Symbolic algorithms for the Painlevé test, special solutions, and recursion operators for nonlinear PDEs.

Douglas Baldwin, Willy Hereman, and Jack Sayers

Abstract. This paper discusses the algorithms and implementations of three Mathematica packages for the study of integrability and the computation of closed-form solutions of nonlinear polynomial PDEs.

The first package, PainleveTest.m, symbolically performs the Painlevé integrability test. The second package, PDESpecialSolutions.m, computes exact solutions expressible in hyperbolic or elliptic functions. The third package, PDERecursionOperator.m, generates and tests recursion operators.

1. Introduction

The investigation of complete integrability of nonlinear partial differential equations (PDEs) is a nontrivial matter [HGC98]. Likewise, finding the explicit form of solitary wave and soliton solutions requires tedious, unwieldy computations which are best performed using computer algebra systems. For example, the symbolic computation of solitons with Hirota’s direct method and the homogenization method, are covered in [HN97, HZ97].

Recently, progress has been made using Mathematica and Maple in applying the inverse scattering transform (IST) method to compute solitons for the Camassa-Holm equation [J03]. Before applying the IST method (a nontrivial exercise in analysis!), one would like to know if the PDE is completely integrable or what elementary travelling wave solutions exist.

This is where the symbolic algorithms and packages presented in this paper come into play.

In this paper, we introduce three algorithms and related Mathematica packages [BH03] which may greatly aid the investigation of integrability and the search for exact solutions.

In Section 2 we present the algorithm for the well-known Painlevé integrability test [AC91, C00, C99], which was recently implemented as PainleveTest.m. Section 3 outlines the algorithm behind PDESpecialSolutions.m, which allows...
one to automatically compute exact solutions expressible in hyperbolic or elliptic functions; full details are presented in \cite{BGH03}. In Section 4 we give an algorithm for computing and testing recursion operators \cite{HG99}; the package PDERecursionOperator.m automates the steps. The latter package builds on the code InvariantsSymmetries.m \cite{GH97b}, which computes conserved densities, fluxes, and generalized symmetries.

In this paper we consider systems of $M$ polynomial differential equations,

\begin{equation}
\Delta(u(x), u'(x), u''(x), \cdots, u^{(m)}(x)) = 0,
\end{equation}

where the dependent variable $u$ has $M$ components $u_i$, the independent variable $x$ has $N$ components $x_j$, and $u^{(m)}(x)$ denotes the collection of mixed derivative terms of order $m$. We assume that any arbitrary coefficients parameterizing the system are strictly positive and denoted as lower-case Greek letters.

Two carefully selected examples will illustrate the algorithms. The first example is the Kaup-Kupershmidt (KK) equation (see e.g. \cite{GH97a, W98}),

\begin{equation}
\frac{u_t}{5} = 5u^2 u_x + 25u^2 u_{2x} + 5uu_x^3 + uu_x^5.
\end{equation}

The second example is the Hirota-Satsuma system of coupled Korteweg-de Vries (KdV) equations (see e.g. \cite{AC91}),

\begin{equation}
\begin{aligned}
\frac{u_t}{\alpha} &= \alpha(6uu_x + u_{3x}) - 2vv_x, \quad \alpha > 0, \\
\frac{v_t}{\alpha} &= -3uv_x - v_{3x}.
\end{aligned}
\end{equation}

The computations for both examples will be done by the software.

2. The Painlevé Test

The Painlevé test verifies whether a system of ODEs or PDEs satisfies the necessary conditions for having the Painlevé property. There is some variation in what is meant by the Painlevé property \cite{KJH97}. As defined in \cite{ARS80}, for a PDE to have the Painlevé property, all ODE reductions of the PDE must have the Painlevé property. While \cite{ARS80} requires that all movable singularities of all solutions are poles, the more general definition used by Painlevé himself requires that all solutions of the ODE are single-valued around all movable singularities. A later version \cite{WTC83} allows testing of PDEs directly without having to reduce them to ODEs. For a thorough discussion of the Painlevé property, see \cite{C00, C99}.

**Definition 2.1.** A PDE has the Painlevé property if its solutions in the complex plane are single-valued in the neighborhood of all its movable singularities.

**2.1. Algorithm and implementation.** Following \cite{WTC83}, we assume a Laurent expansion for the solution

\begin{equation}
\begin{aligned}
&u_i(x) = g^{\alpha_i}(x) \sum_{k=0}^{\infty} u_{i,k}(x)g^k(x), \quad u_{i,0}(x) \neq 0 \quad \text{and} \quad \alpha_i \in \mathbb{Z}^-,
\end{aligned}
\end{equation}

where $u_{i,k}(x)$ is an analytic function in the neighborhood of $g(x)$. The solution should be single-valued in the neighborhood of the non-characteristic, movable singular manifold $g(x)$, which can be viewed as the surface of the movable poles in the complex plane.

The algorithm for the Painlevé test is composed of the following three steps:
Step 1 (Determine the dominant behavior). It suffices to substitute
(2.2)  \( u_i(x) = u_i,0(x)g^{\alpha_i}(x) \)
into (1.1) to determine the strictly negative integer \( \alpha_i \) and the function \( u_i,0(x) \). In the resulting polynomial system, equating every two possible lowest exponents of \( g(x) \) in each equation gives a linear system to determine \( \alpha_i \). The linear system is then solved for \( \alpha_i \).

If one or more exponents \( \alpha_i \) remain undetermined, we assign a strictly negative integer value to the free \( \alpha_i \) so that every equation in (1.1) has at least two different terms with equal lowest exponents. Once \( \alpha_i \) is known, we substitute (2.2) back into (1.1) and solve for \( u_i,0(x) \).

Step 2 (Determine the resonances). For each \( \alpha_i \) and \( u_i,0(x) \), we calculate the integers \( r \) for which \( u_i,r(x) \) is an arbitrary function in (2.1). We substitute
(2.3)  \( u_i(x) = u_i,0(x)g^{\alpha_i}(x) + u_i,r(x)g^{\alpha_i+r}(x) \)
into (1.1), keeping only the most singular terms in \( g(x) \), and require that the coefficients of \( u_i,r(x) \) equate to zero. This is done by computing the roots of \( \det Q = 0 \), where the \( M \times M \) matrix \( Q \) satisfies
(2.4)  \[ Q \cdot u_r = 0, \quad u_r = (u_{1,r} \ u_{2,r} \ldots \ u_{M,r})^T. \]

Step 3 (Find the constants of integration and check compatibility conditions). For the system to possess the Painlevé property, the arbitrariness of \( u_i,r(x) \) must be verified up to the highest resonance level. That is, all compatibility conditions must be trivially satisfied. This is done by substituting
(2.5)  \[ u_i(x) = g^{\alpha_i}(x) \sum_{k=0}^{r_{\text{Max}}} u_{i,k}(x)g^k(x) \]
into (1.1), where \( r_{\text{Max}} \) is the highest positive integer resonance. For the system to have the Painlevé property, there must be as many arbitrary constants of integration at resonance levels as resonances at that level. Furthermore, all constants of integration \( u_{i,k}(x) \) at non-resonance levels must be unambiguously determined.

