Symbolic Software for the Painlevé Test of Nonlinear Ordinary and Partial Differential Equations

Douglas BALDWIN and Willy HEREMAN

Department of Mathematical and Computer Sciences, Colorado School of Mines
Golden, CO 80401, USA
E-mail: painlevetest@douglasbaldwin.com and whereman@mines.edu

Received April 22, 2005; Accepted in Revised Form June 5, 2005

Abstract

The automation of the traditional Painlevé test in Mathematica is discussed. The package PainleveTest.m allows for the testing of polynomial systems of nonlinear ordinary and partial differential equations which may be parameterized by arbitrary functions (or constants). Except where limited by memory, there is no restriction on the number of independent or dependent variables. The package is quite robust in determining all the possible dominant behaviors of the Laurent series solutions of the differential equation. The omission of valid dominant behaviors is a common problem in many implementations of the Painlevé test, and these omissions often lead to erroneous results. Finally, our package is compared with the other available implementations of the Painlevé test.

1 Introduction

Completely integrable nonlinear partial differential equations (PDEs) have remarkable properties, such as infinitely many generalized symmetries, infinitely many conservation laws, the Painlevé property, Bäcklund and Darboux transformations, bilinear forms, and Lax pairs (cf. [2, 11, 24, 25]). Completely integrable equations model physically interesting wave phenomena in reaction-diffusion systems, population and molecular dynamics, nonlinear networks, chemical reactions, and waves in material science. By investigating the complete integrability of a nonlinear PDE, one gains important insight into the structure of the equation and the nature of its solutions.

Broadly speaking, Painlevé analysis is the study of the singularity structure of differential equations. Specifically, a differential equation is said to have the Painlevé property if all the movable singularities of all its solutions are poles. There is strong evidence [48,50,51] that integrability is closely related to the singularity structure of the solutions of a differential equation (cf. [33,38]). For instance, dense branching of solutions around movable singularities has been shown to indicate nonintegrability [49].

At the turn of the nineteenth-century, Painlevé [30] and his colleagues classified all the rational second-order ODEs for which all the solutions are single-valued around all movable

Copyright © 2006 by D Baldwin and W Hereman
singularities. Equations possessing this property could either be solved in terms of known functions or transformed into one of the six Painlevé equations whose solutions define the Painlevé transcendents. The Painlevé transcendents cannot be expressed in terms of the classical transcendental functions, except for special values of their parameters [19].

The complex singularity structure of solutions was first used by Kovalevskaya in 1889 to identify a new integrable system of equations for the motion of a rotating top (cf. [14, 38]). Ninety years later, Ablowitz, Ramani and Segur (ARS) [2, 3] and McLeod and Olver [27] formulated the Painlevé conjecture which gives a useful necessary condition for determining whether a PDE is solvable using the Inverse Scattering Transform (IST) method. Specifically, the Painlevé conjecture asserts that every nonlinear ODE obtained by an exact reduction of a nonlinear PDE solvable by the IST-method has the Painlevé property. While necessary, the condition is not sufficient; in general, most PDEs do not have exact reductions to nonlinear ODEs and therefore satisfy the conjecture by default [41]. Weiss, Tabor and Carnevale (WTC) [44] proposed an algorithm for testing PDEs directly (which is analogous to the ARS algorithm for testing ODEs). For a thorough discussion of the traditional Painlevé property, see [1, 8, 10, 13, 18, 28, 31, 33, 38, 39].

There are numerous methods for solving completely integrable nonlinear PDEs, for instance by explicit transformations into linear equations or by using the IST-method [11]. Recently, progress has been made using Mathematica and Maple in applying the IST-method to difficult equations, including the Camassa-Holm equation [21]. While there is as yet no systematic way to determine if a differential equation is solvable using the IST-method [27], having the Painlevé property is a strong indicator that it will be.

There are several implementations of the Painlevé test in various computer algebra systems, including Reduce, Macsyma, Maple and Mathematica. The implementations described in [34, 35, 37] are limited to ODEs, while the implementations discussed in [16, 45–47] allow the testing of PDEs directly using the WTC algorithm. The implementation for PDEs written in Mathematica by Hereman et al. [16] is limited to two independent variables ($x$ and $t$) and is unable to find all the dominant behaviors in systems with undetermined exponents $\alpha_i$ (as is the case with the Hirota-Satsuma system). Our package PainleveTest.m [4] written in Mathematica syntax, allows the testing of polynomial PDEs (and ODEs) with no limitation on the number of differential equations or the number of independent variables (except where limited by memory). Our implementation also allows the testing of differential equations that have undetermined dominant exponents $\alpha_i$ and that are parameterized by arbitrary functions (or constants). The implementations for PDEs written in Maple by Xu and Li [45–47] were written after the one presented in this paper and are comparable to our implementation.

The paper is organized as follows: in Section 2 we review the basics of Painlevé analysis. Section 3 discusses the WTC algorithm for testing PDEs and uses the Korteweg-de Vries (KdV) equation and the Hirota-Satsuma system of coupled KdV (cKdV) equations to show the subtleties of the algorithm. We detail the algorithms to determine the dominant behavior, resonances, and constants of integration using a generalized system of coupled nonlinear Schrödinger (NLS) equations in Section 4. Additional examples are presented in Section 5 to illustrate the capabilities of the software. Section 6 compares our software package to other codes and briefly discusses the generalizations of the WTC algorithm. The use of the package PainleveTest.m [4] is shown in Section 7. We draw some conclusions and discuss the results in Section 8.
2 Painlevé Analysis

Consider a system of $M$ polynomial differential equations,

$$F_i(u(z), u'(z), u''(z), \ldots, u^{(m_i)}(z)) = 0, \quad i = 1, 2, \ldots, M,$$

(2.1)

where the dependent variable $u(z)$ has components $u_1(z), \ldots, u_M(z)$, the independent variable $z$ has components $z_1, \ldots, z_N$, and $u^{(k)}(z) = \partial^k u(z)/(\partial z_1^{k_1} \partial z_2^{k_2} \cdots \partial z_N^{k_N})$ denotes the collection of mixed derivative terms of order $k$. Let $m = \sum_{i=1}^M m_i$, where $m_i$ is the highest order in each equation. If there are any arbitrary coefficients (constants or analytic functions of $z$) parameterizing the system, we assume they are nonzero. For simplicity, in the examples we denote the components of $u(z)$ by $u(z), v(z), w(z), \ldots$, and the components of $z$ by $x, y, z, \ldots, t$.

A differential equation has the Painlevé property if all the movable singularities of all its solutions are poles. A singularity is movable if it depends on the constants of integration of the differential equation. For instance, the Riccati equation,

$$w'(z) + w^2(z) = 0,$$

(2.2)

has the general solution $w(z) = 1/(z - c)$, where $c$ is the constant of integration. Hence, (2.2) has a movable simple pole at $z = c$ because it depends on the constant of integration. Solutions of ODEs can have various kinds of singularities, including branch points and essential singularities; examples of the various types of singularities [23] are shown in Table 1. As a general property, solutions of linear ODEs have only fixed singularities [19].

<table>
<thead>
<tr>
<th>Type of Singularity</th>
<th>Equation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple fixed pole</td>
<td>$zw' + w = 0$</td>
<td>$w(z) = c/z$</td>
</tr>
<tr>
<td>Simple movable pole</td>
<td>$w' + w^2 = 0$</td>
<td>$w(z) = 1/(z - c)$</td>
</tr>
<tr>
<td>Movable algebraic branch point</td>
<td>$2wzw' - 1 = 0$</td>
<td>$w(z) = \sqrt{z - c}$</td>
</tr>
<tr>
<td>Movable logarithmic branch point</td>
<td>$w'' + w'^2 = 0$</td>
<td>$w(z) = \log(z - c_1) + c_2$</td>
</tr>
<tr>
<td>Non-isolated movable essential singularity</td>
<td>$(1 + w^2)w'' + (1 - 2w)w'^2 = 0$</td>
<td>$w(z) = \tan{\ln(c_1z + c_2)}$</td>
</tr>
</tbody>
</table>

Table 1. Examples of various types of singularities.

