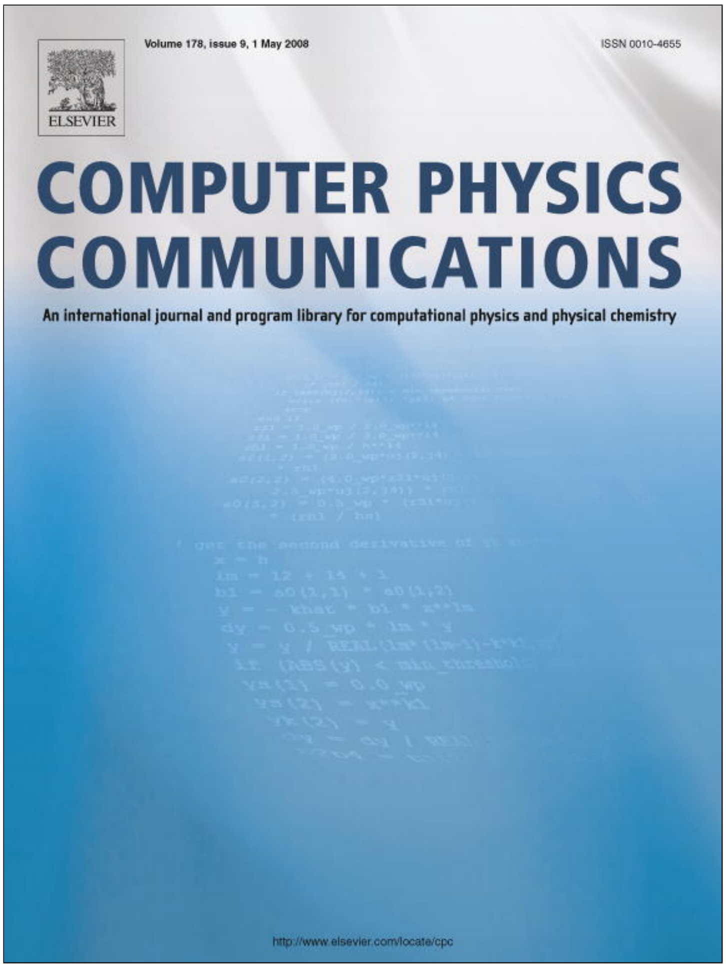
Volume 178, issue 9, 1 May 2008
ISSN 0010-4655
ELSEVIER
COMPUTER PHYSICS COMMUNICATIONS
An international journal and program library for computational physics and physical chemistry
http://www.elsevier.com/locate/cpc

Available online at www.sciencedirect.com
ScienceDirect
Computer Physics Communications 178 (2008) 700–712
Computer Physics Communications
www.elsevier.com/locate/cpc

Automatic computation of the travelling wave solutions to nonlinear PDEs ☆

Songxin Liang *, David J. Jeffrey

Department of Applied Mathematics, University of Western Ontario, London, Ontario, Canada, N6A 5B7

Received 1 June 2007; received in revised form 21 November 2007; accepted 11 December 2007

Available online 23 December 2007

Abstract

Various extensions of the tanh-function method and their implementations for finding explicit travelling wave solutions to nonlinear partial differential equations (PDEs) have been reported in the literature. However, some solutions are often missed by these packages. In this paper, a new algorithm and its implementation called TWS for solving single nonlinear PDEs are presented. TWS is implemented in MAPLE 10. It turns out that, for PDEs whose balancing numbers are not positive integers, TWS works much better than existing packages. Furthermore, TWS obtains more solutions than existing packages for most cases.