### 2.2. Examples of the Painlevé test.

**Example 2.1 (Kaup-Kupershmidt).** To determine the dominant behavior, we substitute (2.2) into (1.2) and pull off the exponents of \( g(x) \) (see Table 1). Removing duplicates and non-dominant exponents, we are left with
(2.6)  \( \{ \alpha_1 - 5, 2\alpha_1 - 3, 3\alpha_1 - 1 \} \).
Considering all possible balances of two or more exponents leads to \( \alpha_1 = -2 \).
Substituting \( u(x) = u_{1,0}(x)g^{-2}(x) \) into (1.2) and solving for \( u_{1,0}(x) \) gives us (2.7)

\[
u_{1,0}(x) = -24g_x^2(x) \quad \text{and} \quad u_{1,0}(x) = -3g_x^2(x).
\]

For the first branch, substituting \( u(x) = -24g_x^2(x)g^{-2}(x) + u_{1,1}(x)g^{-2}(x) \) into (1.2), keeping the most singular terms, and taking the coefficient of \( u_{1,1}(x) \), gives (2.8)

\[ -2(r + 7)(r + 1)(r - 6)(r - 10)(r - 12)g_x^5(x) = 0. \]

Hence, \( r = -7, -1, 6, 10, 12. \)

While we are only concerned with the positive resonances, \( r = -1 \) is often called the universal resonance and corresponds to the arbitrariness of the manifold \( g(x) \). The meaning of other negative integer resonances is not fully understood [KJH97].

The constants of integration at level \( j \) are found by substituting (2.5) into (1.2), where \( r_{\text{Max}} = 12 \), and pulling off the coefficients of \( g^j(x) \). The first few are (2.9)

\[ \text{at} \ j = 1: \quad u_{1,1}(x) = 24g_{2x}(x), \]

(2.10) \[ \text{at} \ j = 2: \quad u_{1,2}(x) = \frac{6g_{2x}^2(x) - 8g_x(x)g_{3x}(x)}{g_x^2(x)}, \]

(2.11) \[ \text{at} \ j = 3: \quad u_{1,3}(x) = \frac{6g_{2x}^3(x) - 8g_x(x)g_{2x}(x)g_{3x}(x) + 2g_x^2(x)g_{4x}(x)}{g_x^2(x)}. \]

The compatibility conditions are satisfied at resonance levels 6, 10, and 12. The remaining constants of integration \( u_{1,j}(x) \) are computed but not shown here.

Likewise, in the second branch, substitute \( u(x) = -3g_x^2(x)g^{-2}(x) + u_{1,1}(x)g^{-2}(x) \) into (1.2), and proceed as before to get \( r = -1, 3, 5, 6, 7 \). The constants of integration at levels \( j = 1, 2 \) and 4 are again found by substituting (2.5) into (1.2) and pulling off the coefficients of \( g^j(x) \). This gives, (2.12)

\[ \text{at} \ j = 1: \quad u_{1,1}(x) = 3g_{2x}(x), \]

(2.13) \[ \text{at} \ j = 2: \quad u_{1,2}(x) = \frac{3g_{2x}^2(x) - 4g_x(x)g_{3x}(x)}{4g_x^2(x)}. \]

The coefficient \( u_{1,4} \) at level \( j = 4 \) is not shown here due to length. At the resonance levels \( r = 3, 5, 6, 7 \), the compatibility conditions are satisfied and (1.2) passes the Painlevé test. It is well-known (see e.g. [GH97a, W98]) that (1.2) is completely integrable.

**Example 2.2 (Hirota-Satsuma).** The Hirota-Satsuma system illustrates the subtleties of determining the dominant behavior.

As in Example 2.1, we substitute (2.2) into (1.3) and pull off the exponents of \( g(x) \) (listed in Table 2). Removing non-dominant exponents and duplicates by

| Term | Exponents of \( g(x) \) |
|------|------------------------|
| \( u_t \) | \( \alpha_1 - 1 \) |
| \(-6\alpha_{3x} \) | \( \alpha_1 - 3, \alpha_1 - 3, \alpha_1 - 3, \alpha_1 - 2, \alpha_1 - 2, \alpha_1 - 1 \) |
| \(-6\alpha_{uu_x} \) | \( 2\alpha_1 - 1 \) |
| \( 2\alpha_{v_x} \) | \( 2\alpha_2 - 1 \) |
| \( v_t \) | \( \alpha_2 - 1 \) |
| \( 3\alpha_{v_x} \) | \( \alpha_2 - 3, \alpha_2 - 3, \alpha_2 - 3, \alpha_2 - 2, \alpha_2 - 2, \alpha_2 - 1 \) |
| \( 3\alpha_{uv} \) | \( \alpha_1 + \alpha_2 - 1 \) |

**Table 2.** The exponents of \( g(x) \) for (1.3).
Similarly, we computed all constants of integration, but ran into compatibility
Hence, $\alpha(2.14) 2\alpha_1 - 1 \leq 2\alpha_2 - 1$ or $-2 \leq \alpha_2 < 0$.

Hence, $\alpha_2 = -1$ or $\alpha_2 = -2$.

Using the two solutions for $\alpha_i$, solving for $u_{i,0}$ results in

$$
\begin{align*}
\alpha_1 &= -2, \quad u_{1,0}(x) = -4g_2^2(x), \\
\alpha_2 &= -2, \quad u_{2,0}(x) = \pm 2\sqrt{6}g_2^2(x),
\end{align*}
$$

$$
\begin{align*}
\alpha_1 &= -2, \quad u_{1,0}(x) = -2g_2^2(x), \\
\alpha_2 &= -1, \quad u_{2,0}(x) \text{ arbitrary}.
\end{align*}
$$

We substitute (2.5) into (1.3) and pulling off the coefficients of $\text{g}^r(x)$ and $v(x) = \pm 2\sqrt{6}g_2^2(x)g^{-2}(x)$ + $u_{2,r}(x)g^{-2}(x)$ into (1.3) and keeping the most singular terms gives

$$
Q \cdot \mathbf{u}_r = \left( -\frac{(r-4)(r^2-5r-18)\alpha g_2^j(x)}{\mp 4(r-4)\sqrt{6}g_2^3(x)} \pm 12\sqrt{6}g_2^2(x) \pm 2(r-2)(r-7)g_2^3(x) \right) \begin{pmatrix} u_{1,r}(x) \\ u_{2,r}(x) \end{pmatrix} = 0.
$$

Setting

$$
det Q = -\alpha(r+2)(r+1)(r-3)(r-4)(r-6)(r-8)g_6^6(x) = 0
$$
yields $r = -2, -1, 1, 3, 4, 6, 8$.

