} (= \mathbf{j}_{,r}), \mathbf{m} (= \mathbf{k}_{,r}), \mathbf{u} (= \mathbf{j}_{,rr}), \mathbf{w} (= \mathbf{k}_{,rr})$ . The system consists of 18 conditions of the form

$$0 = Q_{1,u_1} - Q_{4,w_1} r$$

and 6 conditions of the form

$$0 = Q_{3,j_1} l_1 r^2 + Q_{3,l_1} u_1 r^2 + Q_{3,j_2} l_2 r^2 + Q_{3,l_2} u_2 r^2 + Q_{3,j_3} l_3 r^2 + Q_{3,l_3} u_3 r^2 + Q_{3,k_1} m_1 r^2 + Q_{3,m_1} w_1 r^2 \\ + Q_{3,k_2} m_2 r^2 + Q_{3,m_2} w_2 r^2 + Q_{3,k_3} m_3 r^2 + Q_{3,m_3} w_3 r^2 + Q_{3,r} r^2 - Q_{6,j_1} k_1 r^2 - Q_{6,j_1} m_1 r^3 + Q_{6,l_1} k_1 r$$

$$\begin{aligned}
& -Q_{6,l_1} m_1 r^2 - Q_{6,l_1} w_1 r^3 - 2Q_{6,u_1} k_1 + 2Q_{6,u_1} m_1 r - Q_{6,u_1} w_1 r^2 - Q_{6,j_2} k_2 r^2 - Q_{6,j_2} m_2 r^3 + Q_{6,l_2} k_2 r \\
& -Q_{6,l_2} m_2 r^2 - Q_{6,l_2} w_2 r^3 - 2Q_{6,u_2} k_2 + 2Q_{6,u_2} m_2 r - Q_{6,u_2} w_2 r^2 - Q_{6,j_3} k_3 r^2 - Q_{6,j_3} m_3 r^3 + Q_{6,l_3} k_3 r \\
& -Q_{6,l_3} m_3 r^2 - Q_{6,l_3} w_3 r^3 - 2Q_{6,u_3} k_3 + 2Q_{6,u_3} m_3 r - Q_{6,u_3} w_3 r^2 - Q_{6,k_1} l_1 r^3 - Q_{6,k_1} j_2 k_3 r^3 + Q_{6,k_1} j_3 k_2 r^3 \\
& -Q_{6,m_1} u_1 r^3 - Q_{6,m_1} j_2 m_3 r^3 - Q_{6,m_1} l_2 k_3 r^3 + Q_{6,m_1} j_3 m_2 r^3 + Q_{6,m_1} l_3 k_2 r^3 - Q_{6,w_1} j_2 w_3 r^3 - 2Q_{6,w_1} l_2 m_3 r^3 \\
& -Q_{6,w_1} u_2 k_3 r^3 + Q_{6,w_1} j_3 w_2 r^3 + 2Q_{6,w_1} l_3 m_2 r^3 + Q_{6,w_1} u_3 k_2 r^3 + Q_{6,k_2} j_1 k_3 r^3 - Q_{6,k_2} l_2 r^3 - Q_{6,k_2} j_3 k_1 r^3 \\
& + Q_{6,m_2} j_1 m_3 r^3 + Q_{6,m_2} l_1 k_3 r^3 - Q_{6,m_2} u_2 r^3 - Q_{6,m_2} j_3 m_1 r^3 - Q_{6,m_2} l_3 k_1 r^3 + Q_{6,w_2} j_1 w_3 r^3 + 2Q_{6,w_2} l_1 m_3 r^3 \\
& + Q_{6,w_2} u_1 k_3 r^3 - Q_{6,w_2} j_3 w_1 r^3 - 2Q_{6,w_2} l_3 m_1 r^3 - Q_{6,w_2} u_3 k_1 r^3 - Q_{6,k_3} j_1 k_2 r^3 + Q_{6,k_3} j_2 k_1 r^3 - Q_{6,k_3} l_3 r^3 \\
& - Q_{6,m_3} j_1 m_2 r^3 - Q_{6,m_3} l_1 k_2 r^3 + Q_{6,m_3} j_2 m_1 r^3 + Q_{6,m_3} l_2 k_1 r^3 - Q_{6,m_3} u_3 r^3 - Q_{6,w_3} j_1 w_2 r^3 - 2Q_{6,w_3} l_1 m_2 r^3 \\
& - Q_{6,w_3} u_1 k_2 r^3 + Q_{6,w_3} j_2 w_1 r^3 + 2Q_{6,w_3} l_2 m_1 r^3 + Q_{6,w_3} u_2 k_1 r^3 - Q_{6,t} r^3 - k_1 Q_2 r^2 + k_2 Q_1 r^2
\end{aligned}$$