In general, a function of several complex variables cannot have an isolated singularity [29]. For example, $f(z) = 1/z$ has an isolated singularity at the point $z = 0$, but the function $f(w, z) = 1/z$ of two complex variables, $w = u + iv, z = x + iy$, has a two-dimensional manifold of singularities, namely the points $(u, v, 0, 0)$, in the four-dimensional space of these variables. Therefore, we will define a pole of a function of several complex variables as a point $(a_1, a_2, \ldots, a_N)$, in whose neighborhood the function can be written.
Symbolic Software for the Painlevé Test

in the form \( f(z) = \frac{h(z)}{g(z)} \), where \( g \) and \( h \) are both analytic in a region containing \((a_1, \ldots, a_N)\) in its interior, \( g(a_1, \ldots, a_N) = 0 \), and \( h(a_1, \ldots, a_N) \neq 0 \).

The WTC algorithm considers the singularity structure of the solutions around non-characteristic manifolds of the form \( g(z) = 0 \), where \( g(z) \) is an analytic function of \( z = (z_1, z_2, \ldots, z_N) \) in a neighborhood of the manifold. Specifically, if the singularity manifold is determined by \( g(z) = 0 \) and \( u(z) \) is a solution of the PDE, then one assumes a Laurent series solution

\[
 u_i(z) = g^{\alpha_i}(z) \sum_{k=0}^{\infty} u_{i,k}(z) g^k(z), \quad i = 1, 2, \ldots, M, \tag{2.3}
\]

where the coefficients \( u_{i,k}(z) \) are analytic functions of \( z \) with \( u_{i,0}(z) \neq 0 \) in a neighborhood of the manifold and the \( \alpha_i \) are integers with at least one exponent \( \alpha_i < 0 \). The requirement that the manifold \( g(z) = 0 \) is non-characteristic, ensures that the expansion (2.3) is well defined in the sense of the Cauchy-Kovalevskaya theorem [41, 43].

Substituting (2.3) into (2.1) and equating coefficients of like powers of \( g(z) \) determines the possible values of \( \alpha_i \) and defines a recursion relation for \( u_{i,k}(z) \). The recursion relation is of the form

\[
 Q_k u_k = G_k(u_0, u_1, \ldots, u_{k-1}, g(z)), \quad u_k = (u_{1,k}, u_{2,k}, \ldots, u_{M,k})^T, \tag{2.4}
\]

where \( Q_k \) is an \( M \times M \) matrix and \( T \) denotes transpose.

For (2.1) to pass the Painlevé test, the series (2.3) should have \( m-1 \) arbitrary functions as required by the Cauchy-Kovalevskaya theorem (as \( g(z) \) is the \( m \)-th arbitrary function). If so, the Laurent series solution corresponds to the general solution of the equation [1]. The \( m-1 \) arbitrary functions \( u_{i,k}(z) \) occur when \( k \) is one of the roots of \( \det(Q_k) \). These roots \( r_1 \leq r_2 \leq \cdots \leq r_m \) are called resonances. The resonances are also equal to the Fuchs indices of the auxiliary equations of Darboux [7].

Since the WTC algorithm is unable to detect essential singularities, it is only a necessary condition for the PDE to have the Painlevé property [6]. While rarely done in practice, sufficiency is proved by finding a transformation which linearizes the differential equation, yields an auto-Bäcklund transformation, a Bäcklund transformation, or hodographic transformation [15] to another differential equation which has the Painlevé property (see [8, 23, 33] for more information).

3 Algorithm for the Painlevé Test

In this section, we outline the WTC algorithm for testing PDEs for the Painlevé property. We discuss the Kruskal simplification and the Painlevé test of ODEs after the three main steps are outlined. Each of these steps is illustrated using both the KdV equation and the cKdV equations due to Hirota and Satsuma. Details of the three main steps of the algorithm are postponed till Section 4.

Step 1 (Determine the dominant behavior). It is sufficient to substitute

\[
 u_i(z) = \chi_i g^{\alpha_i}(z), \quad i = 1, 2, \ldots, M, \tag{3.1}
\]

where \( \chi_i \) is a constant, into (2.1) to determine the leading exponents \( \alpha_i \). In the resulting polynomial system, equating every two or more possible lowest exponents of \( g(z) \) in each
equation gives a linear system for $\alpha_i$. The linear system is then solved for $\alpha_i$ and each solution branch is investigated. The traditional Painlevé test requires that all the $\alpha_i$ are integers and that at least one is negative.

If any of the $\alpha_i$ are non-integer in a given branch, then that branch of the algorithm terminates. A non-integer $\alpha_i$ implies that some solutions of (2.1) have movable algebraic branch points. Often, a suitable change of variables in (2.1) can remove the algebraic branch point. An alternative approach is to use the “weak” Painlevé test, which allows certain rational $\alpha_i$ and resonances; see [13,18,32,33] for more information.

If one or more $\alpha_i$ remain undetermined, we assign integer values to the free $\alpha_i$ so that every equation in (2.1) has at least two different terms with equal lowest exponents.

For each solution $\alpha_i$, we substitute
\[
   u_i(z) = u_{i,0}(z)g^{\alpha_i}(z), \quad i = 1, 2, \ldots, M,
\]
into (2.1). We then solve the (typically) nonlinear equation for $u_{i,0}(z)$, which is found by balancing the leading terms. By leading terms, we mean those terms with the lowest exponent of $g(z)$. If any of the solutions contradict the assumption that $u_{i,0}(z) \neq 0$, then that branch of the algorithm fails the Painlevé test.

If any of the $\alpha_i$ are non-integer, all the $\alpha_i$ are positive, or there is a contradiction with the assumption that $u_{i,0}(z) \neq 0$, then that branch of the algorithm terminates and does not pass the Painlevé test for that branch.

**Step 2 (Determine the resonances).** For each $\alpha_i$ and $u_{i,0}(z)$, we calculate the $r_1 \leq \cdots \leq r_m$ for which $u_{i,r}(z)$ is an arbitrary function in (2.3). To do this, we substitute
\[
   u_i(z) = u_{i,0}(z)g^{\alpha_i}(z) + u_{i,r}(z)g^{\alpha_i+r}(z)
\]
into (2.1), and keep only the lowest order terms in $g(z)$ that are linear in $u_{i,r}$. This is done by computing the solutions for $r$ of $\det(Q_r) = 0$, where the $M \times M$ matrix $Q_r$ satisfies
\[
   Q_r u_r = 0, \quad u_r = (u_{1,r} \ u_{2,r} \ \cdots \ u_{M,r})^T.
\]

If any of the resonances are non-integer, then the Laurent series solutions of (2.1) have a movable algebraic branch point and the algorithm terminates. If $r_m$ is not a positive integer, then the algorithm terminates; if $r_1 = -1, r_2 = \cdots = r_m = 0$ and $m-1$ of the $u_{i,0}(z)$ found in Step 1 are arbitrary, then (2.1) passes the Painlevé test. If (2.1) is parameterized, the values for $r_1 \leq \cdots \leq r_m$ may depend on the parameters, and hence restrict the allowable values for the parameters.

There is always a resonance $r = -1$ which corresponds to the arbitrariness of $g(z)$; as such, it is often called the universal resonance. When there are negative resonances other than $r = -1$, (or, more than one resonance equals $-1$) then the Laurent series solution is not the general solution and further analysis is needed to determine if (2.1) passes the Painlevé test. The perturbative Painlevé approach, developed by Conte et al. [9], is one method for investigating negative resonances.

**Step 3 (Find the constants of integration and check compatibility conditions).** For the system to possess the Painlevé property, the arbitrariness of $u_{i,r}(z)$ must be verified up to the highest resonance level. This is done by substituting
\[
   u_i(z) = g^{\alpha_i}(z) \sum_{k=0}^{r_m} u_{i,k}(z)g^k(z)
\]
into (2.1), where \( r_m \) is the largest positive integer resonance.

For (2.1) to have the Painlevé property, the \((M + 1) \times M\) augmented matrix \((Q_k|G_k)\) must have rank \( M \) when \( k \neq r \) and rank \( M - s \) when \( k = r \), where \( s \) is the algebraic multiplicity of \( r \) in \( \det(Q_r) = 0 \), \( 1 \leq k \leq r_m \), and \( Q_k \) and \( G_k \) are as defined in (2.4). If the augmented matrix \((Q_k|G_k)\) has the correct rank, solve the linear system (2.4) for \( u_{1,k}(z), \ldots, u_{M,k}(z) \) and use the results in the linear system at level \( k + 1 \).

