Symbolic computation of conservation laws for nonlinear partial differential equations in multiple space dimensions

Douglas Poole and Willy Hereman

Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA

Abstract
A method for symbolically computing conservation laws of nonlinear partial differential equations (PDEs) in multiple space dimensions is presented in the language of variational calculus and linear algebra. The steps of the method are illustrated using the Zakharov-Kuznetsov and Kadomtsev-Petviashvili equations as examples.

The method is algorithmic and has been implemented in Mathematica. The software package, CONSERVATIONLAWSDM.m, can be used to symbolically compute and test conservation laws for polynomial PDEs that can be written as nonlinear evolution equations.

The code CONSERVATIONLAWSDM.m has been applied to (2+1)-dimensional versions of the Sawada-Kotera, Camassa-Holm, and Gardner equations, and the multi-dimensional Khokhlov-Zabolotskaya equation.

Keywords: Conservation laws; Nonlinear PDEs; Symbolic software; Complete integrability

1. Introduction
Many nonlinear partial differential equations (PDEs) in the applied sciences and engineering are continuity equations which express conservation of mass, momentum, energy, or electric charge. Such equations occur in, e.g., fluid mechanics, particle and quantum physics, plasma physics, elasticity, gas dynamics, electromagnetism, magneto-hydro-dynamics, nonlinear optics, etc. Certain nonlinear PDEs admit infinitely many conservation laws. Although most lack a physical interpretation, these conservation laws play an important role in establishing the complete integrability of the PDE. Completely integrable PDEs are nonlinear PDEs that can be linearized by some transformation or explicitly solved with the Inverse Scattering Transform (IST). See, e.g., Ablowitz and Clarkson (1991).

The search for conservation laws of the Korteweg-de Vries (KdV) equation began around 1964 and the knowledge of conservation laws was paramount for the development of soliton
theory. As Newell (1983) narrates, it led to the discovery of the Miura transformation (which connects solutions of the KdV and modified KdV equations) and the Lax pair for the KdV equation. A Lax pair (Lax, 1968) is a set of two linear equations. The compatibility conditions of these equations is the nonlinear PDE. The Lax pair is the starting point for the IST (Ablowitz and Clarkson, 1991; Ablowitz and Segur, 1981), which has been used to construct soliton solutions, i.e., stable solutions that interact elastically upon collision.

Conversely, the existence of many (independent) conserved densities is a predictor for complete integrability. The knowledge of conservation laws also aids the study of qualitative properties of PDEs, in particular, bi-Hamiltonian structures and recursion operators (Baldwin and Hereman, 2010). Furthermore, if constitutive properties have been added to “close a model,” one should verify that conserved quantities have remained intact. Another application involves numerical solvers for PDEs (Sanz-Serna, 1982), where one checks if the first few (discretized) conserved densities are preserved after each time step.

There are several methods for computing conservation laws as discussed by Bluman *et al.* (2010), Hereman *et al.* (2005), Naz (2008), and Naz *et al.* (2008). One could apply Noether’s theorem, which states that a (variational) symmetry of the PDE corresponds to a conservation law. Using Noether’s method, the differential geometry package in *Maple* contains tools for conservation laws developed by Anderson (2004a) and Anderson and Cheb-Terrab (2009). Circumventing Noether’s Theorem, Wolf (2002) has developed four programs in REDUCE which solve an over-determined system of differential equations to get conservation laws. Based on the integrating factor method, Cheviakov (2007, 2010) has written a *Maple* program that computes a set of multipliers on the PDE. To find conservation laws, here again, one has to solve a system of differential equations. Last, conservation laws can be obtained from the Lax operators, as shown in, e.g., Drinfel’d and Sokolov (1985).

By contrast, the direct method discussed in this paper uses tools from calculus, the calculus of variations, linear algebra, and differential geometry. Briefly, our method works as follows. A candidate (local) density is assumed to be a linear combination with undetermined coefficients of monomials that are invariant under the scaling symmetry of the PDE. Next, the time derivative of the candidate density is computed and evaluated on the PDE. Subsequently, the variational derivative is applied to get a linear system for the undetermined coefficients. The solution of that system is substituted into the candidate density. Once the density is known, the flux is obtained by applying a homotopy operator to invert a divergence. Our method can be implemented in any major computer algebra system (CAS). The package *ConservationLawsMD.m* by Poole and Hereman (2009) is a *Mathematica* implementation based on Hereman *et al.* (2005), with new features added by Poole (2009).

This paper is organized as follows. Section 2 shows conservation laws for the Zakharov-Kuznetsov (ZK) and Kadomtsev-Petviashvili (KP) equations. Section 3 covers the tools that will be used in the algorithm. In Section 4, the algorithm is presented and illustrated for the ZK and KP equations. Section 6 discusses conservation laws of PDEs in multiple space dimensions, including the Sawada-Kotera, Khokhlov-Zabolotskaya, and Camassa-Holm equations. A general conservation law for the KP equation is given in Section 5. Using the (2+1)-dimensional Gardner equation as an example, Section 7 shows how to use *ConservationLawsMD.m*. Finally, some conclusions are drawn in Section 8.
2. Examples of Conservation Laws

This paper deals with systems of PDEs of order \( P \),

\[
\Delta(u^{(p)}(x, t)) = 0, \tag{1}
\]

in \((n + 1)\) dimensions where \( n \) refers to the number of space variables, i.e., the number of components of \( x \). The additional independent variable is time \((t)\). \( u^{(p)}(x, t) \) denotes the dependent variable \( u = (u^1, \ldots, u^j, \ldots, u^N) \) and its partial derivatives (up to order \( M \)) with respect to \( x \) and \( t \).

Although the method discussed in this paper works for arbitrary \( n \), the algorithm and code are restricted to 1 D, 2 D, and 3 D cases. Hence, the space variable \( x \) represents \( x, (x, y), \) or \((x, y, z)\), respectively. For simplicity, in the examples we will denote \( u^1, u^2, u^3, \) etc. Partial derivatives are denoted by subscripts, e.g., \( \partial^3_{y^3} \) is written as \( u_{3x2y} \).

A conservation law for (1) is a scalar PDE in the form

\[
\mathcal{D}_t \rho + \text{Div} \mathbf{J} = 0 \quad \text{on } \Delta = 0, \tag{2}
\]

where \( \rho = \rho(x, t, u^{(M)}(x, t)) \) is the conserved density of order \( M \), and \( \mathbf{J} = \mathbf{J}(x, t, u^{(K)}(x, t)) \) is the associated flux of order \( K \). Equation (2) is satisfied for all solutions of (1). In (2), \( \text{Div} \mathbf{J} = \mathcal{D}_x J^1 + \mathcal{D}_y J^2 \) if \( \mathbf{J} = (J^1, J^2) \) and \( \text{Div} \mathbf{J} = \mathcal{D}_x J^1 + \mathcal{D}_y J^2 + \mathcal{D}_z J^3 \) if \( \mathbf{J} = (J^1, J^2, J^3) \).

Logically, \( \mathcal{D}_t, \mathcal{D}_x, \mathcal{D}_y, \) and \( \mathcal{D}_z \) are total derivative operators and \( \text{Div} \) is the total divergence operator in 2D or 3D, respectively. For example, the total derivative operator \( \mathcal{D}_x \) (in 1D) acting on \( f = f(x, t, u^{(M)}(x, t)) \) of order \( M \) is defined as

\[
\mathcal{D}_x f = \frac{\partial f}{\partial x} + \sum_{j=1}^{N} \sum_{k=0}^{M_j} u_{(k+1)x}^j \frac{\partial f}{\partial u_{kx}^j}, \tag{3}
\]

where \( M_j \) is the order of \( f \) in component \( u^j \) and \( M = \max_{j=1, \ldots, N} M_j \). The partial derivative \( \frac{\partial}{\partial x} \) acts on any \( x \) that appears explicitly in \( f \), but not on \( u^j \) or any partial derivatives of \( u^j \). Total derivative operators in multiple dimensions are defined analogously (see Section 3).

The algorithm described in Section 4 allows one to compute local conservation laws for systems of PDEs that can be written as nonlinear evolution equations, say, in time \( t \),

\[
u_t = G(x, u^{(M)}(x)), \tag{4}
\]

where \( G \) is assumed to be smooth.

Not all multi-dimensional systems of PDEs are evolution equations of type (4). However, the \((3+1)\)-dimensional continuity equation (2), \( \mathcal{D}_t \rho + \mathcal{D}_x J^1 + \mathcal{D}_y J^2 + \mathcal{D}_z J^3 = 0 \), can be taken as, e.g., \( \mathcal{D}_x J^1 + \text{Div}(\rho, J^2, J^3) \), making it possible to first compute \( J^1 \), the \( x \)-component of the flux, then use \( J^1 \) to compute the rest of the conservation law. Thus, if a PDE can be transformed into an evolution equation in one of the space variables, the algorithm will still work. For efficiency, our algorithm does an internal interchange of variables so that the equations adhere to (4). However, the interchange is not used in this paper to keep the description of the algorithm clear.