After introducing new unknown functions  $x_i, y_i$  through  $u_i = x_i + y_i, w_i = x_i - y_i$  the 18 short equations took the form of a total derivative and each one could be integrated on its own but when the computed functions were substituted only indirectly separable equations (ISEs) like (24) were obtained.<sup>1</sup>

Despite of the initial success in performing these integrations all attempts to complete the solution of the over-determined system failed with the 1999 version of CRACK. That this was not simply a matter of lacking computing power became obvious after extracting a small sub-system of equations for only one of the unknown functions<sup>2</sup>  $c_4(t, r, x_1, x_2, x_3, y_1, y_2, y_3)$  where some of the equations are easy to integrate:

$$\begin{aligned}
0 &= c_{4,x_3x_3y_2y_3} = c_{4,x_1x_2y_1y_3y_3} = c_{4,x_1x_2y_1y_1y_3} = c_{4,x_1x_2x_3y_1y_1} = c_{4,x_2x_3x_3x_3y_1y_1y_3} \\
&= c_{4,x_1x_2x_3x_3x_3y_1y_2y_2} = c_{4,x_1x_2x_3x_3y_3y_3y_3} - c_{4,x_2x_3x_3x_3y_1y_3y_3} \\
&= c_{4,x_1x_2x_3x_3y_1y_2y_2} - 2c_{4,x_1x_2x_2x_3y_1y_2y_3} = c_{4,x_1x_2x_3x_3y_1y_2} - 2c_{4,x_1x_2x_2x_3y_1y_3} - c_{4,x_1x_2x_2x_3x_3y_1y_3} x_3 \\
&= c_{4,x_1x_2x_3x_3x_3y_1y_3} x_1 - c_{4,x_1x_2x_3x_3y_3y_3} + c_{4,x_2x_3x_3x_3y_1y_3} \\
&= c_{4,x_3x_3x_3y_1y_3} x_1 + c_{4,x_3x_3y_1y_3y_3} y_1 - c_{4,x_3x_3y_3y_3} \tag{65} \\
&= c_{4,x_1x_2x_3x_3y_1y_2y_2y_2} y_3 + 2c_{4,x_1x_2x_2x_2y_1y_2y_3} - 2c_{4,x_1x_2x_2x_3y_1y_2y_2} + c_{4,x_1x_2x_2x_3x_3y_1y_2y_2} x_3 \\
&= c_{4,x_1x_2x_3x_3y_1y_2y_2} y_3 + 2c_{4,x_1x_2x_2x_2x_3y_1y_3} x_3 + 2c_{4,x_1x_2x_2x_2y_1y_3} - 2c_{4,x_1x_2x_2x_3y_1y_2} \\
&= c_{4,x_1x_2x_3x_3x_3y_1y_2} x_1x_3 - 3c_{4,x_1x_2x_3x_3y_1y_2} x_1 + 6c_{4,x_1x_2x_2x_3y_1y_3} x_1 \\
&\quad - c_{4,x_1x_2x_2x_3x_3y_3y_3} x_3^2 + c_{4,x_2x_2x_3x_3x_3y_1y_3} x_3^2
\end{aligned}$$

Even the solution or at least simplification of this sub-system was not possible. The problem was not to find equations with the form of a total derivative and to integrate them. The problem was the growing number of new functions of integration (which did still depend on 7 variables) and the appearance of too many only indirectly separable equations (ISEs).

Since 1999 the module for handling ISEs has been improved considerably. The current version of CRACK (Dec. 2001) can simplify the above system quickly using the conventional integration of total derivatives. Nevertheless, by adding the ability of performing syzygy based integrations the computation speeds up further and the solution involves fewer redundant arbitrary functions. Tests described below show that syzygy based integrations are well suited to be performed along the computation of a differential Gröbner basis without negative side effects. By that Gröbner basis computations can be cut short and the risk of a memory explosion be lowered.