If the linear system (2.4) does not have a solution, then the Laurent series solution of (2.1) is free of movable logarithmic branch points and (2.1) passes the Painlevé test.

If the algorithm does not terminate, then the Laurent series solutions of (2.1) are free of movable algebraic or logarithmic branch points and (2.1) passes the Painlevé test.

The Painlevé test of PDEs is quite cumbersome; in particular, Step 3 is lengthy and prone to error when done by hand. To simplify Step 3, Kruskal proposed a simplification which now bears his name. In the context of the WTC algorithm, it is sometimes called the Weiss-Kruskal simplification [20, 23]. The manifold defined by \( g(z) = 0 \) is non-characteristic, that means \( g_{\alpha}(z) \neq 0 \) for some \( l \) on the manifold \( g(z) = 0 \). By the implicit function theorem, we can then locally solve \( g(z) = 0 \) for \( z_l \), so that

\[
g(z) = z_l - h(z_1, \ldots, z_{l-1}, z_{l+1}, \ldots, z_N),
\]

for some arbitrary function \( h \). Using (3.6) greatly simplifies the computation of the constants of integration \( u_{i,k}(z_1, \ldots, z_{l-1}, z_{l+1}, \ldots, z_N) \). However, with the Kruskal simplification one loses the ability to use the Weiss truncation method [42] to find a linearising transformation, an auto-Bäcklund transformation, or a Bäcklund transformation (see [8]).

When testing ODEs, (2.3) must be replaced by

\[
u_i(z) = \sum_{k=0}^{\infty} u_{i,k} g^k(z), \quad i = 1, 2, \ldots, M,
\]

where the coefficients \( u_{i,k} \) are constants, \( g(z) = z - z_0 \), and \( z_0 \) is an arbitrary constant. If \( g \) explicitly occurs in the ODE, then it is (automatically) replaced by \( g(z) + z_0 \) prior to Step 1 of the test. An example of the Painlevé test of an ODE is given in Section 5.

### 3.1 The Korteweg-de Vries equation

To illustrate the steps of the algorithm, let us examine the KdV equation [1],

\[
u_t + 6\nu u_x + u_{3x} = 0,
\]

the most famous completely integrable PDE from soliton theory. Note that for simplicity, we use \( u_{i,x} = u_{i,x,x} = \partial^i u/\partial x^i \) and \( g_{xx} = \partial^2 g/\partial x^2 \) when \( i \geq 3 \).

Substituting (3.1) into (3.8) gives

\[
\alpha \chi \{ g \chi g^{-1} + 6 \chi g_x + 2 \chi (\alpha - 1) (\alpha - 2) g_x g_x + 3 \chi g_{xx} g_x + g^2 g_{3x} \} = 0.
\]

The lowest exponents of \( g(x,t) \) are \( \alpha = 3 \) and \( 2\alpha - 1 \). Equating these leading exponents gives \( \alpha = -2 \). Substituting (3.2), \( u(x, t) = u_0(x, t)g^{-2}(x, t) \), into (3.8) and requiring that the leading terms (in \( g^{-5}(x, t) \)) balance, gives \( u_0(x, t) = -\frac{2}{g_x^2}(x, t) \).
Substituting (3.3), \( u(x, t) = -2g^2_2(x, t)g^{-2}(x, t) + u_r(x, t)g^{-2}(x, t) \), into (3.8) and equating the coefficients of the dominant terms (in \( g^{-5}(x, t) \)) that are linear in \( u_r(x, t) \) gives

\[
(r - 6)(r - 4)(r + 1)g_x(x, t)^3 = 0.
\]

(3.10)

Assuming \( g_x(x, t) \neq 0 \), the resonances of (3.8) are \( r_1 = -1, r_2 = 4 \) and \( r_3 = 6 \).

We now substitute

\[
u(x, t) = g^{-2}(x, t) \sum_{k=0}^{6} u_k(x, t)g^k(x, t)
\]

\[
= -2g^2_2(x, t)g^{-2}(x, t) + u_1(x, t)g^{-1}(x, t) + \cdots + u_6(x, t)g^4(x, t)
\]

(3.11)

into (3.8) and group the terms of like powers of \( g(x, t) \). So, we will pull off the coefficients of \( g^{k-5}(x, t) \) at level \( k \). Equating the coefficients of \( g^{-4}(x, t) \) to zero at level \( k = 1 \), gives

\[
u_1(x, t)g^2_2(x, t) = 2g^3_2(x, t)g_{xx}(x, t).\]

Setting \( u_1(x, t) = 2g_{xx}(x, t) \), we get

\[
u_2(x, t) = -\frac{9g^2_2 + 3g_2g_{xx} - 4g^2_5g_{3x}}{6g^2_2}
\]

(3.12)

at level \( k = 2 \). Similarly, at level \( k = 3 \),

\[
u_3(x, t) = \frac{g^2_2g_{xx} - 9g_2g_{xxxx} + 3g^3_{xx} - 4g_5g_{xx}g_{3x} + g^2_5g_{4x}}{6g^2_2}
\]

(3.13)

At level \( k = r_2 = 4 \), we find

\[
(u_1)_t + 6\{u_3(u_0)_x + u_2(u_1)_x + u_1(u_2)_x + u_0(u_3)_x\} + (u_1)_3x = 0,
\]

(3.14)

which is trivially satisfied upon substitution of the solutions of \( u_0(x, t), \ldots, u_3(x, t) \). Therefore, the compatibility condition at level \( k = r_2 = 4 \) is satisfied and \( u_4(x, t) \) is indeed arbitrary. At level \( k = 5 \), \( u_5(x, t) \) is unambiguously determined, but not shown due to length. Finally, the compatibility condition at level \( k = r_3 = 6 \) is trivially satisfied when the solutions for \( u_0(x, t), \ldots, u_3(x, t) \) and \( u_5(x, t) \) are substituted into the recursion relation at that resonance level.

Therefore, the Laurent series solution \( u(x, t) \) of (3.8) in the neighborhood of \( g(x, t) = 0 \) is free of algebraic and logarithmic movable branch points. Furthermore, since the Laurent series solution,

\[
u(x, t) = g^{-2}(x, t) \sum_{k=0}^{\infty} u_k(x, t)g^k(x, t),
\]

(3.15)

has three arbitrary functions, \( g(x, t), u_4(x, t), \) and \( u_6(x, t), \) (as required by the Cauchy-Kovalevskaia theorem since (3.8) is of third order) it is also the general solution. Hence, we conclude that (3.8) passes the Painlevé test.

The Weiss-Kruskal simplification uses \( g(x, t) = x - h(t) \). Consequently, \( g_x = 1, g_{xx} = g_{3x} = \cdots = 0 \), and the Laurent series,

\[
u(x, t) = g^{-2}(x, t) \sum_{k=0}^{\infty} u_{i,k}(t)g^k(x, t),
\]

(3.16)
we obtain the resonances $r_1, r_2$ becomes

$$
u(t) = a(6uu_x + u_{3x}) - 2vv_x, \quad a > 0,$$

$$v_t = -3uv_x - v_{3x}. \quad (3.18)$$

Again, we substitute (3.1), $u(x,t) = \chi_1 g^{\alpha_1}(x,t)$ and $v(x,t) = \chi_2 g^{\alpha_2}(x,t)$, into (3.18) and pull off the lowest exponents of $g(x,t)$. From the first equation, we get $\alpha_1 = 3, 2\alpha_1 - 1$, and $2\alpha_2 - 1$. From the second equation, we get $\alpha_2 - 3$ and $\alpha_1 + \alpha_2 - 1$. Hence, $\alpha_1 = -2$ from the second equation. Substituting this into the first equation gives $\alpha_2 \geq -2$.

Substituting (3.2) into (3.18) and requiring that at least two leading terms balance gives us two branches: $\alpha_1 = \alpha_2 = -2$ and $\alpha_1 = -2, \alpha_2 = -1$. The branches with $\alpha_1 = -2$ and $\alpha_2 = 0$ are excluded for they require that either $u_0(x,t)$ or $v_0(x,t)$ is identically zero.