As in the previous example, the constants of integration at level $j$ are found by substituting (2.5) into (1.3) and pulling off the coefficients of $g^j(x)$. At $j = 1$,

$$
\begin{pmatrix}
-66\alpha g_2^2(x) & \mp 12\sqrt{6}g_2^3(x) \\
\mp 12\sqrt{6}g_2^3(x) & 6g_2^3(x)
\end{pmatrix}
\begin{pmatrix}
\frac{u_{1,1}(x)}{g_2^2(x)} \\
\frac{u_{2,1}(x)}{g_2^3(x)}
\end{pmatrix}
= \begin{pmatrix}
-120\alpha g_2^3(x)g_2(x) \\
\mp 60\sqrt{6}g_2^2(x)g_2(x)
\end{pmatrix}.
$$

Thus,

$$
u_{1,1}(x) = 4g_2^2(x), \quad u_{2,1}(x) = \pm 2\sqrt{6}g_2^3(x).
$$

At $j = 2$,

$$
u_{1,2}(x) = \frac{3g_2^2(x) - g_2(x)(g_2(x) + 4g_4(x))}{3g_2^3(x)},
$$

$$
u_{2,2}(x) = \pm \frac{3g_2^2(x) - 4g_2(x)(g_4(x) - 1 + 2\alpha)g_2(x)}{\sqrt{6}g_2^3(x)}.
$$

The remaining constants of integration are omitted due to length. The compatibility conditions at $r = 3$ and 4 are satisfied. At $r = 6$ and $r = 8$, the compatibility conditions require $\alpha = \frac{1}{6}$.

Likewise, for (2.10), the resonances following from the substitution of $u(x) = -2g_2^2(x)g^{-2}(x) + u_{1,r}(x)g^{-2}(x)$ and $v(x) = u_{2,0}(x)g^{-2}(x) + u_{2,r}(x)g^{-1}(x)$ into (1.3) are $r = -1, 0, 1, 4, 5, 6$. The zero resonance explains the arbitrariness of $u_{2,0}(x)$. Similarly, we computed all constants of integration, but ran into compatibility conditions at $r = 5$ and $r = 6$, which require $\alpha = \frac{1}{5}$. Therefore, (1.3) passes the Painlevé test if $\alpha = \frac{1}{5}$, a fact confirmed by other analyses of integrability [AC91].
3. Travelling Wave Solutions in Hyperbolic or Elliptic Functions

The traveling wave solutions of many nonlinear ODEs and PDEs from soliton theory (and elsewhere) can be expressed as polynomials of hyperbolic or elliptic functions. For instance, the bell shaped sech-solutions and kink shaped tanh-solutions model wave phenomena in fluid dynamics, plasmas, elastic media, electrical circuits, optical fibers, chemical reactions, bio-genetics, etc. An explanation is given in [HT90], while a multitude of references to tanh-based techniques and applications can be found in [BGH03, MH96].

In this section we discuss the tanh-, sech-, cn- and sn-methods as they apply to nonlinear polynomial systems of ODEs and PDEs in multi-dimensions.

3.1. Algorithm and implementation. All four flavors of the algorithm share the same five basic steps.

**Step 1** (Transform the PDE into a nonlinear ODE). We seek solutions in the traveling frame of reference,

\[ \xi = \sum_{j=1}^{N} c_j x_j + \delta, \]

where the components \( c_j \) of the wave vector and the phase \( \delta \) are constants.

In the tanh method, we seek polynomial solutions expressible in hyperbolic tangent, \( T = \tanh \xi \).

\[ \tanh' \xi = \operatorname{sech}^2 \xi \]

\[ \tanh'' \xi = -2 \tanh \xi + 2 \tanh^3 \xi, \]

Therefore, the first and all higher-order derivatives are polynomial in \( T \). Consequently, repeatedly applying the chain rule,

\[ T = \tanh(\xi) : \frac{\partial \bullet}{\partial x_j} = \frac{d \bullet}{dT} \frac{dT}{d\xi} \frac{\partial \xi}{\partial x_j} = c_j (1 - T^2) \frac{d \bullet}{dT} \]

transforms \( \frac{\partial \bullet}{\partial x_j} \) into a coupled system of nonlinear ODEs.

Similarly, using the identity \( \tanh^2 \xi + \operatorname{sech}^2 \xi = 1 \), we get

\[ \operatorname{sech} \xi = -\operatorname{sech} \tanh \xi = -\tanh \xi \sqrt{1 - \operatorname{sech}^2 \xi}. \]

Likewise, for Jacobi's elliptic functions with modulus \( m \), we use the identities

\[ \operatorname{sn}^2(\xi; m) = 1 - \operatorname{cn}^2(\xi; m), \quad \text{and} \quad \operatorname{dn}^2(\xi; m) = 1 - m + m \operatorname{cn}^2(\xi; m), \]

to write, for example, \( \operatorname{cn}' \) in terms of \( \operatorname{cn} \) :

\[ \operatorname{cn}'(\xi; m) = -\operatorname{sn}(\xi; m) \operatorname{dn}(\xi; m) = -\sqrt{(1 - \operatorname{cn}^2(\xi; m))(1 - m + m \operatorname{cn}^2(\xi; m))}. \]

Then, repeatedly applying the chain rules,

\[ S = \operatorname{sech}(\xi) : \frac{\partial \bullet}{\partial x_j} = -c_j S \sqrt{1 - S^2} \frac{d \bullet}{dS}, \]

\[ \operatorname{CN} = \operatorname{cn}(\xi; m) : \frac{\partial \bullet}{\partial x_j} = -c_j \sqrt{(1 - \operatorname{CN}^2)(1 - m + m \operatorname{CN}^2)}, \]

\[ \operatorname{SN} = \operatorname{sn}(\xi; m) : \frac{\partial \bullet}{\partial x_j} = c_j \sqrt{(1 - \operatorname{SN}^2)(1 - m \operatorname{SN}^2)} \frac{d \bullet}{d\operatorname{SN}}. \]
we transform (1.1) into a coupled system of nonlinear ODEs of the form
\[(3.11)\]
\[\Gamma(F, u(F), u'(F), \ldots) + \sqrt{R(F)} \Pi(F, u(F), u'(F), \ldots) = 0,\]
where \(F\) is either \(T, S, \text{CN},\) or \(\text{SN}\), and \(R(F)\) is defined in Table 3.

| \(F\) | \(R(F)\) |
|------|--------|
| \(T\) | 0      |
| \(S\) | \(1 - S^2\) |
| \(\text{CN}\) | \((1 - \text{CN}^2)(1 - m + m \text{CN}^2)\) |
| \(\text{SN}\) | \((1 - \text{SN}^2)(1 - m \text{SN}^2)\) |

Table 3. Values for \(R(F)\) in \((3.11)\).

**Step 2** (Determine the degree of the polynomial solutions). Since we seek polynomial solutions
\[(3.12)\]
\[U_i(F) = \sum_{j=0}^{M_i} a_{ij} F^j,\]
the leading exponents \(M_i\) must be determined before the \(a_{ij}\) can be computed. The process for determining \(M_i\) is quite similar to the one for finding \(\alpha_i\) in Section 2.