Program summary

Program title: TWS
Catalogue identifier: AEAM_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEAM_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 1250
No. of bytes in distributed program, including test data, etc.: 78 101
Distribution format: tar.gz
Programming language: Maple 10
Computer: A laptop with 1.6 GHz Pentium CPU
Operating system: Windows XP Professional
RAM: 760 Mbytes
Classification: 5
Nature of problem: Finding the travelling wave solutions to single nonlinear PDEs.
Solution method: Based on tanh-function method.
Restrictions: The current version of this package can only deal with single autonomous PDEs or ODEs, not systems of PDEs or ODEs. However, the PDEs can have any finite number of independent space variables in addition to time t.
Unusual features: For PDEs whose balancing numbers are not positive integers, TWS works much better than existing packages. Furthermore, TWS obtains more solutions than existing packages for most cases.
Additional comments: It is easy to use.
Running time: Less than 20 seconds for most cases, between 20 to 100 seconds for some cases, over 100 seconds for few cases.
References:

[1] E.S. Cheb-Terrab, K. von Bulow, Comput. Phys. Comm. 90 (1995) 102.
[2] S.A. Elwakil, S.K. El-Labany, M.A. Zahran, R. Sabry, Phys. Lett. A 299 (2002) 179.
[3] E. Fan, Phys. Lett. 277 (2000) 212.
[4] W. Malfliet, Amer. J. Phys. 60 (1992) 650.

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: sliang22@uwo.ca (S. Liang).

doi:10.1016/j.cpc.2007.12.004

[5] W. Malfliet, W. Hereman, Phys. Scripta 54 (1996) 563.
[6] E.J. Parkes, B.R. Duffy, Comput. Phys. Comm. 98 (1996) 288.

PACS: 02.30.Jr; 02.70.Wz; 01.50.Ff

Keywords: Nonlinear partial differential equation; Travelling wave solution; Tanh-function method; Symbolic computation; Exact solution

1. Introduction

Mathematical modelling of physical systems often leads to nonlinear partial differential equations (PDEs). Explicit solutions, especially travelling wave solutions, to such equations are of fundamental importance. In order to help physicists and engineers better understand the mechanisms that govern these physical phenomena, many powerful and direct methods have been proposed. Among these are direct integration (whenever possible), Hirota's bilinear method [4], the inverse scattering transform [4], Painlevé expansion method [2] and the real exponential method [12,13].

A less sophisticated but more direct method, the tanh-function method, was proposed to find explicit travelling wave solutions to nonlinear PDEs. This method was due to Malfliet and Hereman [17,18]. Since then, a lot of relative contributions have been reported in the literature, for example, [5,10,11,19,20].

However, only tanh type travelling wave solutions can be obtained by the tanh-function method. If a given PDE has other types of travelling wave solutions, for example, tan type or rational type solutions, we have to repeat the similar but tedious calculations. With this consideration in mind, Fan [8,9] proposed an extended tanh-function method by which one can simultaneously obtain three types of travelling wave solutions to a given PDE. The key idea of the method is to take advantage of the Riccati equation $w' = b + w^2$ and use its solutions to replace $\tanh(kz)$ according to the sign of b. Later, based on Fan's method, Elwakil and his coworkers proposed a modified extended tanh-function method [6,7].

In principle, the tanh-function method is more straightforward than the real exponential method. However, for other than simple equations it is still tedious to use by hand. If we take full advantage of modern computer algebra systems such as MAPLE and MATHEMATICA, the limitation of the tanh-function method mentioned above would be eliminated.

Based on the tanh-function method, Parkes and Duffy developed a MATHEMATICA package called ATFM which can deal with tedious algebraic computations and output directly the travelling wave solutions to given PDEs [19]. However, ATFM can only carry out some (but not all) steps of the method.

A much more comprehensive MATHEMATICA package called PDESPECIALSOLUTIONS was developed by Hereman and his students in a three-year period starting in 1999 [1]. PDESPECIALSOLUTIONS can deal with not only single PDEs or ordinary differential equations (ODEs) but also systems of PDEs or ODEs. It can perform the computations automatically from start to end without human intervention. However, PDESPECIALSOLUTIONS can only computes the tanh type, sech type, the mixed tanh-sech type, and the Jacobi's CN and SN type travelling wave solutions.