Continuing with the two branches and solving for $u_0(x,t)$ and $v_0(x,t)$ gives

$$
\begin{cases}
\alpha_1 = \alpha_2 = -2, \\
u_0(x,t) = -4g_2^2(x,t), \\
v_0(x,t) = \pm 2\sqrt{6}ag_2^2(x,t),
\end{cases}
\quad \begin{cases}
\alpha_1 = -2, \alpha_2 = -1, \\
u_0(x,t) = -2g_2^2(x,t), \\
v_0(x,t) \text{ arbitrary.}
\end{cases}
\quad (3.19)
\$$

For the branch with $\alpha_1 = \alpha_2 = -2$, substituting (3.3),

$$
\begin{cases}
u(x,t) = -4g_2^2(x,t)g^{-2}(x,t) + u_r(x,t)g^{-2}(x,t), \\
v(x,t) = \pm 2\sqrt{6}ag_2^2(x,t)g^{-2}(x,t) + v_r(x,t)g^{-2}(x,t),
\end{cases}
\quad (3.20)
\$$

into (3.18) and equating to zero the coefficients of the lowest order terms in $g(x,t)$ that are linear in $u_r$ and $v_r$ gives

$$
\begin{pmatrix}
-(r-4)(r^2 - 5r - 18)ag_2^3(x,t) \\
\mp 4(r-4)\sqrt{6}ag_2^2(x,t)
\end{pmatrix}
\begin{pmatrix}
\pm 12\sqrt{6}ag_2^3(x,t) \\
(r-2)(r-7)rg_2^3(x,t)
\end{pmatrix}
\begin{pmatrix}
u_r(x,t) \\
v_r(x,t)
\end{pmatrix}
= 0. \quad (3.21)
$$

From

$$\det(Q_r) = -a(r + 2)(r + 1)(r - 3)(r - 4)(r - 6)(r - 8)g_2^6(x,t) = 0, \quad (3.22)$$

we obtain the resonances $r_1 = -2, r_2 = -1, r_3 = 3, r_4 = 4, r_5 = 6, \text{ and } r_6 = 8$.

By convention, the resonance $r_1 = -2$ is ignored since it violates the hypothesis that $g(x,t)^{-2}$ is the dominant term in the expansion near $g(z) = 0$. Furthermore, this is not a principal branch since the series has only five arbitrary functions instead of the required six.
Then, the constants of integration are found by substituting (3.18) and pulling off the coefficients of $g^{k-5}(x,t)$. At level $k = 1$, 

$$
\begin{pmatrix}
11ag_2^2(x,t) & \pm2\sqrt{6}ag_2^2(x,t) \\
\pm2\sqrt{6}ag_2^2(x,t) & -g_2^2(x,t)
\end{pmatrix}
\begin{pmatrix}
u_1(x,t) \\
v_1(x,t)
\end{pmatrix}
=
\begin{pmatrix}
u_1(x,t) \\
v_1(x,t)
\end{pmatrix} = 
\begin{pmatrix}
20ag_2^2(x,t)g_{xx}(x,t) \\
\pm10\sqrt{6}ag_2^2(x,t)g_{xx}(x,t)
\end{pmatrix},
$$

(3.23)

and thus,

$$u_1(x,t) = 4g_{xx}(x,t), \quad v_1(x,t) = \pm2\sqrt{6}ag_{xx}(x,t).$$

(3.24)

At level $k = 2$,

$$u_2(x,t) = \frac{3g_{xx}^2(x,t) - g_x(x,t)(g_t(x,t) + 4g_{3t}(x,t))}{3g_2^2(x,t)},$$

$$v_2(x,t) = \pm\frac{(1 + 2a)g_t(x,t)g_x(x,t) + 4ag_x(x,t)g_{3t}(x,t) - 3ag_{xx}^2(x,t)}{\sqrt{6}ag_2^2(x,t)}.$$

(3.25)

The compatibility conditions at levels $k = r_3 = 3$ and $k = r_4 = 4$ are trivially satisfied. At levels $k = 5$ and $k = 7$, $u_k(x,t)$ and $v_k(x,t)$ are unambiguously determined (not shown). At resonance levels $k = r_5 = 6$ and $k = r_6 = 8$, the compatibility conditions require $a = \frac{1}{2}$.

Likewise, for the branch with $\alpha_1 = -2, \alpha_2 = -1$, substituting (3.3),

$$\begin{cases}
u(x,t) = -2g_2^2(x,t)g^{-2}(x,t) + u_t(x,t)g^{-2}(x,t), \\
v_t(x,t) = v_0(x,t)g^{-1}(x,t) + v(x,t)g^{-1}(x,t),
\end{cases}$$

(3.26)

into (3.18) gives

$$
\begin{pmatrix}
-a(r + 1)(r - 4)(r - 6)g_x^2(x,t) & -3v_0(x,t)g_x(x,t) \\
0 & r(r - 1)(r - 5)g_2^2(x,t)
\end{pmatrix}
\begin{pmatrix}
u(x,t) \\
v_t(x,t)
\end{pmatrix} = 0.
$$

(3.27)

Since

$$\det(Q_r) = -a(r + 1)r(r - 4)(r - 5)(r - 6)g_x^6(x,t),$$

(3.28)

the resonances are $r_1 = -1, r_2 = 0, r_3 = 1, r_4 = 4, r_5 = 5$, and $r_6 = 6$.

Since $r_2 = 0$ is a resonance, there must be freedom at level $k = r_2 = 0$; indeed, coefficient $u_0(x,t) = -2g_2^2(x,t)$ is unambiguously determined but $v_0(x,t)$ is arbitrary. Then, the constants of integration are found by substituting

$$\begin{cases}
u(x,t) = -2g_2^2(x,t)g^{-2}(x,t) + u_1(x,t)g^{-1}(x,t) + \cdots + u_6(x,t)g^4(x,t), \\
v_t(x,t) = v_0(x,t)g^{-1}(x,t) + v_1(x,t) + \cdots + v_6(x,t)g^5(x,t),
\end{cases}$$

(3.29)

into (3.18) and pulling off the coefficients of $g^{k-5}(x,t)$ in the first equation and $g^{k-4}(x,t)$ in the second equation. At level $k = r_3 = 1$,

$$
\begin{pmatrix}
a & 0 \\
v_0(x,t) & 0
\end{pmatrix}
\begin{pmatrix}
u_1(x,t) \\
v_1(x,t)
\end{pmatrix} = 
\begin{pmatrix}
2ag_{xx}(x,t) \\
2v_0(x,t)g_{xx}(x,t)
\end{pmatrix}.
$$

(3.30)
So, \( u_1(x, t) = 2g_{xx}(x, t) \) and \( v_1(x, t) \) is arbitrary. At level \( k = 2 \),
\[
\begin{pmatrix}
12ag_x^2 & 0 \\
-6v_0g_x & -6g_x^3
\end{pmatrix}
\begin{pmatrix}
u_2 \\
v_2
\end{pmatrix}
= \begin{pmatrix}
2g_xg_z + 6ag_{xx}^2 - v_0^2 - 8ag_zg_{xx} \\
v_0g_t + 6(v_1)g_x + 3(v_0)g_{xx} + 3(v_0)v + v_0g_{xx}
\end{pmatrix},
\]
(3.31)
which unambiguously determines \( u_2(x, t) \) and \( v_2(x, t) \). Similarly, the coefficients in the
Laurent series solution are unambiguously determined at level \( k = 3 \). At resonance level
\( k = r_4 = 4 \), the compatibility condition is trivially satisfied. At resonance levels \( k = r_5 = 5 \)
and \( k = r_6 = 6 \), the compatibility conditions require \( a = \frac{1}{2} \).

Therefore, (3.18) satisfies the necessary conditions for having the Painlevé property
when \( a = \frac{1}{2} \), a fact confirmed by other analyses of complete integrability [1].