## 10.2 A comparison of three computer runs

Before describing the details of 3 different computer runs, a few comments about the setup have to be made. The package CRACK for solving and simplifying over-determined PDE-systems contains

<sup>1</sup>Although each of the ISEs is over-determined on its own, this over-determination can not be utilized easily because there is no independent variable which occurs only explicitly that would lead to direct separations.

<sup>2</sup>New constants and functions of integration are all called  $c_i$  in CRACK with successively increasing subscript.

about 30 modules for different actions to be taken either with individual equations or with groups of equations of the system. Modules used to solve systems like (65) are

1. Direct separation of an equation with respect to some variable that occurs only explicitly in the equation.
2. Substitution of a function  $f$  either by zero or by at most 2 terms and only if other functions occurring in these 2 terms depend on fewer variables than  $f$ .
3. Integration of an equation if it consists of a single derivative with respect to only one variable.
4. Elimination of a function  $f$  from any equation if  $f$  occurs only algebraically and linearly and if  $f$  depends on all variables occurring in this equation. Substitution of  $f$  in all other equations.
5. Deleting of any redundant equations as described on the bottom of the overview in section 5.
6. Integration based on a syzygy in conservation law form.
7. Conventional integration of a PDE but only if sufficiently many integrations are possible such that the integrated equation can be used for a substitution.
8. Indirect separation of an equation (ISE). (This is a complex step which can invoke other direct separations and indirect separations of resulting equations.)
9. Reduction of the leading derivative of one equation with the help of another equation or formulation of an integrability condition between two equations. (This is a typical step in a Gröbner basis computation.)
10. Any integration of any equation even if not complete.

These modules are called in a specific sequence which can be chosen by specifying a list of numbers, each number representing one module. For example, if in table 1, column 2 the priority list of run 1 is chosen to be 1 2 3 4 8 9 7 10 then the modules as numbered above are tried in this order until one module is successful and then they are again tried beginning with 1 and so on. This is only a simplified description of the operation of CRACK but it is sufficient for the purpose of this section.

run	priority list of actions	# of steps	time in sec	# of terms in equ. (67)	# of redundant functions			
					of 6 var.	of 5 var.	of 4 var.	of 3 var.
1	1 2 3 4 8 9 7 10	1077	124	6	7	16	2	0
2	1 2 3 4 7 8 9 10	1175	122	12	4	45	23	5
3	1 2 3 4 5 8 6 9 7 10	362	23	8	2	19	2	0

Table 1. A comparison of three different runs on the system (65).

In table 1 three computer runs are compared. Column 3 gives the number of successful calls of the modules in the priority lists. Times shown in column 4 have been measured in a session of the computer algebra system REDUCE version 3.7 with 120 MByte memory (although only a few MByte are needed for this computation) on a 1.7 GHz PC Pentium 4 under Linux. Column 5 gives the number of terms in the single unsolved equation which in the solution (66) below is the equation (67). In the remaining 4 columns the number of redundant functions of 6, 5, 4, or 3 variables is shown. For example, if two functions  $f(x, y, z)$  and  $zg(x)$  occur always together such that a substitution  $\rightarrow f + zg$  has the same effect as  $g \rightarrow 0$  then  $g$  can be set to zero without loss of generality.

### 10.3 Conclusions

The central issue in these runs is, whether integrations (modules 6 and 7) are given a higher priority than the formulation of integrability conditions (module 10) or a lower priority. If integrability conditions have a higher priority than integrations, as in run 1, then at first a complete differential Gröbner basis is computed before integrations start. The benefit is that the differential order of equations is as low as possible when integrations start (assuming a total degree ordering is used in the differential Gröbner basis computation). Consequently fewer integrations are necessary and fewer functions will be generated which turn out later to be redundant. The disadvantage is that the computation of integrability conditions may take very long and blow up the systems size, or may even be impossible.

One can attempt to give integrations a higher priority at the price of more redundant functions in the solution. This was done in run 2. The benefit may be considerable, only in our small system (65) the Gröbner basis computation is not expensive at all, so the advantage of early integrations does not become obvious here. But the disadvantage becomes obvious. Integrating higher order equations generates more new functions with many of them turning out to be redundant at the end.