We now introduce two well-documented PDEs together with some of their conservation laws. These PDEs will also be used in Section 4 to illustrate the steps of the algorithm.
Example 1. The Zakharov-Kuznetsov (ZK) equation is an evolution equation that characterizes three-dimensional ion-sound solitons in a low pressure uniform magnetized plasma (Zakharov and Kuznetsov, 1974). After re-scaling, it has the form

\[ u_t + \alpha uu_x + \beta \Delta u_x = 0, \]  

where \( u(x, y, z, t) \), \( \alpha \) and \( \beta \) are real parameters, and \( \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \) is the Laplacian in three dimensions. The conservation laws for the (2+1)-dimensional ZK equation,

\[ u_t + \alpha uu_x + \beta(u_{2x} + u_{2y})_x = 0, \]  

where \( u(x, y, t) \), were studied by, e.g., Zakharov and Kuznetsov (1974), Infeld (1985), and Shivamoggi et al. (1993). Summarizing their results,

\[ D_t(u) + D_x \left( \frac{\alpha}{2} u^2 + \beta u_{2x} \right) + D_y(\beta uy_x) = 0, \]  

\[ D_t \left( \frac{3}{2} u^2 - 3\beta u_x^2 + 2\beta u(u_{2x} + u_{2y}) \right) - D_y(2\beta u uy_y) = 0, \]  

\[ D_t \left( u^3 - \frac{3\beta}{\alpha} (u_x^2 + u_y^2) + D_x \left( \frac{3\alpha}{4} u^4 + 3\beta u^2 u_{2x} - 6\beta u(u_x^2 + u_y^2) + \frac{3\beta^2}{\alpha} (u_{2x} - u_{2y}) - \frac{6\beta^2}{\alpha} (u_x(u_{3x} + u_{2y}) + u_y(u_{2xy} + u_{3y})) \right) + D_y \left( 3\beta u^2 u_{xy} + \frac{6\beta^2}{\alpha} u_{xy}(u_{2x} + u_{2y}) \right) \right) = 0, \]  

\[ D_t \left( tu^2 - \frac{2}{\alpha} xu \right) + D_x \left( t(\frac{2\alpha}{3} u^3 - \beta(u_x^2 - u_y^2) + 2\beta u(u_{2x} + u_{2y})) - \frac{2}{\alpha} x(-u_x^2 + \beta u_{2x} + \frac{3\beta}{\alpha} u_x) \right) - D_y \left( 2\beta (u_x uy_y - \frac{1}{\alpha} x u_{xy}) \right) = 0. \]  

Example 2. The well-known (2+1)-dimensional Kadomtsev-Petviashvili (KP) equation,

\[ (u_t + \alpha uu_x + u_{3x})_x + \sigma^2 u_{2y} = 0, \]  

for \( u(x, y, t) \), describes shallow water waves with wavelengths much greater than their amplitude moving in the \( x \)-direction and subject to weak variations in the \( y \)-direction (Kadomtsev and Petviashvili, 1970). The parameter \( \alpha \) occurs after a re-scaling of the physical coefficients and \( \sigma^2 = \pm 1 \). The KP equation is not an evolution equation. However, it can be written as an evolution system in the space variable \( y \),

\[ u_y = v, \quad v_y = -\sigma^2(u_{tx} + \alpha u_x^2 + \alpha uu_{2x} + u_{3x}). \]  

Note that \( \frac{1}{\sigma^2} = \sigma^2 \), and thus \( \sigma^4 = 1 \). System (12) instead of (11) will be used in the algorithm in Section 4.

Equation (11) expresses conservation of momentum:

\[ D_t(u_x) + D_x(\alpha uu_x + u_{3x}) + D_y(\sigma^2 u_y) = 0. \]  

Other well-documented conservation laws (Wolf, 2002) are

\[ D_t \left( fu \right) + D_x \left( f(\frac{1}{2} \alpha u^2 + u_{2x}) + \left( \frac{4}{\alpha} \sigma^2 f' y^2 - f x \right)(u_t + \alpha uu_x + u_{3x}) \right) - D_y \left( f' - uy_y + u_y \left( \frac{1}{2} f' y^2 - \sigma^2 f x \right) \right) = 0, \]  

\[ D_t \left( f y u \right) + D_x \left( fy \left( \frac{1}{2} \alpha u^2 + u_{2x} \right) + \left( \frac{4}{\alpha} \sigma^2 f' y^3 - f x y \right)(u_t + \alpha uu_x - u_{3x}) \right) - D_y \left( fy \left( \frac{1}{2} f' y^2 - \sigma^2 f x \right) - u_y \left( \frac{4}{\alpha} f' y^3 - \sigma^2 f x y \right) \right) = 0, \]  

where \( f(x, y, t) \) is a function of \( x, y, t \).
where \( f = f(t) \) is an arbitrary function. The arbitrary coefficient \( f \) leads to an infinite family of conservation laws, each of the form (14) or (15). In Section 4 we will show how (7)-(10), (14) and (15) are computed with our algorithm.

3. Tools from the Calculus of Variations and Differential Geometry

Operations are carried out on differential functions, \( f(x, t, u^{(M)}(x)) \), on the jet space. Assuming that all partial derivatives of \( u \) with respect to \( t \) are eliminated from \( f \) (viz., \( t \) is a parameter), then

\[
\mathbf{u}^{(M)}(x) = (u^1, u_x^1, u_y^1, u_z^1, u_{2x}^1, u_{2y}^1, u_{2z}^1, \ldots, u_{M_1^N x M_2^N y M_3^N z}^N),
\]

where \( M_1^j, M_2^j, \) and \( M_3^j \) are the orders of component \( u^j \) with respect to \( x, y, \) and \( z \), respectively, and \( M \) is the maximum total order of all terms in the differential function.

Derivatives on the jet space are called total derivatives; an example was given in (3). By contrast, partial derivatives like \( \frac{\partial^{k_1+k_2+k_3}}{\partial x^{k_1} \partial y^{k_2} \partial z^{k_3}} \) are denoted by \( u_{k_1 x k_2 y k_3 z} \). Although variable coefficients are allowed, the differential functions should not contain terms that are functions of \( (x, t) \) only (i.e., without being the coefficient of some variable in the jet space).

Three operators for the calculus of variations and differential geometry will play a major role in the conservation law algorithm. They are: the total derivative, the Euler operator (also known as the variational derivative), and the homotopy operator. All three operators (which act on the jet space) can be defined algorithmically which allows for straightforward and efficient computations.

**Definition 1.** The total derivative operator \( \mathcal{D}_x \) (in 2D), acting on \( f(x, y, t, u^{(M)}(x, y)) \), is defined as

\[
\mathcal{D}_x f = \frac{\partial f}{\partial x} + \sum_{j=1}^{N} \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} u_{(k_1+1)x k_2 y}^j \frac{\partial f}{\partial u_{k_1 x k_2 y}^j},
\]

where \( M_1^j \) and \( M_2^j \) are the orders of \( f \) for component \( u^j \) with respect to \( x \) and \( y \), respectively. \( \mathcal{D}_t, \mathcal{D}_y, \mathcal{D}_z \) can be defined analogously.

If the total derivative operator were applied by hand to a differential function \( f(x, t, u^{(M)}(x)) \), one would use the product and chain rules to complete the computation. However, formulas like (3) and (17) are more suitable for symbolic computation.

The Euler operator, which plays a fundamental role in the calculus of variations (Olver, 1993), allows one to test if differential functions are exact. Testing exactness is a key step in the computation of conservation laws.

**Definition 2.** The Euler operator for the 1D case where \( f = f(x, u^{(M)}(x)) \) is defined as

\[
\mathcal{L}_{u^j(x)} f = \sum_{k=0}^{M_1^j} (-\mathcal{D}_x)^k \frac{\partial f}{\partial u_{kx}^j} = \frac{\partial f}{\partial u^j} - \mathcal{D}_x \frac{\partial f}{\partial u_x^j} + \mathcal{D}_x^2 \frac{\partial f}{\partial u_{2x}^j} - \mathcal{D}_x^3 \frac{\partial f}{\partial u_{3x}^j} + \cdots + (-\mathcal{D}_x)^{M_1^j} \frac{\partial f}{\partial u_{M_1^j x}^j},
\]

5
where $j = 1, \ldots, N$. The 2D and 3D Euler operators (variational derivatives) are defined analogously (Olver, 1993). For example, in the 2D case where $x = (x, y)$ and $f = f(x, u^{(M)}(x))$,

$$\mathcal{L}_{u^j(x,y)}f = \sum_{k_1=0}^{M_1} \sum_{k_2=0}^{M_2} (-D_x)^{k_1}(-D_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x k_2 y}}, \quad j = 1, \ldots, N. \tag{19}$$

We suppressed the parameter $t$ in (17)-(19). As we will see later on, in some applications it will be necessary to swap the meaning of $t$ and $y$ (so that the latter becomes the parameter).

Obviously, application of the 2D Euler operator to $f = f(t, x, u^{(M)}(t, x))$ instead of $f = f(x, y, u^{(M)}(x, y))$ would then be accomplished by using $\mathcal{L}_{u^j(t,x)}$, defined analogously to (19). This remark applies to all operators used in this paper. The following theorem relates to an application of the Euler operator.

**Definition 3.** Let $f = f(x, u^{(M)}(x))$ be a differential function of order $M$. When $x = x$, $f$ is called exact if there exists a differential function $F(x, u^{(M-1)}(x))$ such that $f = D_x F$. When $x = (x, y)$ or $x = (x, y, z)$, $f$ is exact if there exists a differential vector function $F(x, u^{(M-1)}(x))$ such that $f = \text{Div} F$.

**Theorem 1.** A differential function $f = f(x, u^{(M)}(x))$ is exact if and only if $\mathcal{L}_{u^j(x)} f \equiv 0$. Here, $0$ is the vector $(0, 0, \cdots, 0)$ which has $N$ components matching the number of components of $u$.

**Proof.** The proof for a general multi-dimensional case is given, e.g., Poole (2009).

The homotopy operator (Anderson, 2004b; Olver, 1993) integrates exact 1D differential functions, or inverts the total divergence of exact 2D or 3D differential functions. Integration routines of CAS have been unreliable when integrating exact differential expressions involving unspecified functions. Often the built-in integration by parts routines fail when arbitrary functions appear in the integrand. The 1D homotopy operator offers an attractive alternative to integration since it circumvents integration by parts altogether.