4 Key Algorithms

In this section, we present the three key algorithms in greater detail. To illustrate the
steps we consider a generalization
\[
\begin{align*}
iv_t + u_{xx} + (|u|^2 + \beta|v|^2)u + a(x, t)u + c(x, t)v &= 0, \\
v_t + v_{xx} + (|v|^2 + \beta|u|^2)v + b(x, t)v + d(x, t)u &= 0,
\end{align*}
\]
(4.1)
of the coupled nonlinear Schrödinger (NLS) equations [40]
\[
\begin{align*}
iv_t + u_{xx} + (|u|^2 + \beta|v|^2)u &= 0, \\
v_t + v_{xx} + (|v|^2 + \beta|u|^2)v &= 0.
\end{align*}
\]
(4.2)
In (4.1), \( a(x, t), \ldots, d(x, t) \) are arbitrary complex functions and \( \beta \) is a real constant pa-
rameter. Since all the functions in (4.1) are complex, we write the system as
\[
\begin{align*}
iv_t + u_{xx} + (u\bar{u} + \beta v\bar{v})u + a(x, t)u + c(x, t)v &= 0, \\
v_t - \bar{v}_{xx} - (u\bar{u} + \beta v\bar{v})\bar{u} - \bar{a}(x, t)\bar{u} - \bar{c}(x, t)\bar{v} &= 0, \\
v_t + v_{xx} + (v\bar{v} + \beta u\bar{u})v + b(x, t)v + d(x, t)u &= 0, \\
v_t - \bar{v}_{xx} - (v\bar{v} + \beta u\bar{u})\bar{v} - \bar{b}(x, t)\bar{v} - \bar{d}(x, t)\bar{u} &= 0,
\end{align*}
\]
(4.3)
and treat \( u, \bar{u}, v, \) and \( \bar{v} \) as independent complex functions. As is customary, the variables
with overbars denote complex conjugates.

4.1 Algorithm to determine the dominant behavior

Determining the dominant behavior of (2.1) is delicate and the omission of valid dominant
behaviors often leads to erroneous results [33].

**Step 1 (Substitute the leading-order ansatz).** To determine the values of \( \alpha_i \), it is sufficient
to substitute \( u_i(z) = \chi_i g(z)^{\alpha_i} \) into (2.1), where \( \chi_i \) is constant and \( g(z) \) is an analytic
function in a neighborhood of the non-characteristic manifold defined by \( g(z) = 0 \).

**Step 2 (Collect exponents and prune non-dominant branches).** The balance of exponents
must come from different terms in (2.1). For each equation \( \dot{F}_i = 0 \), collect the exponents of
\( g(z) \). Then, remove non-dominant exponents and duplicates (that come from the same term
in (2.1)). For example, $\alpha_1 + 1$ is non-dominant and can be removed from \{\(\alpha_1 - 1, \alpha_1 + 1\}\) since \(\alpha_1 - 1 < \alpha_1 + 1\).

For (4.3), the exponents corresponding to each equation are

\[
\begin{align*}
F_1 & : \{\alpha_1 - 2, 2\alpha_1 + \alpha_2, \alpha_1 + \alpha_3 + \alpha_4\}, \\
F_2 & : \{\alpha_2 - 2, \alpha_1 + 2\alpha_2, \alpha_2 + \alpha_3 + \alpha_4\}, \\
F_3 & : \{\alpha_3 - 2, 2\alpha_3 + \alpha_4, \alpha_1 + \alpha_2 + \alpha_3\}, \\
F_4 & : \{\alpha_4 - 2, \alpha_3 + 2\alpha_4, \alpha_1 + \alpha_2 + \alpha_4\},
\end{align*}
\]

(4.4)

after duplicates and non-dominant exponents have been removed.

**Step 3 (Combine expressions and compute relations for \(\alpha_i\)).** For each \(F_i\) separately, equate all possible combinations of two elements. Then, construct relations between the \(\alpha_i\) by solving for \(\alpha_1, \alpha_2, \text{etc.}, \) one at a time.

For (4.4), we get

\[
\begin{align*}
F_1 & : \{\alpha_1 - 2 = 2\alpha_1 + \alpha_2, \alpha_1 - 2 = \alpha_1 + \alpha_3 + \alpha_4, \\
& \quad 2\alpha_1 + \alpha_2 = \alpha_1 + \alpha_3 + \alpha_4\} \\
& \Rightarrow \{\alpha_1 + \alpha_2 = -2, \alpha_3 + \alpha_4 = -2, \alpha_1 + \alpha_2 = \alpha_3 + \alpha_4\}. \\
\end{align*}
\]

(4.5)

For \(F_2, F_3\) and \(F_4\) we again find that \{\(\alpha_1 + \alpha_2 = -2, \alpha_3 + \alpha_4 = -2, \alpha_1 + \alpha_2 = \alpha_3 + \alpha_4\}\}.

**Step 4 (Combine equations and solve for exponents \(\alpha_i\)).** By combining the sets of expressions in an “outer product” fashion, we generate all the possible linear equations for \(\alpha_i\). Solving these linear systems, we form a set of all possible solutions for \(\alpha_i\).

For (4.3), we have three sets of linear equations

\[
\begin{align*}
\{\alpha_1 + \alpha_2 = -2 \} & \Rightarrow \begin{cases} 
\alpha_1 + \alpha_2 = -2, \\
\alpha_3 + \alpha_4 \geq -2,
\end{cases} \\
\{\alpha_3 + \alpha_4 = -2 \} & \Rightarrow \begin{cases} 
\alpha_1 + \alpha_2 \geq -2, \\
\alpha_3 + \alpha_4 = -2,
\end{cases}
\end{align*}
\]

(4.6)  (4.7)

and

\[
\begin{align*}
\{\alpha_1 + \alpha_2 = \alpha_3 + \alpha_4, \} & \Rightarrow \{\alpha_1 + \alpha_2 = \alpha_3 + \alpha_4 \geq -2. \}
\end{align*}
\]

(4.8)

Although the algorithm treats \(u, \bar{u}, v,\) and \(\bar{v}\) as independent complex functions, we know that \(\alpha_1 = \alpha_2\) and \(\alpha_3 = \alpha_4\) because \(\bar{u}\) and \(\bar{v}\) are the complex conjugates of \(u\) and \(v\).

Our package PainLeveTest.m can take advantage of such additional information by using the option DominantBehaviorConstraints -> \{\(\text{alpha[1]} = \text{alpha[2]}, \text{alpha[3]} = \text{alpha[4]}\}\}. Using this additional information yields three cases, \(\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 \geq -1\) and \(\alpha_1 = \alpha_2 = -1, \alpha_3 = \alpha_4 \geq -1\) and \(\alpha_1 = \alpha_2 \geq -1, \alpha_3 = \alpha_4 = -1\).

**Step 5 (Fix the undetermined \(\alpha_i\)).** First, compute the minimum values for the undetermined \(\alpha_i\). If a minimum value cannot be determined, then the user-defined value DominantBehaviorMin is used. If so, the value of the free \(\alpha_i\) is counted up to a user defined DominantBehaviorMax. If neither of the bounds is set, the software will run the
test for the default values \( \alpha_i = -1, -2 \) and \(-3\). For maximal flexibility, with the option \texttt{DominantBehavior} one can also run the code for user-specified values of \( \alpha_i \). An example is given in Section 7. In any case, the selected or given dominant behaviors are checked for consistency with (2.1).

For (4.3), if we take \( \alpha_1, \ldots, \alpha_4 < 0 \), then we are left with only one branch

\[
\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = -1. \tag{4.9}
\]

**Step 6 (Compute the first terms in the Laurent series).** Using the values for \( \alpha_i \), substitute

\[
u_i(z) = u_{i,0}(z)g^{\alpha_i}(z) \tag{4.10}
\]

into (2.1) and solve the resulting (typically) nonlinear equations for \( u_{i,0}(z) \) using the assumption that \( u_{i,0}(z) \neq 0 \).

For (4.3), we find

\[
\begin{cases}
\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = -1, \\
u_0(x,t) = -2g_\alpha^2(x,t)(1 + \beta)^{-1}u_0^{-1}(x,t), \\
v_0(x,t) = -2g_\alpha^3(x,t)(1 + \beta)^{-1}v_0^{-1}(x,t),
\end{cases} \tag{4.11}
\]

where \( u_0(x,t) \) and \( v_0(x,t) \) are arbitrary functions.