Finally, in the third run we get the best of both previous runs. Here, early integrations use syzygies in conservation law form as soon as they become available. The lowered differential order of equations reduces the complexity of the remaining Gröbner basis computation. Also, because with each integration at least 2 equations  $0 = P^i$  are satisfied, the number of new functions of integration is low and the number of variables these functions depend on is reduced. Consequently, only few functions turn out to be redundant in the computed solution as seen in columns 6-9 of table 1.

The following solution is obtained in run 3 after redundant functions have been deleted (by hand) leaving 11 functions of 6 variables, 8 functions of 5 variables and 2 functions of 4 variables. It is equivalent to the solutions returned in runs 1 and 2.

$$\begin{aligned}
c_4 = & c_{100,x_2} x_3 y_1 y_2 + \frac{1}{2} c_{100,x_3} x_1 y_3^2 + c_{125,x_2} x_2^2 y_1 + c_{125,y_2} x_3 y_1 y_3 + c_{133,x_2} x_3 y_1 + c_{133,y_2} y_1 y_3 \\
& + c_{213,x_3} x_1 + c_{213,y_3} y_1 + c_{100} y_1 y_3 - c_{109} x_2 x_3 y_1 + c_{170} + c_{172} + c_{173} x_3 + c_{181} y_3 + c_{191} \\
& - c_{192} - c_{193} x_3 - c_{194} + c_{200} + c_{205} - \frac{1}{2} c_{65} x_3^2 y_3 - c_{81} x_3 - c_{83}
\end{aligned} \tag{66}$$

All functions depend on  $t, r$  and in addition on further variables in the following way:

$$\begin{aligned}
& c_{83}(x_2, x_3, y_1, y_2), \quad c_{81}(x_2, y_1, y_2, y_3), \quad c_{173}(x_1, x_2, y_2, y_3), \quad c_{172}(x_1, x_2, y_2, y_3), \quad c_{170}(x_1, x_2, x_3, y_2), \\
& c_{194}(x_1, x_3, y_1, y_3), \quad c_{193}(x_1, y_1, y_2, y_3), \quad c_{192}(x_1, y_1, y_2, y_3), \quad c_{191}(x_1, x_3, y_1, y_2), \quad c_{205}(x_2, y_1, y_2, y_3), \\
& c_{200}(x_1, x_2, y_1, y_2), \quad c_{100}(x_1, x_2, x_3), \quad c_{125}(x_1, x_2, y_2), \quad c_{133}(x_1, x_2, y_2), \quad c_{181}(x_1, x_2, x_3), \\
& c_{213}(x_2, x_3, y_3), \quad c_{230}(x_1, y_1, y_3), \quad c_{229}(x_1, y_1, y_3), \quad c_{228}(x_1, x_3, y_1), \quad c_{65}(x_2, y_1), \quad c_{109}(x_1, y_2).
\end{aligned}$$

The function  $c_{194}$  has to satisfy the condition

$$0 = c_{194,x_3 y_1} x_1 + c_{194,y_1 y_3} y_1 - c_{194,y_3} - c_{228} - c_{229} - c_{230} x_3, \tag{67}$$

all other functions are free. The result of the conservation law investigation for the SU(2) chiral equation in the form (62), (63) is that no other conservation laws with integrating factors of at most 2<sup>nd</sup> order exist.

The following final remarks are of more general nature.

- Syzygy based integration can not replace conventional integration. If equations become decoupled then no integrability conditions apply and the equations have to be integrated conventionally if possible.



- The usefulness of conventional integration relies very much on the efficiency of a module for the indirect separation (module 8 in the above list). The corresponding implementation in CRACK will be described elsewhere.
- The issue of avoiding redundant functions is serious when a system like (65) is only a subsystem of a larger system and the solution of the smaller system is to be substituted in the larger one. Redundant functions would complicate the solution of the larger system unnecessarily. On the other hand, the identification and deletion of redundant functions using a method described in [14], is difficult and may be more expensive than the solution/simplification of the system itself.

The package CRACK is distributed together with the computer algebra system REDUCE. A newer version can be down-loaded from <http://lie.math.brocku.ca/twolf/crack>.