**Definition 4.** Let $x = x$ be the independent variable and $f = f(x, u^{(M)}(x))$ be an exact differential function, i.e., there exists a function $F$ such that $F = D_x^{-1} f$. Thus, $F$ is the integral of $f$. The 1D homotopy operator is defined as

$$\mathcal{H}_{u(x)} f = \int_0^1 \left( \sum_{j=1}^N I_{u^j(x)} f \right) \left[ \lambda u \right] \frac{d\lambda}{\lambda}, \tag{20}$$

where $u = (u^1, \ldots, u^j, \ldots, u^N)$. The integrand, $I_{u^j(x)} f$, is defined as

$$I_{u^j(x)} f = \sum_{k=1}^{M_j} \left( \sum_{i=0}^{k-1} u^j_{i x} (-D_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u^j_{k x}}, \tag{21}$$

where $M_j$ is the order of $f$ in dependent variable $u^j$ with respect to $x$. The notation $f[\lambda u]$ means that in $f$ one replaces $u$ by $\lambda u$, $u_x$ by $\lambda u_x$, and so on for all derivatives of $u$, where $\lambda$ is an auxiliary parameter.
Given an exact differential function, the 1D homotopy operator (20) replaces integration by parts (in x) with a sequence of differentiations followed by a standard integration with respect to λ. Indeed, the following theorem states one purpose of the homotopy operator.

**Theorem 2.** Let \( f = f(x, u^{(M)}(x)) \) be exact, i.e., \( D_x F = f \) for some differential function \( F(x, u^{(M-1)}(x)) \). Then, \( F = D_x^{-1} f = \mathcal{H}_{u(x)} f \).

**Proof.** A proof for the 1D case in the language of standard calculus is given in Poole and Hereman (2010). Olver (1993) gives a proof based on the variational complex.

The homotopy operator (20) has been a reliable tool for integrating exact polynomial differential expressions. For applications, see Cheviakov (2007, 2010); Deconinck and Nivala (2009); Hereman (2006); Hereman et al. (2007). However, the homotopy operator fails to integrate certain classes of exact rational expressions as discussed in Poole and Hereman (2010). We will not consider rational expressions in this paper. Yet, the homotopy integrator code in Poole and Hereman (2009) covers large classes of exact rational functions.

CAS cannot invert the divergences of exact 2D and 3D differential functions. The 2D and 3D homotopy operators are valuable tools for these tasks.

**Definition 5.** Let \( f(x, u^{(M)}(x)) \) be an exact differential function involving two independent variables \( x = (x, y) \). The 2D homotopy operator is a “vector” operator with two components,

\[
\left( \mathcal{H}^{(x)}_{u(x,y)} f, \mathcal{H}^{(y)}_{u(x,y)} f \right),
\]

where

\[
\mathcal{H}^{(x)}_{u(x,y)} f = \int_0^1 \left( \sum_{j=1}^N I^{(x)}_{u^{(j)}(x,y)} f \right) [\lambda u] \frac{d\lambda}{\lambda} \quad \text{and} \quad \mathcal{H}^{(y)}_{u(x,y)} f = \int_0^1 \left( \sum_{j=1}^N I^{(y)}_{u^{(j)}(x,y)} f \right) [\lambda u] \frac{d\lambda}{\lambda}.
\]

The \( x \)-integrand, \( I^{(x)}_{u^{(j)}(x,y)} f \), is given by

\[
I^{(x)}_{u^{(j)}(x,y)} f = \sum_{k_1=1}^{M_1} \sum_{k_2=0}^{M_2} \left( \sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} B^{(x)} u^{i_1}_x u^{i_2}_y (\partial f / \partial u^{i_1}_x) \partial f / \partial u^{i_2}_y \right),
\]

with combinatorial coefficient \( B^{(x)} = B(i_1, i_2, k_1, k_2) \) defined as

\[
B(i_1, i_2, k_1, k_2) = \binom{i_1+i_2}{i_1} \binom{k_1+k_2-i_1-i_2-1}{k_1-1+i_1-i_2} \binom{k_1-k_2-i_1}{k_1}. \tag{25}
\]

Similarly, the \( y \)-integrand, \( I^{(y)}_{u^{(j)}(x,y)} f \), is defined as

\[
I^{(y)}_{u^{(j)}(x,y)} f = \sum_{k_1=0}^{M_1} \sum_{k_2=1}^{M_2} \left( \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2} B^{(y)} u^{i_1}_x u^{i_2}_y (\partial f / \partial u^{i_1}_x) \partial f / \partial u^{i_2}_y \right),
\]

where \( B^{(y)} = B(i_2, i_1, k_2, k_1) \).
Definition 6. Let \( f(x, u^{(M)}(x)) \) be a differential function of three independent variables \( x = (x, y, z) \). The homotopy operator in 3D is a three-component vector operator,

\[
(\mathcal{H}_{u^{(x)}(x,y,z)} f, \mathcal{H}_{u^{(y)}(x,y,z)} f, \mathcal{H}_{u^{(z)}(x,y,z)} f),
\]

where the \( x \)-component is given by

\[
\mathcal{H}_{u^{(x)}(x,y,z)} f = \int_0^1 \left( \sum_{j=1}^N I_{u^{(x)}(x,y,z)} f \right) \left[ \lambda u \right] \frac{d\lambda}{\lambda}.
\]

The \( y \)- and \( z \)-components are defined analogously. The \( x \)-integrand is given by

\[
f^{(x)}_{u^{(x)}(x,y,z)} f = \sum_{k_1=1}^{M_1} \sum_{k_2=0}^{M_2} \sum_{k_3=0}^{M_3} \sum_{k_{1-1}=0}^{k_1-1} \sum_{k_{2-2}=0}^{k_2-2} \sum_{k_{3-3}=0}^{k_3-3} (B^{(x)} u^{i_1}_{i_2 i_3}) \frac{\partial f}{\partial u^{i_1}_{i_2 i_3}},
\]

with combinatorial coefficient \( B^{(x)} = B(i_1, i_2, i_3, k_1, k_2, k_3) \) defined as

\[
B(i_1, i_2, i_3, k_1, k_2, k_3) = \frac{(i_1+i_2+i_3)}{i_1} \frac{(i_2+i_3)}{i_2} \frac{(k_1+k_2+k_3-i_1-i_2-i_3)}{k_1-i_1-i_2-i_3} \frac{(k_2+k_3-i_2-i_3)}{k_2-i_2-i_3}.
\]

The integrands \( f^{(y)}_{u^{(x)}(x,y,z)} f \) and \( f^{(z)}_{u^{(x)}(x,y,z)} f \) are defined analogously. Based on cyclic permutations, they have combinatorial coefficients \( B^{(y)} = B(i_2, i_3, i_1, k_2, k_3, k_1) \) and \( B^{(z)} = B(i_3, i_1, i_2, k_3, k_1, k_2) \), respectively.

Using homotopy operators, \( \text{Div}^{-1} \) is guaranteed by the following theorem.

Theorem 3. Let \( f = f(x, u^{(M)}(x)) \) be exact, i.e., \( f = \text{Div} F \) for some \( F(x, u^{(M-1)}(x)) \). Then, in the 2D case, \( F = \text{Div}^{-1} f = \left( \mathcal{H}_{u^{(x)}(x,y,z)} f, \mathcal{H}_{u^{(y)}(x,y,z)} f \right) \). Analogously, in 3D, \( F = \text{Div}^{-1} f = \left( \mathcal{H}_{u^{(x)}(x,y,z)} f, \mathcal{H}_{u^{(y)}(x,y,z)} f, \mathcal{H}_{u^{(z)}(x,y,z)} f \right) \).

Proof. A proof for the 2D case is given in Poole (2009). The 3D case could be proven with similar arguments.

Unfortunately, the outcome of the homotopy operator is not unique. The integral in the 1D case has a harmless arbitrary constant. However, in the 2D and 3D cases there are infinitely many non-trivial choices for \( F \). From vector calculus we know that \( \text{Div} \text{Curl} K = 0 \). Thus, the addition of \( \text{Curl} K \) to \( F \) would not alter \( \text{Div} F \). More precisely, for \( K = (D_y \theta, -D_z \theta, D_x \eta, D_y \xi, D_z \xi, -D_x \theta) \) in 2D, or for \( K = (D_y \eta, -D_z \xi, D_z \theta, -D_x \eta, D_z \xi, -D_x \theta) \) in 3D, \( \text{Div} G = \text{Div} (F + K) = \text{Div} F \), where \( \theta, \eta, \) and \( \xi \) are arbitrary functions. To obtain a concise result for \( \text{Div}^{-1} \), Poole and Hereman (2010) developed an algorithm that removes curl terms.
4. An Algorithm for Computing a Conservation Law

To compute a conservation law we construct a candidate density by taking a linear combination (with undetermined coefficients) of terms that are invariant under the scaling symmetry of the PDE. The total time derivative of the candidate is computed and evaluated on (4), thus removing all time derivatives from the problem. The resulting expression must be exact. Thus, we use Theorem 1 from Section 3 to derive the linear system that gives the undetermined coefficients. Substituting the coefficients into the candidate leads to a valid density. To compute the associated flux we invert the divergence with the homotopy operator. To illustrate the subtleties of the algorithm we intersperse the steps of the algorithm with two examples, viz., the ZK and KP equations.

4.1. Computing the Scaling Symmetry

A PDE has a unique set of Lie-point symmetries which may include translations, rotations, dilations, and other symmetries (Bluman et al., 2010). The application of such symmetries allows one to generate new solutions from known solutions. We will use only one type of Lie-point symmetry, namely, the scaling or dilation symmetry, to formulate a “candidate density.”

Let us assume that a PDE has a scaling symmetry. For example, the ZK equation (6) is invariant under the scaling symmetry

\[(x, y, t, u) \rightarrow (\lambda^{-1}x, \lambda^{-1}y, \lambda^{-3}t, \lambda^2u), \tag{31}\]

where \(\lambda\) is an arbitrary scaling parameter. To compute (31) with linear algebra, each variable is assigned an unknown weight.