If we do not restrict \( \alpha_1, \ldots, \alpha_4 < 0 \), then there are contradictions with the assumption \( u_{i,0}(z) \neq 0 \) for all but two possible dominant behaviors,

\[
\begin{cases}
\alpha_1 = \alpha_2 = -1, \\
\alpha_3 \geq 3, \\
\alpha_4 \geq 3,
\end{cases} \quad \text{and} \quad \begin{cases}
\alpha_1 \geq 3, \\
\alpha_2 \geq 3, \\
\alpha_3 = \alpha_4 = -1.
\end{cases} \tag{4.12}
\]

### 4.2 Algorithm to determine the resonances

**Step 1 (Construct matrix \( Q_r \)).** Substitute

\[
u_i(z) = u_{i,0}(z)g^{\alpha_i}(z) + u_{i,r}(z)g^{\alpha_i+r}(z) \tag{4.13}
\]

into (2.1). Then, the \((i,j)\)-th entry of the \( M \times M \) matrix \( Q_r \) is the coefficients of the linear terms in \( u_{j,r}(z) \) of the leading terms in equation \( F_i = 0 \).

**Step 2 (Find the roots of \( \det(Q_r) \)).** The resonances are the solutions of \( \det(Q_r) = 0 \). If any of these solutions (in a particular branch) is non-integer, then that branch of the algorithm terminates since it implies that some solutions of (2.1) have movable algebraic branch point. If any of the resonances are rational, then a change of variables in (2.1) may remove the algebraic branch point. Such changes are not carried out automatically.

For branch (4.11),

\[
\det(Q_r) = (r - 4)(r - 3)^2 \left( r^2(1 + \beta) - 3r(1 + \beta) - 4(1 - \beta) \right) r^2(r + 1) \\
\times (1 + \beta)^5 u_0^5(x,t)v_0^5(x,t)g_\alpha^8(x,t). \tag{4.14}
\]

Since the roots of (4.14) for \( r \) depend on the constant parameter \( \beta \), we must choose values of \( \beta \) so that all the solutions are integers before proceeding. For \( \beta = 1 \), the resonances are \( r_1 = -1, r_2 = r_3 = r_4 = 0, r_5 = r_6 = r_7 = 3, r_8 = 4 \).
While taking $\beta = 0$ also yields all integer resonances, it violates the assumption that all the parameters in (2.1) are nonzero. Allowing the parameters in (2.1) to be zero could cause a false balance in Algorithm 4.1. Thus, (2.1) with $\beta = 0$ should be treated separately. In this example however, setting $\beta = 0$ does not affect the dominant behavior and the resonances are $r_1 = r_2 = -1, r_3 = r_4 = 0, r_5 = r_6 = 3,$ and $r_7 = r_8 = 4.$

Although taking $\beta = 25/7$ leads to rational resonances at $r_4 = r_5 = 3/2,$ they are not easily resolved by a change of variables in (4.3). The branches with dominant behavior, $\alpha_1 = \alpha_2 = -1, \alpha_3 \geq \alpha_4 \geq 3,$ have resonances $r_1 = -\alpha_3 - 1, r_2 = -\alpha_3 + 2, r_3 = -\alpha_4 - 1, r_4 = -\alpha_4 + 2, r_5 = -1, r_6 = 0, r_7 = 3$ and $r_8 = 4.$ Since $r_1, r_2 < -1$ when $\alpha_3 = \alpha_4 = 3,$ $r_1, r_2, r_3 < -1$ when $\alpha_3 > \alpha_4 = 3,$ and $r_1, \ldots, r_4 < -1$ when $\alpha_3, \alpha_4 > 3,$ these are not principal branches and should be investigated using the perturbative Painlevé approach [9].

4.3 Algorithm to determine the constants of integration and check compatibility conditions

Step 1 (Generate the system for the coefficients of the Laurent series at level $k$). Substitute

$$u_i(z) = g^{\alpha_i}(z) \sum_{k=0}^{r_m} u_{i,k}(z) g^k(z)$$

(4.15)

into (2.1) and multiply $F_i$ by $g^{-\gamma_i}(z),$ where $\gamma_i$ is the lowest exponent of $g(z)$ in $F_i.$ The equations for determining the coefficients of the Laurent series at level $k$ then arise by equating to zero the coefficients of $g^k(z).$ These equations, at level $k,$ are linear in $u_{i,k}(z)$ and depend only on $u_{i,j}(z)$ and $g(z)$ (and their derivatives) for $1 \leq i \leq M$ and $0 \leq j < k.$ Thus, the system can be written as

$$Q_k u_k = G_k(u_0, u_1, \ldots, u_{k-1}, g, z),$$

(4.16)

where $u_k = (u_{1,k}(z), \ldots, u_{M,k}(z))^T.$

Step 2 (Solve the linear system for the coefficients of the Laurent series). If the rank of $Q_k$ equals the rank of the augmented matrix $(Q_k|G_k),$ solve (4.16) for the coefficients of the Laurent series. If $k = r_j,$ check that rank $Q_k = M - s_j,$ where $s_j$ is the algebraic multiplicity of the resonance $r_j$ in det$(Q_r) = 0.$

If rank $Q_k \neq$ rank$(Q_k|G_k),$ Gauss reduce the augmented matrix $(Q_k|G_k)$ to determine the compatibility condition. If all the compatibility conditions can be resolved by restricting the coefficients parameterizing (2.1), then (2.1) has the Painlevé property for those specific values. If any of the compatibility conditions cannot be resolved by restricting the coefficients parameterizing (2.1), then the Laurent series solution for this branch has a movable logarithmic branch point and the algorithm terminates.

For (4.3) with $\beta = 1,$ the principal branch

$$\begin{cases} 
\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = -1, \\
u_0(x,t) = -\bar{u}_0^{-1}(x,t)\{v_0(x,t)\bar{v}_0(x,t) - 2g_z^2(x,t)\}, \\
\bar{u}_0(x,t), v_0(x,t), \bar{v}_0(x,t) \text{ arbitrary,} \\
r_1 = -1, r_2 = r_3 = r_4 = 0, r_5 = r_6 = r_7 = 3, r_8 = 4,
\end{cases}$$

(4.17)

has three compatibility conditions at level $k = r_5 = r_6 = r_7 = 3.$ These compatibility conditions require that $a_x(x,t) = \bar{a}_x(x,t) = b_x(x,t) = \bar{b}_x(x,t)$ and $c_x(x,t) = d_x(x,t) = 0.$
At level $k = r_8 = 4$, the compatibility condition requires $d(t) = \bar{c}(t)$ and
\[
\{(a - \bar{a})^2 + 2i(a - \bar{a})h'(t)\}(2 + v_0 \bar{v}_0) - \{(b - \bar{b})^2 + 2i(b - \bar{b})h'(t)\}v_0 \bar{v}_0
\]
\[
+ 2i(a - \bar{a} + b - \bar{b})(v_0 \bar{v}_0)_t + (v_0)_t \bar{v}_0 + i(a_t - \bar{a}_t - b_t + \bar{b}_t)v_0 \bar{v}_0
\]
\[
+ 2i(a_t - a_t) - (a_{xx} + a_{xx} - \bar{b}_{xx})v_0 \bar{v}_0 - 2a_{xx} + 6\bar{a}_{xx} \equiv 0,
\]
where we have taken $g(x, t) = x - h(t)$. Careful inspection of (4.18) reveals that $a(x, t) = b(x, t)$. Setting $a(x, t) = b(x, t) = r(x, t) + is(x, t)$, where $r(x, t)$ and $s(x, t)$ are arbitrary real functions, (4.18) becomes
\[
2s^2(x, t) + s_t(x, t) + 2h'(t)s_x(x, t) - r_{xx}(x, t) + 2is_{xx}(x, t) \equiv 0.
\]
Since $h'(t)$ is arbitrary, it follows that $s_x(x, t) = 0$. Thus, $r_{xx}(x, t) = 2s^2(t) + s'(t)$ and upon integration
\[
r(x, t) = \frac{1}{2}\{2s^2(t) + s'(t)\}x^2 + r_1(t)x + r_2(t),
\]
where $r_1(t)$ and $r_2(t)$ are arbitrary functions.