Substituting \(U_i(F)\) into \((3.11)\), the coefficients of every power of \(F\) in every equation must vanish. In particular, the highest degree terms must vanish. Since the highest degree terms only depend on \(F^{M_i}\) in \((3.12)\), it suffices to substitute \(U_i(F) = F^{M_i}\) into \((3.11)\). Doing so, we get
\[(3.13)\]
\[P(F) + \sqrt{R(F)} Q(F) = 0,\]
where \(P\) and \(Q\) are polynomials in \(F\). Equating every two possible highest exponents in each \(P_i\) and \(Q_i\) gives a linear system to determine \(M_i\). The linear system is then solved for \(M_i\).

If one or more exponents \(M_i\) remain undetermined, we assign a strictly positive integer value to the free \(M_i\), so that every equation in \((3.12)\) has at least two different terms with equal highest exponents in \(F\).

**Step 3** (Derive the algebraic system for the coefficients \(a_{ij}\)). To generate the system for the unknown coefficients \(a_{ij}\) and wave parameters \(c_j\), substitute \((3.12)\) into \((3.11)\) and set the coefficients of \(F^j\) to zero. The resulting nonlinear algebraic system for the unknown \(a_{ij}\) is parameterized by the wave parameters \(c_j\) and the parameters in \((1.1)\), if any.

**Step 4** (Solve the nonlinear parameterized algebraic system). The most difficult aspect of the method is solving the nonlinear algebraic system. To solve the system we designed a customized, yet powerful, nonlinear solver.

The nonlinear algebraic system is solved with the following assumptions:

1. all parameters (the lower case Greek letters) in \((1.1)\) are strictly positive. (Vanishing parameters may change the exponents \(M_i\) in Step 2). To compute solutions corresponding to negative parameters, reverse the signs of the parameters in \((1.1)\).
2. the coefficients of the highest power terms \((a_{iM_i}, i = 1, \ldots, M_i)\) in \((3.12)\) are all nonzero (for consistency with Step 2).
3. all \(c_j\) are nonzero (demanded by the physical nature of the solutions).
STEP 5 (Build and test solutions). Substitute the solutions from Step 4 into \( (3.12) \) and reverse Step 1 to obtain the explicit solutions in the original variables. It is prudent to test the solutions by substituting them into \( (1.1) \).

3.2. Examples of travelling wave solutions.

EXAMPLE 3.1 (Kaup-Kupershmidt). While the tanh-, sech-, cn- and sn-methods find solutions for \( (1.2) \), we demonstrate the steps of the algorithm using the tanh-method. After which, we summarize the results for the other methods.

First, transform \( (1.2) \) into a nonlinear ODE by repeatedly applying chain rule \( (3.1) \). The resulting ODE is

\[
2c_2 U' + c_1 \left[ 10U^2U' + c_1^2 [25(T^2 - 2)U'' + (T^2 - 1)U'''] \right] + 10U[(6T^2 - 2)U' + 6T(T^2 - 1)U''' + (T^2 - 1)^2 U'''] + c_1^2 (16(15T^4 - 15T^2 + 2)U' + 40(T^2 - 1)(6T(2T^2 - 1)U'') + (6T^4 - 7T^2 + 1)U''' + T(T^2 - 1)^2 U^{(4)}) + (T^2 - 1)^4 U^{(5)})] = 0,
\]

where \( T = \tanh(\xi) \) and \( U = U(T) \).

Next, to compute the degree of the polynomial solution(s), substitute \( U(T) = T^{M_1} \) into \( (3.13) \) and pull off the exponents of \( T \) (see Table 4). Remove duplicates and non-dominant exponents, to get

\[
\{3M_1 - 1, 2M_1 + 1, M_1 + 3\}.
\]

Consider all possible balances of two or more exponents to find \( M_1 = 2 \).

Substitute

\[
U(T) = a_{10} + a_{11}T + a_{12}T^2
\]

into \( (3.14) \) and equate the coefficients of \( T^i \) to zero (where \( i = 0, 1, \ldots, 5 \)) to get

\[
(a_{12} + 3c_1^2)(a_{12} + 24c_1^2) = 0,
\]

\[
a_{11}(5a_{12}^2 + 55a_{12}c_1^2 + 24c_1^4) = 0,
\]

\[
a_{11}(5a_{10}c_1 - 10a_{10}c_1^3 + 25a_{12}c_1^3 + 16c_1^5 + c_2) = 0,
\]

\[
(a_{11}(a_{11} + 6a_{10}a_{12} + 6a_{10}c_1^2 - 48a_{12}c_1^2 - 24c_1^4) = 0,
\]

\[
4a_{12}a_{11} + 4a_{10}a_{12} + 11a_{11}c_1^2 + 24a_{10}a_{12}c_1^2 - 56a_{12}c_1^2 - 192a_{12}c_1^4 = 0.
\]

\[
10a_{10}a_{11}c_1 + 10a_{10}a_{12}c_1 - 35a_{12}c_1^3 - 80a_{10}a_{12}c_1^3 + 50a_{12}c_1^3 + 272a_{12}c_1^5 + 2a_{12}c_2 = 0.
\]
Solve the nonlinear algebraic system with the assumption that \(a_{12}, c_1, \) and \(c_2\) are all nonzero. Two solutions are obtained:

\[
\begin{align*}
\begin{cases}
a_{10} &= 16c_1^2, & a_{11} &= 0, \\
a_{12} &= -24c_1^2, & c_2 &= -176c_1^5,
\end{cases}
\quad\text{and}\quad
\begin{cases}
a_{10} &= 2c_1^2, & a_{11} &= 0, \\
a_{12} &= -3c_1^2, & c_2 &= -c_1^5,
\end{cases}
\end{align*}
\]

where \(c_1\) is arbitrary.

Substitute the solutions into (3.16) and return to \(u(x, t)\) to get

\[
\begin{align*}
u(x, t) &= 16c_1^2 - 24c_1^2 \tanh^2(c_1 x - 176c_1^5 t + \delta), \\
u(x, t) &= 2c_1^2 - 3c_1^2 \tanh^2(c_1 x - c_1^5 t + \delta).
\end{align*}
\]

Using the sech-method, one finds

\[
\begin{align*}
u(x, t) &= -8c_1^2 + 24c_1^2 \sech^2(c_1 x - 176c_1^5 t + \delta), \\
u(x, t) &= -c_1^2 + 3c_1^2 \sech^2(c_1 x - c_1^5 t + \delta).
\end{align*}
\]

Alternatively, the latter solutions can be found directly from the tanh-method solutions by using the identity \(\tanh^2 \xi = 1 - \sech^2 \xi\).

For the \(cn\) - and \(sn\)-methods, one gets

\[
\begin{align*}
u(x, t) &= 8c_1^2 \left[1 - 2m + 3m \cosech^2(c_1 x - 176c_1^5 (m^2 - m + 1)t + \delta; m)\right], \\
u(x, t) &= c_1^2 \left[1 - 2m + 3m \cosech^2(c_1 x - c_1^5 (m^2 - m + 1)t + \delta; m)\right], \\
u(x, t) &= 8c_1^2 \left[1 + m - 3m \sn^2(c_1 x - 176c_1^5 (m^2 - m + 1)t + \delta; m)\right], \\
u(x, t) &= c_1^2 \left[1 + m - 3m \sn^2(c_1 x - c_1^5 (m^2 - m + 1)t + \delta; m)\right].
\end{align*}
\]

Example 3.2 (Hirota-Satsuma). As in the previous example, the tanh-, sech-, \(cn\)- and \(sn\)-methods all find solutions for (1.3). In this example, however, we will illustrate the steps using the sech-method.