There is a built-in package called TWSolutions (actually it is one of the commands in the package *PDEtools*) in MAPLE 10 for solving the travelling wave solutions to given PDEs. It is based on the work by Cheb-Terrab and von Bulow [3]. TWSolutions can find different types of solutions according to a list of pre-selected functions instead of just tanh. It works for systems of PDEs or ODEs as well as single PDEs or ODEs.

Other Maple packages worthy of mention are those developed by Li and Liu [14–16]. The package RATH automates the tanh-function method for single PDEs [14]. The package AJFM automates the Jacobi elliptic function method for single PDEs [16]. Later, based on these packages, they developed a much more comprehensive package called RAEEM [15] which, as they claimed, can obtain solutions of polynomial, rational, exponential, triangular, hyperbolic, Jacobi elliptic and Weierstrass elliptic types. Furthermore, RAEEM also works for systems of PDEs as well as single PDEs.

However, we notice that, these packages do not work well when the balancing numbers of the given PDEs are not positive integers. Moreover, some types of solutions are often missed by these packages. These observations motivate us to propose a new algorithm to overcome these weaknesses. A new package called TWS based on the new algorithm has been implemented in MAPLE 10. Instead of extending the scope of problems, on the contrary, we confine ourselves to single PDEs and ODEs. In this way, we can focus on refining existing algorithms and implementing the new algorithm carefully. The new package TWS has been used to solve almost one hundred PDEs and ODEs on MAPLE 10. It turns out that, for PDEs whose balancing numbers are not positive integers, TWS works much better than the Maple packages RAEEM and TWSolutions. Furthermore, for most cases, TWS obtains more solutions than RAEEM and TWSolutions.

This paper is organized as follows. In Section 2, the main steps of the algorithm are presented and discussed. In Section 3, examples are shown to demonstrate the advantages of the package in comparison with other Maple packages. In Section 4, the limitation of the package and our future plan are mentioned.

2. Algorithms

As mentioned in Section 1, the current version of the package can only work for single PDEs or ODEs, not for systems of PDEs. Furthermore, every PDE must be a polynomial in a function (for example, $u(x_1, \ldots, x_k, t)$) and its partial derivatives, and must be autonomous. On the other hand, the function may have any finite number of independent space variables in addition to time t.

Suppose we are given such a PDE (denoted by *pde*):

$$H(u, u_t, u_{x_1}, u_{x_2}, \ldots, u_{x_1x_1}, u_{x_1x_2}, \ldots) = 0 \tag{1}$$

for a function $u(x_1, \ldots, x_k, t)$. Like the package TWSolutions, we select a list of functions instead of just tanh for finding travelling wave solutions. The functions we choose are (denoted by *funclist*): [rational, exp, sinh, csch, cosh, sech, tanh, coth, sin, csc, cos, sec, tan, cot, JacobiCN, JacobiSN].

Now, for every function f in *funclist* and the balancing number m (we will explain it in the main algorithm below), we have a subalgorithm called *TWSsolve* to compute all the travelling wave solutions of f type, where *plist* is some list of the parameters occurred in *pde*, and *real* is a boolean value which determines the output of the solutions. If *real* is *true* then only real solutions are output, otherwise, all real and complex solutions are obtained.

Algorithm 1. TWSsolve.
Input: *pde*, m, f, *plist*, *real*.
Output: The travelling wave solutions to *pde* with respect to f.
Procedure:

S1 Substitute $u(x_1, \ldots, x_k, t) = U(\eta)$ where

$$\eta = \sum_{i=1}^{k} \lambda_i x_i + \lambda_{k+1} t + \delta \tag{2}$$

into *pde* to get an ODE *ode* with dependent variable $U(\eta)$. If the input equation is an ODE with dependent variable $u(x)$, then (2) becomes $\eta = \lambda x + \delta$.

S2 Substitute

$$U(\eta) = \sum_{i=-m}^{m} a_i T^i, \tag{3}$$

where $T = f(\eta)$ into *ode* and eliminate the common denominator to get an equation. Instead of using traditional polynomial form, we use this symmetric one (3) which comes from [6] to obtain more solutions.