Therefore, the generalized coupled NLS equations,
\[
iu_t + u_{xx} + (|u|^2 + |v|^2)u + \{s^2(t) + \frac{1}{2}s'(t)\}x^2 + r_1(t)x + r_2(t) + is(t) u = \bar{c}(t)v = 0,
\]
\[
i v_t + v_{xx} + (|u|^2 + |v|^2)v + \{s^2(t) + \frac{1}{2}s'(t)\}x^2 + r_1(t)x + r_2(t) + is(t) v = \bar{c}(t)u = 0,
\]
passes the Painlevé test, where $r_1(t), r_2(t)$, and $s(t)$ are arbitrary real functions and $c(t)$ is an arbitrary complex function.

When $\beta = 0$, the two compatibility conditions at level $k = r_5 = r_6 = 3$ require that
\[
c(x, t) = d(x, t) = \bar{c}(x, t) = \bar{d}(x, t) = 0.
\]
Similarly, the compatibility conditions at level $k = r_7 = r_8 = 4$, require that
\[
a(x, t) = \{s^2(t) - \frac{1}{2}s'(t)\}x^2 + r_1(t)x + r_2(t) + is(t),
\]
where $r_1(t), r_2(t)$ and $s(t)$ are arbitrary real functions. Therefore,
\[
iu_t + u_{xx} + |u|^2 u + \{s^2(t) - \frac{1}{2}s'(t)\}x^2 + r_1(t)x + r_2(t) + is(t) u = 0,
\]
passes the Painlevé test, a fact confirmed in [1].

5 Additional Examples

5.1 A peculiar ODE

Consider the ODE [34]
\[
u^2u'' - 3(u')^3 = 0.
\]
Substituting (3.1) into (5.1) gives $\alpha(\alpha + 2)(2\alpha - 1)\chi^3 g(z)^{3(\alpha - 1)} = 0$. So, both the terms in (5.1) have the same leading exponent, $3(\alpha - 1)$. Using the procedure in Section 4.1, in Step 5 the software automatically runs the test for the default values $\alpha = -3, -2,$ and $-1$. The choices $\alpha = -1$ and $-3$ are incompatible with the assumption $u_0 \neq 0$. The leading term vanishes for $\alpha = -2$ and $u_0$ is arbitrary. Substituting $u(z) = u_0 g^{-2}(z) + u_{11}g^{-2}(z)$, we find that $r_1 = -1, r_2 = 0,$ and $r_3 = 10$. Thus, the Laurent series solution of (5.1) is
\[
u(z) = u_0(z - z_0)^{-2} + u_0(z - z_0)^{8} + \cdots,
\]
where $z_0, u_0$ and $u_{10}$ are arbitrary constants. Hence, (5.1) passes the Painlevé test.
5.2 The sine-Gordon equation

Consider the sine-Gordon equation [1],
\[ u_{tt} + u_{xx} = \sin u. \]  
(5.3)

Using the transformation \( v(x, t) = e^{i u(x, t)} \), we obtain a polynomial differential equation
\[ vv_{tt} + vv_{xx} - v_t^2 - v_x^2 = \frac{1}{2} v(v^2 - 1). \]  
(5.4)

The dominant behavior of (5.4) is \( v(x, t) \sim 4(g_x^2(x, t) + g_t^2(x, t))g^{-2}(x, t) \), with resonances \( r_1 = -1 \) and \( r_2 = 2 \). The Laurent series solution of (5.4) is
\[ v = 4(g_x^2 + g_t^2)g^{-2} - 4(g_{xx} + g_{tt})g^{-1} + v_2 + \cdots, \]  
(5.5)

where \( g(x, t) \) and \( v_2(x, t) \) are arbitrary functions. The sine-Gordon equation passes the Painlevé test and is indeed completely integrable [1].

5.3 The cylindrical Korteweg-de Vries equation

Consider the generalized KdV equation,
\[ u_t + 6u u_x + u_{3x} + a(t)u = 0, \]  
(5.6)

where \( a(t) \) is an arbitrary function parameterizing the equation. The dominant behavior of (5.6) is \( u(x, t) \sim -2g_x^2(x, t)g^{-2}(x, t) \), with resonances \( r_1 = -1, r_2 = 4 \) and \( r_3 = 6 \). At level \( k = r_3 = 6 \), we obtain the compatibility condition
\[ \frac{2a(t)^2 + a'(t)}{6g_x(x, t)} = 0. \]  
(5.7)

So, (5.6) passes the Painlevé test if \( a(t) = \frac{1}{2t} \). In this case, (5.6) reduces to the cylindrical KdV, which is completely integrable as confirmed by other analyses [1].

5.4 A fifth-order generalized Korteweg-de Vries equation

Consider the generalized fifth-order KdV equation,
\[ u_t + au_x u_{xx} + bu u_{3x} + cu^2 u_x + u_{5x} = 0, \]  
(5.8)

with constant parameters \( a, b, \) and \( c \). The dominant behavior of (5.8) is
\[ u(x, t) \sim -\frac{3g_x^2(x, t)}{c} \left\{ (a + 2b) \pm \sqrt{a^2 + 4ab + 4b^2 - 40c} \right\} g^{-2}(x, t). \]  
(5.9)

The resonances are the roots of
\[ \det(Q_r) = -c(r - 6)(r + 1) \left( 3 \sqrt{(a + 2b)^2 - 40c(2a - b(r - 4))} \right. 
\[ - 6(a + 2b)^2 + 240c + (3b(a + 2b) - 86c)r + 15cr^2 - cr^3 \right) g_x^5. \]  
(5.10)
Symbolic Software for the Painlevé Test

Determining what values of \( a, b, \) and \( c \) that lead to integer roots of (5.10) is difficult by hand or with a computer. An investigation of the scaling properties of (5.8) reveals that only the ratios \( a/b \) and \( c/b^2 \) are important. Let us consider the well-known special cases.

If we take \( a = b = 5c = b^2 \), then (5.8) passes the Painlevé test with resonances \( r_1 = -2, r_2 = -1, r_3 = 5, r_4 = 6, r_5 = 12 \) and \( r_1 = -1, r_2 = 2, r_3 = 3, r_4 = 6, r_5 = 10 \). Taking \( b = 5 \), equation (5.8) becomes the completely integrable equation

\[
    u_t + 5u_xu_{xx} + 5uu_{3x} + 5u^2u_x + u_{5x} = 0, \tag{5.11}
\]
due to Sawada and Kotera [36] and Caudrey et al. [5].

If we take \( a = 2b \) and \( 10c = 3b^2 \), then (5.8) passes the Painlevé test with resonances \( r_1 = -3, r_2 = -1, r_3 = 6, r_4 = 8, r_5 = 10 \) and \( r_1 = -1, r_2 = 2, r_3 = 5, r_4 = 6, r_5 = 8 \). For \( b = 10 \), equation (5.8) is a member of the completely integrable KdV hierarchy

\[
    u_t + 10uu_{3x} + 20u_xu_{xx} + 30u^2u_x + u_{5x} = 0, \tag{5.12}
\]
due to Lax [26].

If we take \( 2a = 5b \) and \( 5c = b^2 \), then (5.8) passes the Painlevé test with resonances \( r_1 = -7, r_2 = -1, r_3 = 6, r_4 = 10, r_5 = 12 \) and \( r_1 = -1, r_2 = 3, r_3 = 5, r_4 = 6, r_5 = 7 \). When \( b = 10 \), equation (5.8) is the Kaup-Kupershmidt equation [12,17],

\[
    u_t + 10uu_{3x} + 20u_x^2u_x + 25uu_{xxx} + u_{5x} = 0, \tag{5.13}
\]
which is also known to be completely integrable.

While there are many other values for \( a, b, \) and \( c \), for which (5.10) only has integer roots, but compatibility conditions prevent (5.8) from having the Painlevé property. For instance, when \( a = 2b \) and \( 5c = 2b^2 \), the resonances are \( r_1 = -1, r_2 = 0, r_3 = 6, r_4 = 7, r_5 = 8 \). At level \( k = r_2 = 0 \), we are forced to take \( u_0(x, t) = -30g^3_0(x, t)/b \), so the Laurent series solution is not the general solution and (5.8) fails the Painlevé test. Similarly, when \( 7a = 19b \) and \( 49c = 9b^2 \), we have resonances \( r_1 = -1, r_2 = 3 \) and \( r_3 = r_4 = r_5 = 6 \), so the Laurent series solution is not the general solution and, again (5.8) fails the Painlevé test.