**Definition 7.** The weight of a variable is defined as the exponent \(p\) in the factor \(\lambda^p\) that multiplies the variable. For the scaling symmetry \(x \rightarrow \lambda^{-p}x\), the weight is denoted \(W(x) = -p\). Total derivatives carry a weight. Indeed, if \(W(x) = -p\), then \(W(D_x) = p\).

When a PDE is invariant under a scaling symmetry, we say that it is *uniform in rank.*

**Definition 8.** The rank of a monomial is the sum of the weights of the variables in the monomial. A differential function is *uniform in rank* if all monomials in the differential function have the same rank. A differential function is *multi-uniform in rank* if it is uniform in rank for more than one scaling symmetry.

**Example 3.** The term \(\alpha uu_x\) in the ZK equation (6) has rank \(W(u) + W(u) + W(D_x)\). Using (31), \(W(u) + W(u) + W(D_x) = 5\). The ZK equation is *uniform in rank* (invariant under a scaling symmetry) since all other terms have a rank of 5. Note that the parameter \(\alpha\) is assumed to have zero weight.

**Step 1-ZK** (Computing the scaling symmetry). We assume that the PDE is uniform in rank. Under that assumption, we can form a system of weight-balance equations corresponding to the terms in the PDE. The solution of that system determines the scaling symmetry.
The weight-balance equations for the ZK equation (6) are

\[ W(u) + W(D_t) = 2W(u) + W(D_x) = W(u) + 3W(D_x) = W(u) + W(D_x) + 2W(D_y). \]  (32)

The parameters \( \alpha \) and \( \beta \) are assumed to have zero weight, hence, they are not included in the system. Solving the linear system gives

\[ W(u) = 2W(D_x), \quad W(D_t) = 3W(D_x), \quad W(D_y) = W(D_x). \]  (33)

To get (31), set \( W(D_x) = 1 \). The solution to the weight-balance system is then

\[ W(u) = 2, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 1. \]  (34)

Since \( \lambda \) is arbitrary, the weight of at least one of the variables will remain undetermined. We express the weights of the dependent variables in terms of those of the independent variables, and then assign a value(s) to the arbitrary weight(s). The choice, \( W(D_x) = 1 \) in (34) is convenient, however, all undetermined weights must be chosen in such a way that all weights are positive.

If the system of weight-balance equations has no solution, or requires variables to have a zero weight, then the PDE does not have a scaling symmetry. As shown in Section 6.3, it is possible to induce a scaling symmetry by either giving weights to existing parameters in the PDE, or by multiplying some terms in the PDE by weighted parameters. Any such auxiliary parameters can be set to one later.

**Step 1-KP** (Computing the scaling symmetry). Assuming that the evolution system for the KP equation (12) is uniform in rank, one gets the weight-balance system,

\[ W(u) + W(D_y) = W(v) \]
\[ W(u) + W(D_t) + W(D_x) = 2W(u) + 2W(D_x) = W(u) + 4W(D_x) = W(v) + W(D_y). \]  (35)

The parameters \( \alpha \) and \( \sigma^2 \) have zero weights. The solution of this linear system is

\[ W(u) = 2W(D_x), \quad W(v) = 4W(D_x), \quad W(D_t) = 3W(D_x), \quad W(D_y) = 2W(D_x). \]  (36)

Choosing \( W(D_x) = 1 \) leads to the scaling symmetry

\[ (x, y, t, u, v) \rightarrow (\lambda^{-1}x, \lambda^{-2}y, \lambda^{-3}t, \lambda^2u, \lambda^4v). \]  (37)

4.2. **Constructing a Candidate Component**

Conservation law (2) must hold on all solutions of the PDE. Therefore, the conserved density and its associated flux must obey the scaling symmetry of the PDE, that is, the conservation law itself must be uniform in rank. Thus, adhering to the scaling symmetry of the PDE, we can construct a candidate density that is a linear combination of terms of a pre-selected rank.

**Step 2-ZK** (Building the candidate density). In this step we will construct a candidate density with, say, rank 6 for the ZK equation (6).
(a) Construct a list, $P$, of differential terms containing all powers of dependent variables and products of dependent variables that have rank 6 or less. In regard to (34), $P = \{u^3, u^2, u\}$.

(b) Bring all of the terms in $P$ up to rank 6 and put them into a new list, $Q$. This is done by applying the total derivative operators with respect to the space variables. Taking the terms in $P$, $u^3$ has rank 6 and is placed directly into $Q$. The term $u^2$ has rank 4 and can be brought up to rank six in three ways: either by applying $D_x$ twice, or by applying $D_y$ twice, or by applying each of $D_x$ and $D_y$ once. All three possibilities are considered and the resulting terms are put into $Q$. Similar to $u^2$, the term $u$ can be brought up to rank 6 in five ways, and all results are placed into $Q$. Doing so, $P$ is replaced by

$$Q = \{u^3, u^2, uu_{2x}, u^2_y, uu_{2y}, u_x u_y, uu_{xy}, u_{4x}, u_{3xy}, u_{2x2y}, u_{x3y}, u_{4y}\}, \quad (38)$$

in which all monomials are of rank 6.

(c) With the goal of constructing a nontrivial density with the least number of terms, remove all terms that are divergences or are divergence-equivalent to other terms in $Q$.

**Definition 9.** A term or expression $f$ is a divergence when there exists a vector $F$ such that $f = \text{Div } F$. In the one-dimensional case, $f$ is a total derivative when there exists a function $F$ such that $f = D_x F$. Note that $D_x f$ is essentially a one-dimensional divergence, so from here onwards, the term “divergence” will also cover the one-dimensional (total derivative) scenario. Two or more terms are divergence-equivalent when a linear combination of the terms is a divergence.

Removing divergences and divergence-equivalent terms is algorithmic and best accomplished by using the Euler operator. Apply the Euler operator (19) to each term in (38) to get

$$L_{u(x,y)} Q = \{3u^2, -2u_{2x}, 2u_{2x}, -2u_{2y}, 2u_{2y}, -2u_{xy}, 2u_{xy}, 0, 0, 0, 0\}. \quad (39)$$

By Theorem 1, divergences are terms corresponding to 0 in (39). Hence, $u_{4x}$, $u_{3xy}$, $u_{2x2y}$, $u_{x3y}$, and $u_{4y}$ are divergences and can be removed from $Q$. Now, all divergence-equivalent terms will be removed. Following Hereman et al. (2005), form a linear combination of the terms that remained in (39) with undetermined coefficients $p_i$, and set it equal to zero,

$$3p_1 u^2 - 2p_2 u_{2x} + 2p_3 u_{2x} - 2p_4 u_{2y} + 2p_5 u_{2y} - 2p_6 u_{xy} + 2p_7 u_{xy} = 0. \quad (40)$$

After gathering like terms and forming a linear system of equations consisting of coefficients, we find that $p_1 = 0, p_2 = p_3, p_4 = p_5$, and $p_6 = p_7$. Thus, the terms with coefficients $p_3, p_5$, and $p_7$ are divergence-equivalent to the terms with coefficients $p_2, p_4$, and $p_6$, respectively. For each divergence-equivalent pair, the terms with the highest order are removed from $Q$ in (38). With all divergences and divergence-equivalent terms removed, $Q = \{u^3, u^2_x, u^2_y, uxuy\}$.

(d) A candidate density is obtained by forming a linear combination of the remaining terms in $Q$ using undetermined coefficients $c_i$. Thus, the candidate density with a rank of 6 for the ZK equation is

$$\rho = c_1 u^3 + c_2 u^2_x + c_3 u^2_y + c_4 u_x u_y. \quad (41)$$
To determine the undetermined coefficients in (41), we must compute $D_t \rho$ and eliminate $u_t$ (and all its differential consequences). That requires that the PDE is an evolution equation or can be written in evolution form in $t$. The ZK equation (6) is an evolution equation.

As shown in (12), the KP equation can be written as an evolution system with respect to independent variable $y$. Thus, one could either interchange the variables and compute the density, or alternatively, construct a “candidate” for the $y$-component of the flux, as we will do in a Step 2-KP below.

To complicate matters further, the conservation laws for the KP equation, (14) and (15), involve an arbitrary function $f = f(t)$. The weight of $f(t)$ depends on the degree if $f(t)$ is polynomial; whereas no weight can be assigned if $f(t)$ is non-polynomial. In general, working with undetermined functional (instead of constant) coefficients $f(x,y,t)$ would require a sophisticated solver for PDEs for $f$ (see Wolf (2002)). Therefore, we can not automatically compute (14) and (15) with our method. However, our algorithm can find conservation laws with explicit variable coefficients, e.g., $tx^2, txy$, etc., as long as the degree is specified. Allowing such coefficients causes candidate densities to have a negative rank. By inspecting several conserved densities with explicit variable coefficients it is often possible to conjecture (and subsequently test) the form of a conservation law with arbitrary functional coefficients.

**Step 2-KP (Building a candidate for the y-component of the flux).** For the KP equation, we will compute a candidate for the $y$-component of the flux with a rank equal to $-3$, whose terms are multiplied by $c_i t^n x^m y^p$, where $n + m + p \leq 3$ and $n, m, p$ are positive integers.

(a) Construct lists of independent variable coefficients and dependent variable terms so that the combined rank is equal to $-3$. From (37),

$$W(u) = 2, \quad W(v) = 4, \quad W(D_t) = 3, \quad W(D_x) = 1, \quad W(D_y) = 2.$$  \hspace{1cm} (42)

Consequently, $W(t) = -3, W(x) = -1, \text{ and } W(y) = -2$. Table 1 shows the factors $t^n x^m y^p$, up to degree 3, together with the dependent variable terms they will be multiplied to so that the resulting product has a rank equal to $-3$. Since we are computing the $y$-component of the flux, only $x$- and $t$-derivatives of dependent variables are being considered.