Transform (1.3) into a coupled system of ODEs, apply chain rule (3.8) and cancel the common \(S\sqrt{1 - S^2}\) factors to get

\[
\begin{align*}
c_2U_1'' - 6\alpha c_1 U_1 U_1' - \alpha c_1^3 [(1 - 6S^2)U_1' + 3S(1 - 2S^2)U_1'' + S^2(1 - 2S^2)U_1''' + 2c_1 U_2 U_2'] = 0, \\
c_2U_2'' + 3c_1 U_1 U_2' + c_1^3 [(1 - 6S^2)U_2' + 3S(1 - 2S^2)U_2'' + S^2(1 - 2S^2)U_2'''] = 0.
\end{align*}
\]

To find the degree of the polynomials, substitute \(U_1(S) = SM_1, U_2(S) = SM_2\) into (3.27) and first equate the highest exponents from \(\Delta_2\) to get

\[
M_1 + M_2 - 1 = 1 + M_2 \quad \text{or} \quad M_1 = 2.
\]

The maximal exponents coming from \(\Delta_1\) are \(2M_1 - 1\) (from the \(U_1 U_1'\) term), \(M_1 + 1\) (from \(U_1'''\)), and \(2M_2 - 1\) (from \(U_2 U_2'\)).

Since \(M_1 = 2\) balances at least two of the possible dominant exponents in \(\Delta_1\), namely \(2M_1 - 1\) and \(M_1 + 1\), one is again left with \(1 \leq M_2 \leq M_1 = 2\), or

\[
\begin{align*}
M_1 &= 2, & U_1(S) &= a_{10} + a_{11} S + a_{12} S^2, \\
M_2 &= 1, & U_2(S) &= a_{20} + a_{21} S,
\end{align*}
\]

\[
\begin{align*}
M_1 &= 2, & U_1(S) &= a_{10} + a_{11} S + a_{12} S^2, \\
M_2 &= 2, & U_2(S) &= a_{20} + a_{21} S + a_{22} S^2.
\end{align*}
\]
To derive the algebraic system for \(a_{ij}\), substitute (3.29) into (3.27), cancel common numerical factors, and organize the equations (according to complexity):

\[
\begin{align*}
& a_{11}a_{21}c_1 = 0, \\
& \alpha a_{11}c_1(3a_{12} - c_1^2) = 0, \\
& \alpha a_{12}c_1(a_{12} - 2c_1^2) = 0, \\
& a_{21}c_1(a_{12} - 2c_1^2) = 0, \\
& a_{21}(3a_{10}c_1 + c_1^3 + c_2) = 0, \\
& 6\alpha a_{10}a_{11}c_1 - 2a_{20}a_{21}c_1 + \alpha a_{11}c_1^3 - a_{12}c_2 = 0, \\
& 3\alpha a_{11}c_1 + 6\alpha a_{10}a_{12}c_1 - a_{21}c_1 + 4a_{12}c_1^3 - a_{12}c_2 = 0.
\end{align*}
\]

Similarly, after substitution of (3.30) into (3.27), one gets

\[
\begin{align*}
& a_{22}c_1(a_{12} - 4c_1^2) = 0, \\
& a_{21}(3a_{10}c_1 + c_1^3 + c_2) = 0, \\
& c_1(a_{12}a_{21} + 2a_{11}a_{22} - 2a_{21}c_1^2) = 0, \\
& c_1(3\alpha a_{11}a_{12} - a_{21}a_{22} - \alpha a_{11}c_1^2) = 0, \\
& c_1(3\alpha a_{12}^2 - a_2^2 - 6\alpha a_{12}c_1^2) = 0, \\
& 6\alpha a_{10}a_{11}c_1 - 2a_{20}a_{21}c_1 + \alpha a_{11}c_1^3 - a_{11}c_2 = 0, \\
& 3a_{11}a_{21}c_1 + 6a_{10}a_{22}c_1 + 8a_{22}c_1^3 + 2a_{22}c_2 = 0, \\
& 3\alpha a_{11}c_1 + 6\alpha a_{10}a_{12}c_1 - a_{21}c_1 - 2a_{20}a_{22}c_1 + 4a_{12}c_1^3 - a_{12}c_2 = 0.
\end{align*}
\]

Since \(a_{12}, a_{21}, \alpha, c_1, \) and \(c_2,\) are nonzero, the solution of (3.31) is

\[
\begin{align*}
& a_{10} = -(c_1^3 + c_2^3)/(3c_1), \quad a_{20} = 0, \\
& a_{11} = 0, \quad a_{21} = \pm\sqrt{4\alpha c_1^4 - 2(1 + 2\alpha)c_1c_2}, \\
& a_{12} = 2c_1^2.
\end{align*}
\]

For \(a_{12}, a_{22}, \alpha, c_1, \) and \(c_2\) nonzero, the solution of (3.32) is

\[
\begin{align*}
& a_{10} = -(4c_1^3 + c_2^3)/(3c_1), \quad a_{20} = \pm[4\alpha c_1^3 + (1 + 2\alpha)c_2]/(c_1\sqrt{6\alpha}), \\
& a_{11} = 0, \quad a_{21} = 0, \\
& a_{12} = 4c_1^2, \quad a_{22} = \mp2c_1^2\sqrt{6\alpha}.
\end{align*}
\]

The solutions of (1.3) involving sech are then

\[
\begin{align*}
& u(x, t) = -\frac{c_1^3 + c_2}{3c_1} + 2c_1^2 \operatorname{sech}^2(c_1 x + c_2 t + \delta), \\
& v(x, t) = \pm\sqrt{4\alpha c_1^4 - 2(1 + 2\alpha)c_1c_2 \operatorname{sech}(c_1 x + c_2 t + \delta)},
\end{align*}
\]

and

\[
\begin{align*}
& u(x, t) = -\frac{4c_1^3 + c_2}{3c_1} + 4c_1^2 \operatorname{sech}^2(c_1 x + c_2 t + \delta), \\
& v(x, t) = \pm\frac{4\alpha c_1^3 + (1 + 2\alpha)c_2}{c_1\sqrt{6\alpha}} \mp 2c_1^2\sqrt{6\alpha} \operatorname{sech}^2(c_1 x + c_2 t + \delta).
\end{align*}
\]

In both sets of solutions, \(c_1, c_2, \alpha, \) and \(\delta\) are arbitrary.
4. Recursion Operators

In this section, we only consider polynomial system of evolution equations in (1 + 1) dimensions,
\begin{equation}
\mathbf{u}_t(x, t) = \mathbf{F}(\mathbf{u}(x, t), \mathbf{u}_x(x, t), \mathbf{u}_{2x}(x, t), \ldots, \mathbf{u}_{mx}(x, t)),
\end{equation}
where \( \mathbf{u} \) has \( M \) components \( u_i \) and \( \mathbf{u}_{mx} = \partial^m \mathbf{u} / \partial x^m \).
For brevity, we write \( \mathbf{F}(\mathbf{u}) \), although \( \mathbf{F} \) typically depends on \( \mathbf{u} \) and its \( x \)-derivatives up to order \( m \).