6 Brief Review of Symbolic Algorithms and Software

There is a variety of methods for testing nonlinear ODEs and PDEs for the Painlevé property. While the WTC algorithm discussed in this paper is the most common method used in Painlevé analysis, it is not appropriate in all cases. For instance, there are numerous completely integrable differential equations which have algebraic branching in their series solutions; a property that is allowed by the so-called “weak” Painlevé test (see [13,32,33]). A more thorough approach for testing differential equations with branch points is the poly-Painlevé test (see [22,23]). The perturbative Painlevé test [9] was developed to check the compatibility conditions of negative resonances other than \( r = -1 \).

For testing ODEs, there are several implementations: ODEPAINLEVE developed by Rand and Winternitz [34] in Macsyma is restricted to scalar differential equations; PTEST.RED by Renner in Reduce [35]; and, a Reduce package by Scheen [37] which implements both the traditional and the perturbative Painlevé tests. For testing PDEs, there are a few implementations. The package PAINMATH.M by Hereman et al. [16] is unable to find all the
dominant behaviors in systems with undetermined $\alpha_i$ and is limited to two independent variables.

Only the Maple package PDEPtest by Xu and Li [45–47] is comparable to our package PainleveTest.m [4]. The package PDEPtest was written after our package and allows the testing of systems of PDEs (but not ODEs) parameterized by arbitrary functions using either the traditional WTC algorithm or the simplification proposed by Kruskal (see Section 3). While PDEPtest can find all the dominant behaviors in some systems with undetermined $\alpha_i$ (such as the Hirota-Satsuma system), it fails to find the dominant behaviors for systems in which more than one $\alpha_i$ is undetermined (such as the NLS equation, $iu_t + u_{zz} + 2u|u|^2 = 0$, which is completely integrable [1]). Furthermore, PDEPtest requires that all the $\alpha_i$ are negative, a weakness of the implementation, since it is standard to allow some positive exponents (see equation (2.4) in [33] with leading exponents $-1$ and 1).

7 Using the Software Package PainleveTest.m

The package PainleveTest.m has been tested on both PCs and UNIX work stations with Mathematica versions 3.0, 4.0, 4.1, 5.0, 5.1, and 6.0 using a test set of over 50 PDEs and two dozen ODEs. The Backus-Naur form of the function is

\[
\langle \text{Main Function} \rangle \to \text{PainleveTest}[(\text{Equations}), (\text{Functions}), (\text{Variables}), (\text{Options})]
\]

\[
\langle \text{Options} \rangle \to \text{Verbose} \to (\text{Boolean}) | \text{KruskalSimplification} \to (\text{Variable}) | \text{DominantBehaviorMin} \to (\text{Negative Integer}) | \text{DominantBehaviorMax} \to (\text{Integer}) | \text{DominantBehavior} \to (\text{List of Rules}) | \text{DominantBehaviorConstraints} \to (\text{List of Constraints}) | \text{DominantBehaviorVerbose} \to (\text{Range}) | \text{ResonancesVerbose} \to (\text{Range}) | \text{ConstantsOfIntegrationVerbose} \to (\text{Range})
\]

\[
\langle \text{Bool} \rangle \to \text{True} | \text{False}
\]

\[
\langle \text{Range} \rangle \to 0 | 1 | 2 | 3
\]

\[
\langle \text{List of Rules} \rangle \to \{\{\text{alpha}[1] \to (\text{Integer}), \text{alpha}[2] \to (\text{Integer}), \ldots\}, \ldots\}
\]

\[
\langle \text{List of Constraints} \rangle \to \{\text{alpha}[1] == \text{alpha}[2], \ldots\}
\]

The output of the function is

\[
\{\{\{\text{Dominant behavior}\}, \{\text{Resonances}\}, \\
\{\{\text{Laurent series coefficients}\}, \{\text{Compatibility conditions}\}\}, \ldots\}\}
\]

If using a PC, place the package PainleveTest.m in a directory, say myDirectory on drive C. Start a Mathematica notebook session and execute the commands:

\[
\text{In}[1] = \text{SetDirectory}["c:\myDirectory"] ; \quad (\ast \text{Specify the directory } \ast)
\]
Symbolic Software for the Painlevé Test

In[2] = Get["PainleveTest.m"] (* Read in the package *)

In[3] = PainleveTest[
   (* Test the KdV equation *)
   {D[u[x,t],t]+6*u[x,t]*D[u[x,t],x]+D[u[x,t],{x,3}] == 0},
   u[x, t], {x,t}, KruskalSimplification -> x]

Out[3] =

\[
\left\{ \begin{array}{l}
\{ \alpha_1 \to -2 \}, \{ r \to -1, r \to 4, r \to 6 \}, \\
\{ u_{1,0} \to -2, u_{1,1} \to 0, u_{1,2} \to \frac{h'(t)}{6}, u_{1,3} \to 0, \\
u_{1,4} \to C_1(t), u_{1,5} \to \frac{h''(t)}{36}, u_{1,6} \to C_2(t) \} \end{array} \right\}
\]

The option KruskalSimplification \(\to x\) allows one to use \(g(x,t) = x - h(t)\) in the calculation of the constants of integration and in checking the compatibility conditions.

In[4] = PainleveTest[
   (* Eq. (2.4) in Ramani et al. [32] *)
   {D[x[z], z] == x[z]*(a - x[z] - y[z]),
    D[y[z], z] == y[z]*(x[z] - 1)},
   {x[z], y[z]}, {z}, DominantBehaviorMax -> 1 ]

Out[4] =

\[
\left\{ \begin{array}{l}
\{ \alpha_1 \to -1, \alpha_2 \to -1 \}, \{ r \to -1, r \to 2 \}, \{ \{ u_{1,0} \to -1 \}, \{ a + 1 \to 0 \} \}, \\
\{ \alpha_1 \to -1, \alpha_2 \to 1 \}, \{ r \to -1, r \to 0 \}, \{ \{ u_{1,0} \to 1, u_{2,0} \to C_1 \} \} \end{array} \right\}
\]

In this example, if the DominantBehaviorMax option was not used, we would wrongly conclude that the system only passes the Painlevé test when \(a = -1\). However, by allowing positive \(\alpha_i\), we find the second branch \(\alpha_1 = -1\) and \(\alpha_2 = 1\), for which the system passes the Painlevé test without restricting the value of the parameter \(a\). Alternatively, executing

In[5] = PainleveTest[
   { D[x[z], z] == x[z]*(a - x[z] - y[z]),
    D[y[z], z] == y[z]*(x[z] - 1)},
   {x[z], y[z]}, {z},
   DominantBehavior -> \{\{\alpha[1] \to -1, \alpha[2] \to 1\}\} ]

would only test the branch with \(\alpha_1 = -1\) and \(\alpha_2 = 1\). For an example of the option DominantBehaviorConstraints, see Step 4 of Algorithm 4.1.

The option Verbose \(\to \) True gives a brief trace of the calculations in each of the three steps of the algorithm. The options DominantBehaviorVerbose, ResonancesVerbose, and ConstantsOfIntegrationVerbose allow for a more detailed trace of the calculation. For instance, DominantBehaviorVerbose \(\to 1\) would show the result of substituting the ansatz, the exponents before and after removing non-dominant powers, etc. While DominantBehaviorVerbose \(\to 3\) shows the result of nearly every line of code in the package, allowing the user to check the results in the trickiest cases.
8 Discussion and Conclusions

Our software package PainleveTest.m is applicable to polynomial systems of nonlinear ODEs and PDEs. While the Painlevé test does not guarantee complete integrability, it helps in identifying candidate differential equations for complete integrability in a straightforward manner. For differential equations with parameters (including arbitrary functions of the independent variables), our software allows the user to determine the conditions under which the differential equations may possess the Painlevé property. Therefore, by finding the compatibility conditions, classes of parameterized differential equations can be analyzed and candidates for complete integrability can be identified.

The difficulty in completely automating the Painlevé test lies in determining the dominant behaviors of the Laurent series solutions; specifically, determining all the valid dominant behaviors when one or more of the \( \alpha_i \) are undetermined. While there are other implementations for the Painlevé test, ours is currently the only implementation in Mathematica which allows the testing of polynomial systems of nonlinear PDEs with no limitations on the number of differential equations or the number of independent variables (except where limited by memory).

References