(b) Combine the terms in Table 1 to create a list of all possible terms with rank $-3$,

$$Q = \{tx^2 u, xy^2 u, t y u, y^3 u_x, t x y u_x, t^2 u_x, t^2 x u^2, t y^2 u^2, t^2 x u^2, t^2 x v, t y^2 v, t^2 y u_x, t^2 y u_t, t^2 y u_{3x}, t^2 y v_x, t^3 u, t^3 u^3, t^3 u^2, t^3 y_v, t^3 u_{x_t}, t^3 u_{u_x}, t^3 u_{2x}, t^3 u_{3x}, t^3 v_{2x}\}.$$  \hspace{1cm} (43)

(c) Remove all divergences and divergence-equivalent terms. Apply the Euler operator to each term in (43). Next, linearly combine the resulting terms to get

$$p_1 \left( \begin{array}{c} t x^2 \\ 0 \end{array} \right) + p_2 \left( \begin{array}{c} x y^2 \\ 0 \end{array} \right) + p_3 \left( \begin{array}{c} t y \\ 0 \end{array} \right) - p_5 \left( \begin{array}{c} t y \\ 0 \end{array} \right) + p_7 \left( \begin{array}{c} 2 t^2 x u \\ 0 \end{array} \right) + p_8 \left( \begin{array}{c} 2 t y^2 u \\ 0 \end{array} \right) + p_{11} \left( \begin{array}{c} 0 \\ t^2 x \end{array} \right) + p_{12} \left( \begin{array}{c} 0 \\ t y^2 \end{array} \right) - p_{14} \left( \begin{array}{c} 2 t y \\ 0 \end{array} \right) + p_{17} \left( \begin{array}{c} 3 t^3 u^2 \\ 0 \end{array} \right) + p_{18} \left( \begin{array}{c} t^3 v \\ t^3 u \end{array} \right) - p_{19} \left( \begin{array}{c} 2 t^3 u_{2x} \\ 0 \end{array} \right) + p_{21} \left( \begin{array}{c} 2 t^3 u_{2x} \\ 0 \end{array} \right) = 0.$$  \hspace{1cm} (44)
Table 1: Factors $t^n x^m y^p$ paired with dependent variable terms whose product has a rank equal to $-3$.

<table>
<thead>
<tr>
<th>Monomial factors $t^n x^m y^p$</th>
<th>Dependent Variable Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>Coefficient</td>
</tr>
<tr>
<td>-5</td>
<td>$tx^2, xy^2, ty$</td>
</tr>
<tr>
<td>-6</td>
<td>$yx^2, txy, t^2$</td>
</tr>
<tr>
<td>-7</td>
<td>$t^2x, ty^2$</td>
</tr>
<tr>
<td>-8</td>
<td>$t^2y$</td>
</tr>
<tr>
<td>-9</td>
<td>$t^3$</td>
</tr>
</tbody>
</table>

where the subscript of the undetermined coefficient, $p_i$, corresponds to the $i$th term in $Q$. Missing $p_i$ correspond to terms that are divergences. After gathering like terms, setting their coefficients equal to zero, and solving the resulting linear system for the $p_i$, we get $p_1 = p_2 = p_7 = p_9 = p_{11} = p_{12} = p_{17} = p_{18} = 0$, $p_3 = p_5 + 2p_{14}$, and $p_{19} = p_{21}$. Thus, both terms with coefficients $p_3$ and $p_{14}$ are divergence-equivalent to the term with coefficient $p_5$. Likewise, the term with coefficient $p_{21}$ is divergence-equivalent to the term with coefficient $p_{19}$. For each divergence-equivalent pair, the terms with the highest order are removed from (43). After all divergences and divergence-equivalent terms are removed

$$Q = \{tx^2u, xy^2u, tyu, t^2xu^2, ty^2u^2, t^2xv, ty^2v, t^3u^3, t^3uv, t^3u_x^2\}. \quad (45)$$

(d) A linear combination of the terms in (45) with undetermined coefficients $c_i$ yields the candidate (of rank $-3$) for the $y$-component of the flux,

$$J^2 = c_1 tx^2u + c_2 xy^2u + c_3 txy + c_4 t^2xu^2 + c_5 ty^2u^2 + c_6 t^2xv + c_7 ty^2v + c_8 t^3 u^3 + c_9 t^3u v + c_{10} t^3u_x^2. \quad (46)$$

4.3. Evaluating the Undetermined Coefficients

All, part, or none of the candidate density (41) may be an actual density for the ZK equation. It is also possible that the candidate is a linear combination of two or more independent densities. The true nature of the density will be revealed by computing values for the undetermined coefficients. Using (2), we see that $D_t \rho = -\text{Div} \mathbf{J}$, so $D_t \rho$ must be a divergence with respect to the space variables. This requirement leads to an algorithm for computing the undetermined coefficients.

**Step 3-ZK (Computing the undetermined coefficients).** To compute the undetermined coefficients, we form a system of linear equations for these coefficients. As part of the solution process, we also generate compatibility conditions for the constant parameters in the PDE, if present.

(a) Compute the total $t$-derivative of (41),

$$D_t \rho = 3c_1 u^2 u_t + 2c_2 u_x u_{tx} + 2c_3 u_y u_{ty} + c_4 (u_{tx} u_y + u_x u_{ty}). \quad (47)$$
Let \( E = -D_t \rho \) after all \( u_t, u_{tx} \), etc. have been replaced using (6). This yields
\[
E = 3c_1 u^2(\alpha uu_x + \beta (u_{3x} + u_{x2y})) + 2c_2 u_x(\alpha uu_x + \beta (u_{3x} + u_{x2y}))_x \\
+ 2c_3 u_y(\alpha uu_x + \beta (u_{3x} + u_{x2y}))_y + c_4 (u_y(\alpha uu_x + \beta (u_{3x} + u_{x2y}))_x \\
+ u_x(\alpha uu_x + \beta (u_{3x} + u_{x2y})))_y). \tag{48}
\]

(b) By the continuity equation (2), \( E = \text{Div} \, J \). Therefore, by Theorem 1, \( \mathcal{L}_{u(x)} E \equiv 0 \). Apply the Euler operator to (48) and set the result identically equal to zero:
\[
0 \equiv \mathcal{L}_{u(x)} E = -2((3c_1 \beta + c_3 \alpha) u_x u_{2y} + 2(3c_1 \beta + c_3 \alpha) u_y u_{xy} \\
+ 2c_4 \alpha u_x u_{xy} + c_4 \alpha u_y u_{2x} + 3(3c_1 \beta + c_2 \alpha) u_x u_{2x}). \tag{49}
\]

(c) Form a linear system of equations for the undetermined coefficients by setting each factor equal to zero thus satisfying (49). After eliminating duplicate equations, the system is
\[
3c_1 \beta + c_3 \alpha = 0, \quad c_4 \alpha = 0, \quad 3c_1 \beta + c_2 \alpha = 0. \tag{50}
\]

(d) Check for possible compatibility conditions on the parameters \( \alpha \) and \( \beta \) in (50). This is done by setting each \( c_i = 1 \), one at a time, and algebraically eliminating the other undetermined coefficients. Consult Göktas and Hereman (1997) for details about searching for compatibility conditions. System (50) is compatible for all nonzero \( \alpha \) and \( \beta \).

(e) Solve (50), taking into account the compatibility conditions (if applicable). Here,
\[
c_2 = -\frac{3\beta}{\alpha} c_1, \quad c_3 = -\frac{3\beta}{\alpha} c_1, \quad c_4 = 0, \tag{51}
\]
where \( c_1 \) is arbitrary. We set \( c_1 = 1 \) so that the density (41) will be normalized on the highest degree term, yielding
\[
\rho = u^3 - \frac{3\beta}{\alpha} (u_x^2 + u_y^2). \tag{52}
\]

**Step 3-KP** (Computing the undetermined coefficients).

(a) Compute the total \( y \)-derivative of (46),
\[
D_y J^2 = (c_1 x + 2c_4 tu)tx u_y + c_2 xy(2u + yu_y) + (c_3 t + 2c_5 ty)(u + yu_y) \\
+ c_4 t^2 xy + c_7 ty(2v + yv_y) + 3c_8 t^3 u^2 y_y + c_9 t^3 (u_y v + uv_y) + 2c_{10} t^3 u_x u_{xy}. \tag{53}
\]
and replace all \( u_y \) (and differential consequences) using (12). Thus,
\[
E = -(c_1 x + 2c_4 tu)tx v - c_2 xy(2u + yv) - (c_3 t + 2c_5 ty)(u + yv) \\
+ \sigma^2 (c_4 t^2 x + c_7 ty^2 + c_9 t^3 u)(u_{tx} + \alpha u_x^2 + \alpha uu_x + u_{4x}) - 2c_7 tyv \\
- 3c_8 t^3 u^2 v - c_9 t^3 v^2 - 2c_{10} t^3 u_x v_x. \tag{54}
\]

(b) Apply the Euler operator to (54) and set the result identically equal to zero. This yields
\[
0 \equiv \mathcal{L}_{u(t,x)} = -(2c_2 xy + (c_3 - 2c_2 c_6) t + 2c_4 t^2 x v + 2c_5 ty(2u + yv) + 6c_8 t^3 u v \\
- 2\sigma^3 c_9 y^2 (\frac{4}{3} u_x + tu_{tx} + \alpha u_x^2 + \alpha uu_x + tu_{4x}) - 2c_{10} t^3 v_{2x} + c_1 t x^2 + c_2 x y^2 \\
+ (c_3 + 2c_7) ty + 2c_4 t^2 x u + 2c_5 t y^2 u + 3c_8 t^3 u^2 + 2c_9 t^3 v - 2c_{10} t^3 u_{2x}). \tag{55}
\]
(c) Form a linear system for the undetermined coefficients $c_i$. After duplicate equations and common factors have been removed, one gets

$$c_1 = 0, \ c_2 = 0, \ c_3 - 2\sigma^2c_6 = 0, \ c_3 + 2c_7 = 0, \ c_4 = 0, \ c_5 = 0, \ c_8 = 0, \ c_9 = 0, \ c_{10} = 0. \quad (56)$$

(d) Compute potential compatibility conditions on the parameters $\alpha$ and $\sigma$. Again, the system is compatible for all nonzero $\alpha$ and for $\sigma^2 = \pm 1$.