If present, any parameters in the system are strictly positive and denoted as lower-case Greek letters.

The algorithm in Section 4.2 will use the concepts of dilation invariance, densities, and symmetries.

4.1. Scaling invariance, densities, symmetries. A PDE is dilation invariant if it is invariant under a dilation symmetry.

**Example 4.1 (Kaup-Kupershmidt).** As an example, \( \mathbf{u}_t \equiv \mathbf{u} \) is invariant under the dilation (scaling) symmetry
\begin{equation}
(t, x, u) \rightarrow (\lambda^{-5} t, \lambda^{-1} x, \lambda^2 u),
\end{equation}
where \( \lambda \) is an arbitrary parameter, leaving \( \lambda^7 \) as a common factor upon scaling.

To find the dilation symmetry, set the weight of the \( x \)-derivative to one, \( w(D_x) = 1 \), and require that all terms in \( \mathbf{F}(\mathbf{u}) \) have the same weight. For \( \mathbf{u}_t \), we have
\begin{equation}
w(u) + w(D_t) = 3w(u) + 1 = 2w(u) + 3 = w(u) + 5,
\end{equation}
or \( w(u) = 2 \) and \( w(D_t) = 5 \). Consequently, in \( \mathbf{u}_t \) the sum of the weights or rank of each term is 7.

A recursion operator, \( \mathcal{R} \), is a linear integro-differential operator which links generalized symmetries \[ \mathcal{O}_{93} \]
\begin{equation}
\mathcal{G}^{(j+s)} = \mathcal{R}\mathcal{G}^{(j)}, \quad j \in \mathbb{N},
\end{equation}
where \( s \) is the seed (\( s = 1 \) in most, but not all cases) and \( \mathcal{G}^{(j)} \) is the \( j \)-th symmetry.

A generalized symmetry, \( \mathcal{G}(\mathbf{u}) \), leaves \( \mathbf{u}_t \equiv \mathbf{u} \) invariant under the replacement \( \mathbf{u} \rightarrow \mathbf{u} + \epsilon \mathcal{G} \) within order \( \epsilon \). Hence, \( \mathcal{G} \) must satisfy the linearized equation \[ \mathcal{O}_{93} \]
\begin{equation}
D_t \mathcal{G} = \mathbf{F}'(\mathbf{u})[\mathcal{G}],
\end{equation}
on solutions of \( \mathbf{u}_t \). \( \mathbf{F}'(\mathbf{u})[\mathcal{G}] \) is the Fréchet derivative of \( \mathbf{F} \) in the direction of \( \mathcal{G} \).

For details about the computation of generalized symmetries, see \[ \mathcal{G}_{97b}, \mathcal{G}_{99} \].

A conservation law \[ \mathcal{O}_{93} \]
\begin{equation}
D_t \rho(x, t) + D_x J(x, t) = 0,
\end{equation}
valid for solutions of \( \mathbf{u}_t \equiv \mathbf{u} \), links a conserved density \( \rho(x, t) \) with the associated flux \( J(x, t) \). For details about the computation of conservation laws, see \[ \mathcal{G}_{97a}, \mathcal{G}_{97b} \].

If \( \mathbf{u}_t \) is scaling invariant, then its conserved densities, fluxes, generalized symmetries, and recursion operators are also dilation invariant. One could say they ‘inherit’ the scaling symmetry of the original PDE. The existence of an infinite number of symmetries or an infinite number of conservation laws assures complete integrability \[ \mathcal{O}_{93} \] Once the first few densities and symmetries are computed, a recursion operator can be constructed with the following algorithm.
4.2. Algorithm and implementation.

STEP 1 (Determine the rank of the recursion operator). The rank of the recursion operator is determined by the difference in ranks of the generalized symmetries it links,

\[ \text{rank} \mathcal{R}_{ij} = \text{rank} G_i^{(k+s)} - \text{rank} G_j^{(k)}, \]

where \( \mathcal{R} \) is an \( M \times M \) matrix and \( G \) has \( M \) components.

STEP 2 (Determine the form of the recursion operator). The recursion operator naturally splits into two pieces [HG99],

\[ \mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1, \]

where \( \mathcal{R}_0 \) is a differential operator and \( \mathcal{R}_1 \) is an integral operator. The differential operator, \( \mathcal{R}_0 \), is a linear combination (with constant coefficients) of terms of type \( D^i_x u^j \) \((i, j \in \mathbb{Z}^+)\), which must be of the correct rank. To standardize the form of \( \mathcal{R}_0 \), propagate \( D_x \) to the right, for example, \( D_2^2 u = u_{2x} I + u_x D_x + u D_x^2 \).

The integral operator, \( \mathcal{R}_1 \), is composed of the terms

\[ \sum_i \sum_j \beta_{ij} G_i^{(j)} D_x^{-1} \otimes \rho^{(j)'} \]

of the correct rank, where \( \otimes \) is the matrix outer product and \( \rho^{(j)'} \) is the covariant (Fréchet derivative of \( \rho^{(j)} \)). To standardize the form of \( \mathcal{R}_1 \), propagate \( D_x \) to the left, for example, \( D_x^{-1} u_x D_x = u_x I - D_x^{-1} u_{2x} I \).

STEP 3 (Determine the coefficients). To determine the coefficients in the form of the recursion operator, we substitute \( \mathcal{R} \) into the defining equation [HG99, W98],

\[ \frac{\partial \mathcal{R}}{\partial t} + \mathcal{R}' [F] + \mathcal{R} \circ F'(u) - F'(u) \circ \mathcal{R} = 0, \]

where \( \circ \) denotes a composition of operators, \( \mathcal{R}' [F] \) is the Fréchet derivative of \( \mathcal{R} \) in the direction of \( F \), and \( F'(u) \) is the Fréchet derivative with entries

\[ F'_{ij}(u) = \sum_{k=0}^{m} \left( \frac{\partial F_i}{\partial (u_j)_{kx}} \right) D_x^k. \]

4.3. Examples of scalar and matrix recursion operators.

EXAMPLE 4.2 (Kaup-Kupershmidt). Using the weights \( w(u) = 2, w(D_x) = 1 \), and \( w(D_t) = 5 \), we find

\[ \text{rank} G^{(1)} = 3, \quad \text{rank} G^{(2)} = 7, \quad \text{rank} G^{(3)} = 9, \]
\[ \text{rank} G^{(4)} = 13, \quad \text{rank} G^{(5)} = 15, \quad \text{rank} G^{(6)} = 19. \]

We guess that \( \text{rank} \mathcal{R} = 6 \) and \( s = 2 \), since \( \text{rank} G^{(2)} = \text{rank} G^{(1)} \neq \text{rank} G^{(3)} = \text{rank} G^{(2)} \) but \( \text{rank} G^{(3)} = \text{rank} G^{(1)} = \text{rank} G^{(4)} \neq \text{rank} G^{(2)} = 6 \).