## Appendix A: Under-determination of syzygies

If one interprets syzygies as PDEs for unknowns  $e_k$ , then the original equations  $e_k = e_k(x^i, f_J^\alpha)$  are special solutions of these syzygies where the  $f^\alpha$  play the role of arbitrary functions in these solutions. Because at least one of the  $f^\alpha$  depends on all variables  $x^i$  (otherwise the original system consists only of ISEs to be treated differently, not by checking integrability conditions), the syzygies must be an under-determined PDE-system for the unknowns  $e_k$ . Computing conservation laws for under-determined systems of PDEs is an even more over-determined problem. The conservation law conditions have to be satisfied identically in a jet space with coordinates  $x^n, e_a$  and all partial derivatives of all  $e_a$ . The more  $e_a$  occur in the syzygies the more restrictive are their conservation law conditions. Another way to see this is that conditions for integrating factors to give conservation laws are obtained by applying the variational derivative (Euler-Lagrange operator) to the product of integrating factors and syzygies (see [7]). Because there is one Euler operator for each  $e_a$  we get as many conditions as there are different  $e_a$ . Finally, the more over-determined a system of conditions is, the easier it is to solve. Therefore the subtask of computing conservation laws of systems of syzygies is usually not a problem.

## Appendix B: The algorithm DIVINT

The following algorithm computes expressions  $Q^{ij}(x^n, f_j^\alpha) = Q^{[ij]}$  that satisfy  $D_j Q^{ij} = P^i$ . The given  $P^i = P^i(x^n, f_j^\alpha)$  are assumed to satisfy  $D_i P^i = 0$  identically in all  $f_j^\alpha$ .

```

1  Algorithm DIVINT
2  Input   variables:  $x^n$ , functions:  $f^\alpha$  and conserved current:  $P^i = P^i(x^n, f_j^\alpha)$ 
3  Output  $Q^{ij}(x^n, f_j^\alpha)$ ,  $j > i$    % satisfying  $D_j Q^{ij} = P^i$ ,
4           $E, F$                                %  $E$  : list of new additional equations
5          %  $F$  : list of new additional functions
6  Body                                     % no summation over double indices below
7     $E := \{\}$ ,  $F := \{\}$ ,  $Q^{ij} := 0$ ,   with  $i, j \in 1, \dots, p$ ,  $j > i$ 
8
9    % Integrate all terms with functions  $f^\alpha$  depending on all variables
10   for  $i := 1$  to  $(p - 1)$  do
11     for  $j := i + 1$  to  $p$  do
12       while  $P^i$  contains a term  $a^{iJ} \partial_j f_j^\alpha$  do           % i.e. while any derivative of any  $f^\alpha$ 
13                                     % occurs that involves  $\partial_j$ 
14          $P^i \rightarrow P^i - D_j(a^{iJ} f_j^\alpha)$ 
15          $P^j \rightarrow P^j + D_i(a^{iJ} f_j^\alpha)$ 
16          $Q^{ij} \rightarrow Q^{ij} + a^{iJ} f_j^\alpha$ 
17
18     % Integrate all derivatives involving functions  $f^\alpha$  not depending on all variables
19     for  $i := 2$  to  $p$  do
20       for  $j := 1$  to  $i - 1$  do
21         while  $P^i$  contains a term  $a^{iJ} \partial_j f_j^\alpha$  do           % i.e. while any derivative of any  $f^\alpha$ 
22                                     % occurs that involves  $\partial_j$ 
23          $P^i \rightarrow P^i - D_j(a^{iJ} f_j^\alpha)$ 
24          $Q^{ji} \rightarrow Q^{ji} - a^{iJ} f_j^\alpha$ 
25
26     % Integrate remaining terms
27     for  $i := 1$  to  $p$  do
28       if  $P^i \neq 0$  then
29         % integrate each term  $a^{iJ} f_j^\alpha$  of  $P^i$  with respect to any one  $x^j \neq x^i$ 
30         % preferably one  $x^j$  with  $\partial_j f^\alpha = 0$  in the following way:
31         if  $\partial_j f^\alpha = 0$  then
32            $q := f_j^\alpha \int a^{iJ} dx^j$ 
33            $P^i \rightarrow P^i - D_j q$ 
34           if  $j > i$  then  $Q^{ij} \rightarrow Q^{ij} + q$ 
35           else  $Q^{ji} \rightarrow Q^{ji} - q$ 
36         else
37           Introduce a new function  $f^\beta(x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^p)$ 
38            $F \rightarrow F \cup \{f^\beta\}$ 
39            $E \rightarrow E \cup \{0 = \partial_j f^\beta - a^{iJ} f_j^\alpha\}$ 
40            $P^i \rightarrow P^i - a^{iJ} f_j^\alpha$ 
41           if  $j > i$  then  $Q^{ij} \rightarrow Q^{ij} + f^\beta$ 
42           else  $Q^{ji} \rightarrow Q^{ji} - f^\beta$ 
43   return  $Q^{ij}(x^n, f_j^\alpha)$ ,  $E$  (list of new equations),  $F$  (list of new functions)