(e) Solve the linear system, yielding

$$c_1 = c_2 = c_4 = c_5 = c_6 = c_9 = c_{10} = 0, \ c_6 = \frac{1}{2}\sigma^2c_3, \ c_7 = -\frac{1}{2}c_3. \quad (57)$$

Set $c_3 = -2$ (to normalize the density) and substitute the result into (46), to obtain

$$J^2 = -2t\alpha u - tv(\sigma^2tx - y^2). \quad (58)$$

4.4. Completing the Conservation Law

Once we have one component of the conservation law, e.g., the density or a component of the flux, we can compute the remaining components using the homotopy operator.

Step 4-ZK (Computing the flux). For (6), $\rho$ is given in (52), so we will compute the flux $\mathbf{J}$. Once again, by the continuity equation (2), $\text{Div} \mathbf{J} = -D_t\rho = E$. Therefore, we must compute $\text{Div}^{-1}E$, where the divergence is with respect to $x$ and $y$. After substitution of (51) and $c_1 = 1$ into (48),

$$E = 3u^2(\alpha u_x + \beta u_{3x} + \beta u_{x2y}) - \frac{6\beta}{\alpha}u_x(\alpha u_x + \beta u_{3x} + \beta u_{x2y})x$$

$$- \frac{6\beta}{\alpha}u_y(\alpha u_x + \beta u_{3x} + \beta u_{x2y})y. \quad (59)$$

Now we apply the 2D homotopy operator. The integrands (24) and (26) are

$$I_{u(x,y)}^{u(x,y)} = 3\alpha u^4 + \beta \left(9u^2(u_{2x} + 2u_{2y}) - 6u(3u_x^2 + u_y^2)\right) + \frac{\partial^2}{\alpha} \left(6u_{2x}^2 + 5u_{2y}^2 + 3u(u_{2xy} + u_{4y}) - u_x(12u_{3x} + 7u_{x2y}) - u_y(3u_{3y} + 8u_{x2y}) + \frac{5}{2}u_{2x}u_{2y}\right), \quad (60)$$

and

$$I_{u(x,y)}^{(u)} = 3\beta(uu_{x2y} - 4u_x + u_{y2}) - \frac{\sigma^2}{\alpha} \left(3u(u_{3xy} + u_{x3y}) + u_x(13u_{2xy} + 3u_{3y}) + 5u_y(u_{3x} + 3u_{2xy}) - 9u_{xy}(u_{2x} + u_{2y})\right), \quad (61)$$

respectively. The 2D homotopy operator formulas (23) yield

$$\mathcal{H}_{u(x,y)}^{(u)} = \int_0^1 \left(\mathcal{T}_{u(x,y)}^{(u)} \right)[\lambda u] \frac{d\lambda}{\lambda}$$

$$= \frac{3}{4}\alpha u^4 + \beta \left(3u^2(u_{2x} + 2u_{2y}) - 2u(3u_x^2 + u_y^2)\right) + \frac{\sigma^2}{\alpha} \left(3u_{2x}^2 + 5u_{2y}^2 + 3u_{x2y}^2 \right. \left. + \frac{3}{4}u(u_{2xy} + u_{4y}) - u_x(6u_{3x} + 7u_{x2y}) - u_y(\frac{3}{2}u_{3y} + 4u_{2xy}) + \frac{5}{4}u_{x2}u_{2y}\right), \quad (62)$$

and

$$\mathcal{H}_{u(x,y)}^{(u)} = \int_0^1 \left(\mathcal{T}_{u(x,y)}^{(u)} \right)[\lambda u] \frac{d\lambda}{\lambda}$$

$$= \beta(uu_{x2y} - 4u_x + u_{y2}) - \frac{\sigma^2}{\alpha} \left(3u(u_{3xy} + u_{x3y}) + u_x(13u_{2xy} + 3u_{3y}) + 5u_y(u_{3x} + 3u_{2xy}) - 9u_{xy}(u_{2x} + u_{2y})\right). \quad (63)$$
The flux \( J = \left( \mathcal{H}_{u(x,y)}^{(x)} E, \mathcal{H}_{u(x,y)}^{(y)} E \right) \) has a curl term, \( K = (D_y \theta, -D_x \theta) \), with
\[
\theta = 2 \beta u^2 y + \frac{\partial^2}{\partial x^2} \left( 3 u (u_{xy} + u_{3y}) + 5 (2 u_x u_{xy} + 3 u_y u_{2y} + u_{2x} u_y) \right). \tag{64}
\]

After removing \( K \), we have conservation law (9).

**Step 4-KP** (Computing the density and the \( x \)-component of the flux). To complete the conservation law for (11) for which we know the \( y \)-component of the flux (58), we must compute the density and the \( x \)-component of the flux. By the continuity equation (2), \( D_t \rho + D_x J^1 = -D_y J^2 = E \). Thus, to find \( (\rho, J^1) \), we must compute \( \text{Div}^{-1} E \), where this time the divergence is with respect to \( t \) and \( x \). Since we are inverting a 2D divergence, we use (23). After substituting (57) and \( c_3 = -2 \) into (54),
\[
E = 2 tu - (t^2 x - \sigma^2 ty^2) (u_{tx} + \alpha u_x^2 + \alpha uu_{2x} + u_{4x}). \tag{65}
\]

The integrands for the homotopy operator are
\[
\begin{align*}
I_{u(t,x)}^{(t)} E &= -\frac{1}{2} (u D_x - u_x I) \frac{\partial E}{\partial u_{tx}} = \frac{1}{2} (t^2 u - (t^2 x - \sigma^2 ty^2) u_x), \tag{66} \\
I_{v(t,x)}^{(t)} E &= 0, \tag{67} \\
I_{u(t,x)}^{(x)} E &= u \frac{\partial E}{\partial u_x} - \frac{1}{2} (u D_t - u_t I) \frac{\partial E}{\partial u_{tx}} - (u D_x - u_x I) \frac{\partial E}{\partial u_{tx}} - (u D_x - u_x I) \frac{\partial E}{\partial u_{tx}} - (u D_x - u_x I) \frac{\partial E}{\partial u_{tx}} \\
&= \alpha t^2 u + t^2 u_{2x} - (t^2 x - \sigma^2 ty^2) \left( \frac{1}{2} u_t + 2 \alpha uu_x + u_{3x} \right) + (tx - \frac{1}{2} \sigma^2 y^2) u, \tag{68} \\
I_{v(t,x)}^{(x)} E &= 0. \tag{69}
\end{align*}
\]

Hence,
\[
\begin{align*}
\hat{\rho} &= \mathcal{H}_{u(t,x)}^{(t)} E = \int_0^1 \left( I_{u(t,x)}^{(t)} E + I_{v(t,x)}^{(t)} E \right) [\lambda u] \frac{d\lambda}{\lambda} \\
&= \int_0^1 \left( \frac{1}{2} (t^2 u - (t^2 x - \sigma^2 ty^2) u_x) \right) d\lambda = \frac{1}{2} (t^2 u - (t^2 x - \sigma^2 ty^2) u_x), \tag{70} \\
\hat{J}^1 &= \mathcal{H}_{u(t,x)}^{(x)} E = \int_0^1 \left( I_{u(t,x)}^{(x)} E + I_{v(t,x)}^{(x)} E \right) [\lambda u] \frac{d\lambda}{\lambda} \\
&= \int_0^1 \left( \alpha t^2 u + t^2 u_{2x} - (t^2 x - \sigma^2 ty^2) \left( \frac{1}{2} u_t + 2 \alpha uu_x + u_{3x} \right) + (tx - \frac{1}{2} \sigma^2 y^2) u \right) d\lambda \\
&= \frac{1}{2} \alpha t^2 u + t^2 u_{2x} - (t^2 x - \sigma^2 ty^2) \left( \frac{1}{2} u_t + \alpha uu_x + u_{3x} \right) + (tx - \frac{1}{2} \sigma^2 y^2) u. \tag{71}
\end{align*}
\]

The solution \( (\hat{\rho}, \hat{J}^1) \) contains a curl term, \( K = (D_x \theta, -D_t \theta) \), with \( \theta = -\frac{1}{2} (t^2 x - \sigma^2 ty^2) u \). After removing the curl term, we find
\[
\begin{pmatrix}
\hat{\rho} \\
\hat{J}^1
\end{pmatrix} = \begin{pmatrix} t^2 \left( \frac{1}{2} \alpha u_x^2 + u_{2x} \right) + (\sigma^2 ty^2 - t^2 x) (u_t + \alpha uu_x + u_{3x}) \end{pmatrix}. \tag{72}
\]

The computed conservation law is the same as (14) where \( f = t^2 \) and \( v = u_y \).
5. A Generalized Conservation Law for the KP Equation

The generalization of (72) to (14) is based on inspection of the conservation laws shown in Table 2. Indeed, these conservation laws suggest a density of the form \( f(t)u \), where \( f(t) \) is an arbitrary function. The corresponding flux would be harder to guess. However, it can be computed as follows. Since the KP equation (12) is an evolution equation in \( y \), we construct a suitable candidate for \( J^2 \). Guided by the results in Table 2, we take

\[
J^2 = c_1 f'(t) y u + c_2 f'(t) y^2 v + c_3 f(t) x v,
\]

where \( c_1, c_2, \) and \( c_3 \) are undetermined coefficients, and \( u_y \) is replaced by \( v \) in agreement with (12). As before, we compute \( D_y J^2 \) and replace \( u_y \) and \( v_y \) using (12). Doing so,

\[
E = D_y J^2 = c_1 f' u + (c_1 + 2c_2) f' y v - (\sigma^2 c_2 f' y^2 + \sigma^2 c_3 f x)(u_t + \alpha u_x + \alpha u_2 + u_4),
\]

By (2), \( D_y J^2 = -\text{Div}(\rho, J^1) \). Therefore, by Theorem 1,

\[
0 \equiv L_{u(t,x)} E = \left( \frac{c_1 - \sigma^2 c_3}{c_1 + 2c_2} f' y \right).
\]

Clearly, \( c_2 = -\frac{1}{2} c_1 \) and \( c_3 = \sigma^2 c_1 \). If we set \( c_1 = -1 \), we obtain the \( y \)-component of (14), where \( v = u_y \). Application of the homotopy operator (in this case to an expression with arbitrary functional coefficients) yields \( (\rho, J^1) \).