Thus, taking all \( D^i_x u^j \) \((i, j \in \mathbb{Z}^+)\) such that \( \text{rank} D_x^i u^j = 6 \) gives

\[ \mathcal{R}_0 = c_1 D_6^6 + c_2 u D_4^4 + c_3 u_x D_3^3 + c_4 u^2 D_2^2 + c_5 u_{2x} D_x^2 + c_6 u_x^2 D_x + c_7 x^3 D_x + c_8 u^3 I + c_9 u_x^3 I + c_{10} u_{2x} I + c_{11} u_{4x} I. \]
Using the densities $\rho^{(1)} = u$ and $\rho^{(2)} = 3u^2 - 4v^2$, and the symmetries $G^{(1)} = u_x$, and $G^{(2)} = F(u) = 5u^2u_x + \frac{25}{2}u_xu_{2x} + 5uu_{4x} + u_{5x}$ from (4.12), we compute

$$\mathcal{R}_1 = \tilde{c}_1G^{(1)}D_x^{-1}\rho^{(2)}' + \tilde{c}_2G^{(2)}D_x^{-1}\rho^{(1)}'$$

$$= \tilde{c}_1u_xD_x^{-1}(6u_xD_x - 12u^2I) + \tilde{c}_2G^{(2)}D_x^{-1}$$

$$= c_{12}u_x[D_x^{-1}(u_xI + 2u^2I) - u_xI] + c_{13}G^{(2)}D_x^{-1}. (4.14)$$

Substituting $\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1$ and $G^{(2)} = F$ into (4.10) gives us 49 linear equations for $c_i$. Solving, we find

$$c_1 = \frac{4c_9}{69}, c_2 = \frac{8c_9}{23}, c_3 = \frac{24c_9}{23}, c_4 = \frac{12c_9}{23}, c_5 = \frac{98c_9}{69}, c_6 = \frac{40c_9}{23},$$

$$c_7 = \frac{10c_9}{69}, c_8 = \frac{16c_9}{69}, c_{10} = \frac{82c_9}{69}, c_{11} = \frac{26c_9}{69}, c_{12} = \frac{2c_9}{69}, c_{13} = \frac{4c_9}{69},$$

where $c_9$ is arbitrary. Taking $c_9 = 69/4$, we find the recursion operator in [W98].

$$\mathcal{R} = D_x^2 + 6uD_x^4 + 18u_xD_x^3 + 9u^2D_x^2$$

$$+ \frac{49}{2}u_xD_x^2 + 30uu_xD_x + \frac{35}{2}u_x^2D_x + 4u^3I + \frac{69}{4}u_x^2I$$

$$+ \frac{41}{2}u_x^2u_x + \frac{13}{2}u_x^4I + \frac{1}{2}u_xD_x^{-1}(u_xI + 2u^2I) + G^{(2)}D_x^{-1}. (4.16)$$

**Example 4.3** (Hirota-Satsuma). Only when $\alpha = \frac{1}{2}$ does (1.3) have infinitely many densities and symmetries. The first few are

$$\rho^{(1)} = u, \rho^{(2)} = 3u^2 - 2v^2, \ G^{(1)} = \begin{pmatrix} u_x \\ v_x \end{pmatrix}, \ G^{(2)} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} \alpha(6uu_x + u_{3x} - 2vv_x) \\ -(3uv_x + v_{3x}) \end{pmatrix}. (4.17)$$

We also computed the $G^{(3)}$ and $G^{(4)}$, but they are not shown due to length. Solving the weight equations

$$\begin{cases} w(u) + w(D_t) = 2w(u) + 1 = w(u) + 3 = 2w(v) + 1, \\
w(v) + w(D_t) = w(u) + w(v) + 1 = w(v) + 3, \end{cases} (4.18)$$

yields $w(u) = w(v) = 2$ and $w(D_t) = 3$. Based on these weights, rank $\rho^{(1)} = 2$, rank $\rho^{(2)} = 4$, and

$$\text{rank } G^{(1)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \quad \text{rank } G^{(2)} = \begin{pmatrix} 5 \\ 5 \end{pmatrix},$$

$$\text{rank } G^{(3)} = \begin{pmatrix} 7 \\ 7 \end{pmatrix}, \quad \text{rank } G^{(4)} = \begin{pmatrix} 9 \\ 9 \end{pmatrix}. (4.19)$$

We would first guess that rank $\mathcal{R}_{ij} = 2$ and $s = 1$. If indeed the symmetries were linked consecutively, then

$$\mathcal{R}_0 = \begin{pmatrix} c_1D_x^2 + c_2uI + c_3vI \\ c_4D_x^2 + c_5uI + c_6vI \end{pmatrix}, \quad \begin{pmatrix} c_7D_x^2 + c_8uI + c_9vI \\ c_{10}D_x^2 + c_{11}uI + c_{12}vI \end{pmatrix}. (4.20)$$

Using (4.17), we have

$$\mathcal{R}_1 = \tilde{c}_{11}G^{(1)}D_x^{-1} \otimes \rho^{(1)'} = \tilde{c}_{11} \begin{pmatrix} u_x \\ v_x \end{pmatrix} D_x^{-1} \otimes (I \ 0) = c_{13} \begin{pmatrix} u_xD_x^{-1} \\ v_xD_x^{-1} \end{pmatrix}.$$
Substituting \( R = R_0 + R_1 \) into (4.10), we find \( c_1 = \cdots = c_{13} = 0 \). Therefore, either the form of \( R \) is incorrect or the system does not have a recursion operator. Let us now repeat the process taking \( s = 2 \), so rank \( R_{ij} = 4 \). Then,

\[
R = \begin{pmatrix} (R_0)_{11} & (R_0)_{12} \\ (R_0)_{21} & (R_0)_{22} \end{pmatrix} + \tilde{c}_{12} G^{(1)} D_x^{-1} \otimes \rho^{(2)*} + \tilde{c}_{21} G^{(2)} D_x^{-1} \otimes \rho^{(1)*},
\]

with \( (R_0)_{ij} \) a linear combination of \( \{D_x^4, u D_x^2, v D_x^2, u_x D_x, v_x D_x, u^2, uv, v^2, u_{2x}, v_{2x}\} \).