```

## Explanation of the algorithm

*Lines 9 - 16*

This part of the procedure is sufficient if the input expressions  $P^i(x^n, f_J^\alpha)$  do only contain functions  $f^\alpha$  depending on all  $p$  independent variables  $x^1, \dots, x^p$ .

A typical example: If an expression  $P^y$  contains a term  $f_{,z}$  then  $D_y P^y$  (no summation) contains  $\partial_y f_{,z}$  which has to be cancelled by  $-\partial_z f_{,y}$  from  $D_z P^z$  (no summation) to give  $0 = D_k P^k$  (summation) identically in all  $f_J$ . This means  $P^z$  contains  $-f_{,y}$ . In this short example the lines 14 - 16 would subtract  $f_{,z}$  from  $P^y$ , subtract  $-f_{,y}$  from  $P^z$  and add  $f$  to  $Q^{yz}$ . There is no principal difference between  $P^y$  containing a term  $f_{,z}$  or  $P^y$  containing  $a^{ij} \partial_z f^\alpha$ .

As both,  $P^i$  and  $P^j$  are updated in lines 14 and 15,  $j$  does not run over indices  $1 \dots i-1$ . Because  $Q^{ii} = 0$  ( $Q^{ij}$  is antisymmetric) there is no need to integrate an  $i$ -derivative in  $P^i$  and therefore  $j$  starts from  $i+1$  in line 11.

If all terms in all  $P^i$  contain a function  $f^\alpha$  of all variables then any term in any  $Q^{ij}$  occurs twice, once with an  $x^j$ -derivative in  $P^i$  and once as negative  $x^i$ -derivative in  $P^j$ . When the program completed lines 10 - 16, all  $P^i$  have the value zero and the solution  $Q^{ij}$  is found (for  $i < j$ , values for  $Q^{ji}$  follow from the antisymmetry).

*Lines 18 - 42*

The only possibility that after completing lines 10 - 16 not all  $P^i$  are already zero occurs if some  $f^\alpha$  do not depend on all variables. That is, for example, the case if functions entered the problem due to running DIVINT previously in earlier integrations. In general, if terms remain in some  $P^i$  which necessarily depend on less than all variables then one can always complete the integrations by introducing new functions (collected in a list  $F$  in line 38) that have to satisfy additional equations (collected in a list  $E$  in line 39). In order to minimize the number of additional functions and additional equations the lines 19 - 24 integrate terms that are  $x^j$ -derivatives in  $P^i$  ( $j \neq i$ ) and lines 31 - 35 integrate terms by changing the explicit appearance of  $x^j$ . This is shown in the following examples.

*Example:* Independent variables:  $x, y, z$ , initial values:

$$\begin{aligned} P^x &= A(y, z)_{,y} + B(y, z)_{,z} + C(y, z) + D(y) + G(z) \\ P^y &= H(x, z)_{,x} + K(x, z)_{,z} + L(x) + M(x, z) + N(z) \\ P^z &= R(x, y)_{,x} + S(x, y)_{,y} + T(x) + U(y) + W(x, y) \\ Q^{xy} &= Q^{xz} = Q^{yz} = 0 \end{aligned}$$

containing undetermined functions  $A, B, C, D, G, H, K, L, M, N, R, S, T, U$  and  $W$ . After completing the program up to line 18 the values are

$$\begin{aligned} P^x &= C(y, z) + D(y) + G(z) \\ P^y &= H(x, z)_{,x} + L(x) + M(x, z) + N(z) \\ P^z &= R(x, y)_{,x} + S(x, y)_{,y} + T(x) + U(y) + W(x, y) \\ Q^{xy} &= A(y, z) \\ Q^{xz} &= B(y, z) \\ Q^{yz} &= K(x, z). \end{aligned}$$