Due to the presence of an arbitrary function \( f(t) \), it was impossible to directly compute (14) with our algorithm. However, pattern matching with the results in Table 2 and some interactive work lead to (14), which can be then be verified with CONSERVATION LAWS MD.M. Conservation law (15) was obtained in a similar way.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Conservation Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>( D_t(u) + D_x \left( \frac{1}{2} \alpha u_x + u_2 - x(u_t + \alpha u_x + u_3) \right) - D_y (\sigma^2 x v) = 0 )</td>
</tr>
</tbody>
</table>
| 2    | \( D_t(t u) + D_x \left( t \left( \frac{1}{2} \alpha u_x + u_2 \right) + \left( \frac{1}{2} \sigma^2 y^2 - t x \right)(u_t + \alpha u_x + u_3) \right) \)  
|      | \( - D_y \left( y u - v(\frac{1}{2} y^2 - \sigma^2 t x) \right) = 0 \). |
| -1   | \( D_t \left( t^2 u \right) + D_x \left( t^2 \left( \frac{1}{2} \alpha u_x + u_2 \right) + (\sigma^2 t^2 y^2 - t^3 x)(u_t + \alpha u_x + u_3) \right) \)  
|      | \( - D_y \left( 2 t y u - v(t y^2 - \sigma^2 t^2 x) \right) = 0 \) |
| -4   | \( D_t \left( t^3 u \right) + D_x \left( t^3 \left( \frac{1}{2} \alpha u_x + u_2 \right) + \left( \frac{3}{2} \sigma^2 t^2 y^2 - t^3 x \right)(u_t + \alpha u_x + u_3) \right) \)  
|      | \( - D_y \left( 3 t^2 y u - v(3 t^2 y^2 - \sigma^2 t^3 x) \right) = 0 \) |
6. Applications

In this section we apply our algorithm to a variety of (2+1)- and (3+1)-dimensional non-linear PDEs. These PDEs highlight many of the issues that arise when using the algorithm.

6.1. The Sawada-Kotera Equation in 2D

The (1+1)-dimensional Sawada-Kotera (SK) equation belongs to a family of fifth-order KdV-type equations. As shown by Göktaş and Hereman (1997), the SK equation has infinitely many conservation laws and is known to be completely integrable. The (2+1)-dimensional SK equation,

\[
\frac{\partial u}{\partial t} = 5u^2_1 + 5uu_3 + 5uu_y + 5u_xu_2 + 5u_{2xy} + u_{5x} - 5\frac{\partial^{-1}}{\partial x}u_{2y} + 5u_x\frac{\partial^{-1}}{\partial y},
\]  

(76)

with \( u(x,y,t) \) was derived by Konopelchenko and Dubrovsky (1984) as a higher-dimensional completely integrable equation. Our algorithm can not handle the integral terms like \( \frac{\partial^{-1}}{\partial x}f \). Therefore, setting \( v = \frac{\partial^{-1}}{\partial x}u_y \) allows us to write (76) as a system of evolution equations in \( y \) :

\[
v_y = -\frac{1}{5}u_t + u^2_1 + uu_3 + uu_2 + u_xv + \frac{1}{5}u_{5x} + u_xv, \quad u_y = v_x.
\]

(77)

Application of our algorithm to (77) then yields

\[
\begin{align*}
  &\mathcal{D}_t(fu) - \mathcal{D}_x\left(5f\left(\frac{1}{3}u^3 + uu_2 + u_{xy} + \frac{1}{5}u_{4x} - f'yu\right) + \mathcal{D}_y\left(5f - f'yu\right)\right) = 0, \\
  &\mathcal{D}_t(fyu) - \mathcal{D}_x\left(5fy\left(\frac{1}{3}u^3 + uu_2 + u_{xy} + \frac{1}{5}u_{4x} + \left(5fx - \frac{1}{2}f'y^2\right)v\right)\right) \\
  &\quad + \mathcal{D}_y\left((5fx - \frac{1}{2}f'y^2)u + 5fyv\right) = 0,
\end{align*}
\]

(79)

where \( f = f(t) \) is an arbitrary function.

Note that the densities in (78) and (79) are identical to those in (14) and (15) for the KP equation. These two densities occur often in (2+1)-dimensional PDEs that have a \( u_{tx} \) instead of a \( u_t \) term, as show in the next example.

6.2. The Khokhlov-Zabolotskaya Equation in 2D and 3D

The Khokhlov-Zabolotskaya (KZ) equation or dispersionless KP equation, describes the propagation of sound in non-linear media in two or three space dimensions (Sanders and Wang, 1997a). The (2+1)-dimensional KZ equation,

\[
(u_t - uu_x)_x - u_{2y} = 0,
\]

(80)

with \( u(x,y,t) \) can be written as a system of evolution equations in \( y \),

\[
u_y = v, \quad v_y = u_{tx} - u_x^2 - uu_{2x},
\]

(81)
by setting \( v = u_y \). Again, two familiar densities appear in the following conservation laws,

\[
\begin{align*}
D_t(u_x) - D_x(uu_x) - D_y(u_y) &= 0, \\
D_t(fu) - D_x(f^2u^2 + (fx + \frac{1}{2}f'y^2)(u_t - uu_x)) + D_y((fx + \frac{1}{2}f'y^2)uy - f'yu) &= 0, \\
D_t(fyu) - D_x(fyu^2 + (fxy + \frac{1}{6}f'y^3)(u_t - uu_x)) + D_y((fxy + \frac{1}{6}f'y^3)uy - (fx + \frac{1}{2}f'y^2)u) &= 0.
\end{align*}
\]

(82)  (83)  (84)

The computation of conservation laws for the (3+1)-dimensional KZ equation,

\[
(u_t - uu_x)x - u_{2y} - u_{2z} = 0,
\]

(85)

where \( u(x, y, z, t) \), is more difficult. This equation can be written as a system of evolution equations in either \( y \) or \( z \). Although the intermediate results will differ, either choice will lead to equivalent conservation laws. Writing (85) as an evolution system in \( z \),

\[
u_z = v, \quad v_z = uu_x - u_x^2 - uu_{2x} - u_{2y},
\]

(86)

we were able to compute a variety of conservation laws whose densities are shown in Table 3.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Densities</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>( \rho_1 = xu_x )</td>
</tr>
<tr>
<td>0</td>
<td>( \rho_2 = xyu_x, \quad \rho_3 = xzu_x )</td>
</tr>
<tr>
<td>-1</td>
<td>( \rho_4 = tu )</td>
</tr>
<tr>
<td>-2</td>
<td>( \rho_5 = xyzu_x, \quad \rho_6 = (y^2 - z^2) xu_x )</td>
</tr>
<tr>
<td>-3</td>
<td>( \rho_7 = tyzu, \quad \rho_8 = tzu )</td>
</tr>
<tr>
<td>-4</td>
<td>( \rho_9 = t^2u, \quad \rho_{10} = (y^2 - 3yz^2) xu_x, \quad \rho_{11} = (3y^2z^2 - z^3) xu_x )</td>
</tr>
<tr>
<td>-5</td>
<td>( \rho_{12} = tyzu, \quad \rho_{13} = (y^2 - z^2)tu )</td>
</tr>
<tr>
<td>-6</td>
<td>( \rho_{14} = t^2yu, \quad \rho_{15} = t^2xzu, \quad \rho_{16} = (y^3z - yz^3) xu_x, \quad \rho_{17} = (y^4 - 6y^2z^2 + z^4) xu_x )</td>
</tr>
<tr>
<td>-7</td>
<td>( \rho_{18} = t^3u, \quad \rho_{19} = (y^3 - 3yz^2)tu, \quad \rho_{20} = (3y^2z^2 - z^3)tu )</td>
</tr>
<tr>
<td>-8</td>
<td>( \rho_{21} = t^2yzu, \quad \rho_{22} = (y^2 - z^2)t^2u, \quad \rho_{23} = (y^5 - 10y^3z^2 + 5yz^4) xu_x, \quad \rho_{24} = (5y^4z - 10y^2z^3 + z^5) xu_x )</td>
</tr>
</tbody>
</table>

Density \( \rho_1 = xu_x \) in Table 3 is part of conservation law

\[
D_t(xu_x) + D_x(\frac{1}{2}u^2 - xuux) - D_y(xu_y) - D_z(xu_z) = 0,
\]

(87)
which can be rewritten as
\[ D_t(u) - D_x\left(\frac{1}{2}u^2 + xu - uu_x\right) + D_y(xu_y) + D_z(xu_z) = 0 \]  
by swapping terms in the density and the \( x \)-component of the flux. More general, if a factor \( xu_x \) appears in a density then that factor can be replaced by \( u \). Doing so, all densities in Table 3 that can be expressed as \( \rho = P(y, z, t)u \). Introducing arbitrary functions \( f = f(y, z, t) \) and \( g = g(y, z, t) \), the conservation laws corresponding to the densities in Table 3 can be summarized as
\[
D_t(fu) - D_x\left(\frac{1}{2}fu^2 + (fx + g)(u_t - uu_x)\right) - D_y((fy + g_y)u - (fx + g)u_y) \\
- D_z((fx + g_z)u - (fx + g)u_z) = -(f_{2y} + f_{2z})xu - (g_{2y} + g_{2z} - f_t)u. \tag{89}
\]