For instance,

\[
R_{12} = c_{11} D_x^4 + c_{12} u D_x^2 + c_{13} v D_x^2 + c_{14} u_x D_x + c_{15} v_x D_x + c_{16} u^2 I + c_{17} u v I + c_{18} v^2 I + c_{19} u_{2x} I + c_{20} v_{2x} I.
\]

Using (4.11), the first term of \( R_1 \) in (4.21) is

\[
R_1^{(1)} = \tilde{c}_{12} G^{(1)} D_x^{-1} \otimes \rho^{(2)*} = \tilde{c}_{12} \begin{pmatrix} u_x \\ v_x \end{pmatrix} D_x^{-1} \otimes \begin{pmatrix} 6uI & -4vI \\ -2u_x D_x^{-1} vI & 3v_x D_x^{-1} uI \end{pmatrix} = c_{41} \begin{pmatrix} 2u_x D_x^{-1} uI & -2u_x D_x^{-1} vI \\ 3v_x D_x^{-1} uI & -2v_x D_x^{-1} vI \end{pmatrix}.
\]

The second term of \( R_1 \) in (4.21) is

\[
R_1^{(2)} = \tilde{c}_{21} G^{(2)} D_x^{-1} \otimes \rho^{(1)*} = \tilde{c}_{21} \begin{pmatrix} F_1(u) \\ F_2(u) \end{pmatrix} D_x^{-1} \otimes \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} = c_{42} \begin{pmatrix} F_1(u) D_x^{-1} I & 0 \\ F_2(u) D_x^{-1} & 0 \end{pmatrix}.
\]

Substituting the form of \( R = R_0 + R_1 = R_0 + R_1^{(1)} + R_1^{(2)} \) into (4.10), the linear system for \( c_i \) has a non-trivial solution. Solving the linear system, we finally obtain

\[
R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix},
\]

where

\[
R_{11} = D_x^4 + 8u D_x^2 + 12u_x D_x + 16u^2 I + 8u_{2x} I - \frac{16}{3} v^2 I + 4u_x D_x^{-1} uI + 12u_u D_x^{-1} + 2u_{3x} D_x^{-1} - 8v_x D_x^{-1},
\]

\[
R_{12} = -\frac{20}{3} v D_x^2 - \frac{16}{3} v_x D_x^4 - \frac{16}{3} u v I - \frac{4}{3} v_{2x} I - \frac{8}{3} u_x D_x^{-1} vI,
\]

\[
R_{21} = -10v_x D_x^4 - 12v_{2x} I + 4v_x D_x^{-1} uI - 12u_{vv} D_x^{-1} - 4v_{3x} D_x^{-1},
\]

\[
R_{22} = -4D_x^4 + 16u D_x^2 - 8u_x D_x^4 - \frac{16}{3} v^2 I - \frac{8}{3} v_x D_x^{-1} vI.
\]

A similar algorithm for the symbolic computation of recursion operators of systems of differential-difference equations (DDEs) is given elsewhere in these proceedings [HSSW04]. The full implementation of these algorithms for PDEs and DDEs is a work in progress.

References

[AC91] M. J. Ablowitz, P. A. Clarkson, *Solitons, Nonlinear Evolution Equations and Inverse Scattering*, London Math. Soc. Lect. Note Ser. 149 (1991), Cambridge (U.K.): Cambridge University Press.

[ARS80] M. J. Ablowitz and A. Ramani and H. Segur, *A connection between nonlinear evolution equations and ordinary differential equations of P-type. I*, J. Math. Phys. 21 (1980), 715–721 and 1006–1015.
[BGH03] D. Baldwin, Ü. Göktaş, W. Hereman, L. Hong, R. S. Martino and J. C. Miller, *Symbolic computation of exact solutions expressible in hyperbolic and elliptic functions for nonlinear PDEs*, J. Symb. Comp. (2003), In Press.

[BH03] D. Baldwin and W. Hereman, The Mathematica packages are available at [http://www.mines.edu/~home/whereman/](http://www.mines.edu/~home/whereman/).

[C00] A. R. Chowdhury, *Painlevé Analysis and its Applications*, Monographs and Surveys in Pure and Applied Mathematics 105 (2000), Boca Raton: Chapman & Hall/CRC.

[C99] R. Conte (ed.), *The Painlevé Property: One Century Later*, CRM Series in Mathematical Physics (1999), New York: Springer-Verlag.

[GH97a] Ü. Göktaş and W. Hereman, *Symbolic computation of conserved densities for systems of nonlinear evolution equations*, J. Symb. Comp. 24 (1997), 591–621.

[GH97b] , Mathematica package *InvariantSymmetries.m* is available since 1997 from [http://www.mathsource.com/cgi-bin/msitem?0208-932](http://www.mathsource.com/cgi-bin/msitem?0208-932).

[GH99] , *Algorithmic computation of higher-order symmetries for nonlinear evolution and lattice equations*, Adv. Comp. Math. 11 (1999), 55–80.

[HG99] W. Hereman and Ü. Göktaş, *Integrability Tests for Nonlinear Evolution Equations*. Computer Algebra Systems: A Practical Guide, (M. Wester, ed.), Wiley, New York, 1999, pp. 211–232.

[HGC98] W. Hereman, Ü. Göktaş, M. D. Colagrosso and A. J. Miller, *Algorithmic integrability tests for nonlinear differential and lattice equations*, Comp. Phys. Comm. 115 (1998), 428–446.

[H97] W. Hereman and A. Nuseir, *Symbolic methods to construct exact solutions of nonlinear partial differential equations*, Math. Comp. Sim. 43 (1997), 13–27.

[HSSW04] W. Hereman, J. A. Sanders, J. Sayers and J. P. Wang, *Symbolic computation of conserved densities, generalized symmetries, and recursion operators for nonlinear differential-difference equations*, present proceedings.

[HT90] W. Hereman and M. Takaoka, *Solitary wave solutions of nonlinear evolution and wave equations using Macsyma*. J. Phys. A: Math. Gen. 23 (1990), 4805–4822.

[H97] W. Hereman and W. Zhang, *Symbolic software for soliton theory*, Acta Appl. Math. 39 (1995), 316–378.

[J03] R. S. Johnson, *On solutions of the Camassa-Holm equation*, Proc. Roy. Soc. Lond. A 459 (2003), 1687–1708.

[KJH97] M. D. Kruskal and N. Joshi and R. Halburd, *Analytic and Asymptotic Methods for Nonlinear Singularity Analysis: a Review and Extensions of Tests for the Painlevé Property*, Integrability and nonlinear systems, Lecture notes in physics, (Springer-Verlag, Heidelberg, 1997), pp. 171–205.

[ML96] W. Malfliet and W. Hereman, *The Tanh method: I. Exact solutions of nonlinear evolution and wave equations*, Phys. Scripta 54 (1996), 563–568.

[O93] P. J. Olver, *Applications of Lie groups to Differential Equations*, 2nd ed. Graduate Texts in Mathematics 107 (1993), New York: Springer-Verlag.

[W98] J. P. Wang, *Symmetries and conservation laws of evolution equations*, Ph.D. thesis, Thomas Stieltjes Institute for Mathematics, Amsterdam, September 1998.

[WTC83] J. Weiss and M. Tabor and G. Carnevale, *The Painlevé property for partial differential equations*, J. Math. Phys. 24 (1983), 522–526.

**Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401-1887, U.S.A.**

*E-mail address: dbaldwin@mines.EDU*

*E-mail address: whereman@mines.EDU*

**Physics Department 103-33, California Institute of Technology, Pasadena, CA 91125, U.S.A.**

*E-mail address: jack@its.caltech.edu*