After completing the program up to line 26 the values are

$$\begin{aligned} P^x &= C(y, z) + D(y) + G(z) \\ P^y &= L(x) + M(x, z) + N(z) \\ P^z &= T(x) + U(y) + W(x, y) \end{aligned}$$

$$\begin{aligned}
Q^{xy} &= A(y, z) - H(x, z) \\
Q^{xz} &= B(y, z) - R(x, y) \\
Q^{yz} &= K(x, z) - S(x, y).
\end{aligned}$$

The loop beginning in line 27 will integrate the remaining terms in  $P^i$ . The lines 32 - 35 will integrate the terms  $D, G, L, N, T, U$  and lines 37 - 42 the terms  $C, M, W$  to obtain

$$\begin{aligned}
P^x &= P^y = P^z = 0 \\
Q^{xy} &= A(y, z) - H(x, z) + yG(z) - xN(z) + F^1(y, z) \\
Q^{xz} &= B(y, z) - R(x, y) + zD(y) - xU(y) - F^3(x, y) \\
Q^{yz} &= K(x, z) - S(x, y) + zL(x) - yT(x) + F^2(x, z)
\end{aligned}$$

with a list  $F$  of new additional functions  $F^1(y, z), F^2(x, z), F^3(x, y)$  and list  $E$  of new additional equations

$$\begin{aligned}
F^1(y, z)_{,y} &= C(y, z) \\
F^2(x, z)_{,z} &= M(x, z) \\
F^3(x, y)_{,x} &= W(x, y)
\end{aligned}$$

each in less than 3 variables.

## References

- [1] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Computing representations for radicals of finitely generated differential ideals. In *Proceedings of ISSAC 95*, pages 158–166. ACM Press, 1995.
- [2] E. Hubert. Essential components of algebraic differential equations. *J. of Symb. Comp.*, 28(4-5):657–680, 1999.
- [3] E. Hubert. Factorisation free decomposition algorithms in differential algebra. *J. of Symb. Comp.*, 29(4-5), 2000.
- [4] E. Landau. Über irreduzible Differentialgleichungen. *J. für die reine und angewandte Mathematik*, 124:115–120, 1902.
- [5] A. Loewy. Über reduzible lineare homogene Differentialgleichungen. *Math. Annalen*, 56:549–584, 1903.
- [6] E.L. Mansfield. The differential algebra package diffgrob2. *Mapletech*, 3:33–37, 1996.
- [7] P. J. Olver. *Applications of Lie Groups to Differential Equations*, volume 107 of *gtm*. Springer Verlag, New York-Berlin-Heidelberg-Tokyo, 1986.
- [8] G.J. Reid, A.D. Wittkopf, and A. Boulton. Reduction of systems of nonlinear partial differential equations to simplified involutive forms. *Europ. J. of Appl. Math.*, 7:604–635, 1996.
- [9] S.P. Tsarev. An algorithm for complete enumeration of all factorizations of a linear ordinary differential operator. In *Proceedings of ISSAC 96*, pages 226–231. ACM Press, 1996.
- [10] T. Wolf. *The program CRACK for solving PDEs in General Relativity*, pages 241–251. Springer Verlag, 1996.

- [11] T. Wolf. A comparison of four approaches to the calculation of conservation laws. preprint, to appear in *Euro. J. of Appl. Math.*, 1998.
- [12] T. Wolf. A linearization of PDEs based on conservation laws. preprint, 1999.
- [13] T. Wolf. The symbolic integration of exact PDEs. *J. Symb. Comp.*, 30(5):619–629, 2000.
- [14] T. Wolf, A. Brand, and M. Mohammadzadeh. Computer algebra algorithms and routines for the computation of conservation laws and fixing of gauge in differential expressions. *J. Symb. Comp.*, 27:221–238, 1999.