Equation (89) is only a conservation law when the constraints \( \Delta f = 0 \) and \( \Delta g = f_t \) are satisfied, where \( \Delta = \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \). Thus \( f \) must be a harmonic function and \( g \) must satisfy the Poisson equation with \( f_t \) on the right hand side. Combining both equations produces the biharmonic equation \( \Delta^2 g = 0 \). As shown by Tikhonov and Samarskii (1963), \( \Delta^2 g = 0 \) has general solutions of the form
\[ g = y g_1(y, z) + g_2(y, z) \quad \text{and} \quad g = z g_1(y, z) + g_2(y, z), \tag{90} \]
where \( \Delta g_1 = 0 \) and \( \Delta g_2 = 0 \). Treating \( t \) as a parameter, four solutions for \( g(y, z, t) \) are
\[
g(y, z, t) = \frac{1}{2}y \partial_y^{-1} f_t(y, z, t), \tag{91} \\
g(y, z, t) = \frac{1}{2} \partial_y^{-1}(y f_t) = \frac{1}{2}(y \partial_y^{-1} f_t(y, z, t) - \partial_y^{-2} f_t(y, z, t)), \tag{92} \\
g(y, z, t) = \frac{1}{2}z \partial_z^{-1} f_t(y, z, t), \tag{93} \\
g(y, z, t) = \frac{1}{2} \partial_z^{-1}(z f_t) = \frac{1}{2}(z \partial_z^{-1} f_t(y, z, t) - \partial_z^{-2} f_t(y, z, t)). \tag{94} 
\]

Thus, \( g \) can be written in terms of \( f \). For every conservation law corresponding to the densities in Table 3, \( g \) can be computed explicitly using one of the equations in (91)-(94).

Conservation laws for the KZ equation have been reported in literature by Sharomet (1989) and Sanders and Wang (1997a). However, substitution of these results into (2) reveals inaccuracies. After bringing the mistake to their attention, Sanders and Wang (1997b) have since corrected their conservation law.

6.3. The Camassa-Holm Equation in 2D

The \((2+1)\)-dimensional Camassa-Holm (CH) equation,
\[ (u_t + ku_x - u_{tx} + 3uu_x - 2u_xu_{2x} - uu_{3x})_x + uu_y = 0, \tag{95} \]
for \( u(x, y, t) \) was derived in a study of water waves (Johnson, 2002) as an extension of the \((1+1)\)-dimensional CH equation, which is a completely integrable PDE due to Camassa and Holm (1993). A study by Gordoa et al. (2004), concluded that (95) is not completely integrable.
The CH equation (95) can be written as a system of evolution equations in $y$. Indeed,

$$u_y = v, \quad v_y = -(\alpha u_t + \kappa u_x - u_{t2x} + 3\beta uu_x - 2u_xu_{2x} - uu_{3x})_x.$$  

(96)

Note that we introduced auxiliary parameters $\alpha$ and $\beta$ as coefficients of the $u_t$ and $uu_x$ terms, respectively. Analysis of the weight-balance equations in Poole (2009) shows that $\alpha, \beta$ and $\kappa$ must carry weights to make (95) and (96) scaling invariant.

Without conditions on these parameters, we found two conservation laws:

$$D_t(f u) + D_x \left( \begin{array}{c} 2 \frac{1}{\alpha} f \left( \frac{3}{2} \beta u^2 + \kappa u - \frac{1}{2} u_x^2 - uu_{2x} - u_{ttx} \right) \left( \frac{1}{\alpha} f x - \frac{1}{2} f' y^2 \right)(\alpha u_t + \kappa u_x \\
+ 3\beta uu_x - 2u_xu_{2x} - uu_{3x} - u_{t2x} \right) \right) - D_y \left( \left( \frac{1}{\alpha} f x - \frac{1}{2} f' y^2 \right)u_y + f'y u \right) = 0,$$

(97)

$$D_t(f yu) + D_x \left( \begin{array}{c} 2 \frac{1}{\alpha} f y \left( \frac{3}{2} \beta u^2 + \kappa u - \frac{1}{2} u_x^2 - uu_{2x} - u_{ttx} \right) - y \left( \frac{1}{\alpha} f x - \frac{1}{2} f' y^2 \right)(\alpha u_t + \kappa u_x \\
+ 3\beta uu_x - 2u_xu_{2x} - uu_{3x} - u_{t2x} \right) \right) - D_y \left( \left( \frac{1}{\alpha} f x - \frac{1}{2} f' y^2 \right)yu_y - \left( \frac{1}{\alpha} f x - \frac{1}{2} f' y^2 \right)u \right) = 0,$$

(98)

where $f(t)$ is an arbitrary function. By setting $\alpha = \beta = 1$, the conservation laws correspond to (95). When adding weighted parameters to an equation, it is possible that the program returns conservation laws accompanied by conditions on the weighted parameters. If such conditions are not satisfied for $\alpha = \beta = 1$, then the conservation laws should be discarded.

7. Using the Program ConservationLawsMD.m

Before using ConservationLawsMD.m, all data files provided with the program, as well as additional data files created by the user, must be placed into one directory. Next, open the Mathematica notebook ConservationLawsMD.nb which contains instructions for loading the code. Executing the command ConservationLawsMD[] will open a menu, offering the choice of computing conservation laws for a PDE from the menu or from a data file prepared by the user. All PDEs listed in the menu have matching data files. An example of a data file is shown in Figure 1. The independent space variables must be $x$, $y$, and $z$. The symbol $t$ must be used for time. Dependent variables must be entered as $u_i$, $i = 1, \ldots, N$, where $N$ is the total number of dependent variables. In a (1+1)-dimensional case, the dependent variables (in Mathematica syntax) are $u[1][x,t]$, $u[2][x,t]$, etc. In a (3+1)-dimensional cases, $u[1][x,y,z,t]$, etc., where $t$ is always the last argument.

8. Conclusions

We have presented an algorithm and a software package, ConservationLawsMD.m, to compute conservation laws of nonlinear polynomial PDEs in multiple space dimensions. The algorithm uses tools from calculus, the calculus of variations, linear algebra, and differential geometry. Conservation laws have been computed for a variety of multi-dimensional PDEs, demonstrating the versatility of the software package.

The software is easy to use, runs fast, and has been tested on a variety of PDEs. Many of the test cases have been added to the menu of the program. In addition, the program allows
(** data file d_kd2d.m **)  
(** Menu item 2-10 **)  

(* *** (2+1)-dimensional Gardner equation *** *)  
(* as proposed by Konopelchenko and Dubrovsky (1984) *)

\[ eq[1] = D[u[1][x,y,t],y] - D[u[2][x,y,t],x]; \]
\[ eq[2] = -D[u[1][x,y,t],t] + D[u[2][x,y,t],x,3] + 6*beta*u[1][x,y,t]*D[u[1][x,y,t],x] \]
\[ - (3/2)*alpha^2*u[1][x,y,t]^2*D[u[1][x,y,t],x] \]
\[ + 3*D[u[2][x,y,t],y] - 3*alpha*D[u[1][x,y,t],x]*u[2][x,y,t]; \]

\text{diffFunctionListINPUT = \{eq[1],eq[2]\};}
\text{numDependentVariablesINPUT = 2;}
\text{independentVariableListINPUT = \{x,y\};}
\text{nameINPUT = ”(2+1)-dimensional Gardner Equation”;}
\text{noteINPUT = ”Any additional information can be put here.”;}
\text{parametersINPUT = \{alpha\};}
\text{All parameters without weight must be placed in this list.}
\text{weightedParametersINPUT = \{beta\};}
\text{Parameters that carry a weight must be placed in this list.}
\text{userWeightRulesINPUT = \{\};}
\text{Optional: the user can choose weights for variables.}
\text{rankRhoINPUT = Null;}
\text{Can be changed to a list of values if the user wishes to test several ranks at one}
\text{time. The program runs automatically when such values are given.}
\text{explicitIndependentVariablesInDensitiesINPUT = Null;}
\text{Can be set to 0, 1, 2, ..., specifying the maximum degree (n+m+p) of coefficients}
\text{c_{i}t^{n}x^{m}y^{p} in the density.}
\text{formRhoINPUT = \{\};}
\text{The user can give a density to be tested. This works only for evolution equations}
\text{in variable t.}

(* end of data file d_kd2d.m *)

Figure 1: Data file for the (2+1)-dimensional Gardner equation.

the user to test conservation laws either computed with other methods, obtained from
the literature, or conjectured after work with the code. The latter is particularly relevant for
finding conservation laws involving arbitrary functions, cf. the KP equation.

Currently, \textsc{ConservationLawsMD.m} has two major limitations: (i) the PDE must
either be an evolution equation or must correspond to a system of evolution equations, per-
haps after an interchange of variables or some other transformation; and (ii) the program
can only generate local polynomial densities and fluxes. The \textit{testing} capabilities of \textsc{Con-
servationLawsMD.m} are quite versatile. The code can be used to test conservation laws
involving smooth functions of the independent variables and the densities and fluxes are not restricted to polynomial differential functions.

Future versions of the program will be able to handle PDEs with mixed derivatives, transcendental nonlinearities, and PDEs that are not of evolution type.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation (NSF) under Grant No. CCF-0830783. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Mark Hickman (University of Canterbury, Christchurch, New Zealand) and Bernard Deconinck (University of Washington, Seattle) are gratefully acknowledged for valuable discussions. Undergraduate students Jacob Rezac, John-Bosco Tran, and Travis “Alan” Volz are thanked for their help with this project.

References


Poole, D., Hereman, W., 2009. HOMOTOPYINTEGRATOR.M: A Mathematica package for the application of the homotopy method for (i) integration by parts of expressions involving unspecified functions of one variable and (ii) the inversion of a total divergence involving unspecified functions of two or three independent variables; software available at http://inside.mines.edu/~whereman under scientific software.