In order to see what the resulting equation looks like, we list the first-order and/or second-order derivatives of the functions in *funclist* as follows.

$$\left(\frac{1}{\eta}\right)' = -\left(\frac{1}{\eta}\right)^2;$$
$$\left(e^{\eta}\right)' = e^{\eta},$$
$$\tanh(\eta)' = 1 - \tanh(\eta)^2,$$
$$\coth(\eta)' = 1 - \coth(\eta)^2,$$
$$\tan(\eta)' = 1 + \tan(\eta)^2,$$
$$\cot(\eta)' = -1 - \cot(\eta)^2,$$
$$\sinh(\eta)' = \sqrt{1 + \sinh(\eta)^2}, \qquad \sinh(\eta)'' = \sinh(\eta),$$
$$\cosh(\eta)' = \sqrt{\cosh(\eta)^2 - 1}, \qquad \cosh(\eta)'' = \cosh(\eta),$$
$$\sin(\eta)' = \sqrt{1 - \sin(\eta)^2}, \qquad \sin(\eta)'' = -\sin(\eta),$$
$$\cos(\eta)' = -\sqrt{1 - \cos(\eta)^2}, \qquad \cos(\eta)'' = -\cos(\eta),$$
$$\operatorname{csch}(\eta)' = -\operatorname{csch}(\eta)\sqrt{1 + \operatorname{csch}(\eta)^2}, \qquad \operatorname{csch}(\eta)'' = \operatorname{csch}(\eta) + 2\operatorname{csch}(\eta)^3,$$
$$\operatorname{sech}(\eta)' = -\operatorname{sech}(\eta)\sqrt{1 - \operatorname{sech}(\eta)^2}, \qquad \operatorname{sech}(\eta)'' = \operatorname{sech}(\eta) - 2\operatorname{sech}(\eta)^3,$$

$$\csc(\eta)' = -\csc(\eta)\sqrt{\csc(\eta)^2 - 1}, \qquad \csc(\eta)'' = 2\csc(\eta)^3 - \csc(\eta),$$
$$\sec(\eta)' = \sec(\eta)\sqrt{\sec(\eta)^2 - 1}, \qquad \sec(\eta)'' = 2\sec(\eta)^3 - \sec(\eta),$$
$$\mathrm{JacobiCN}(\eta,\omega)' = -\sqrt{(1 - \omega^2 + \omega^2\,\mathrm{JacobiCN}(\eta,\omega)^2)(1 - \mathrm{JacobiCN}(\eta,\omega)^2)},$$
$$\mathrm{JacobiCN}(\eta,\omega)'' = (2\omega^2 - 1)\,\mathrm{JacobiCN}(\eta,\omega) - 2\omega^2\,\mathrm{JacobiCN}(\eta,\omega)^3,$$
$$\mathrm{JacobiSN}(\eta,\omega)' = \sqrt{(1 - \omega^2\,\mathrm{JacobiSN}(\eta,\omega)^2)(1 - \mathrm{JacobiSN}(\eta,\omega)^2)},$$
$$\mathrm{JacobiSN}(\eta,\omega)'' = 2\omega^2\,\mathrm{JacobiSN}(\eta,\omega)^3 - (\omega^2 + 1)\,\mathrm{JacobiSN}(\eta,\omega),$$

where ω is the modulus and $0 \leqslant \omega \leqslant 1$.

From the formulas above, we can see that for every function f in *funclist*, any order derivative of $f(\eta)$ with respect to η is a polynomial in $f(\eta)$, or of the form $\Phi\sqrt{\Gamma}$ where Φ and Γ are polynomials in $f(\eta)$. Therefore, after substituting (3) into *ode* and eliminating the common denominator which is a power of $f(\eta)$, we obtain the desired equation of the form

$$\Phi + \Psi\sqrt{\Gamma} = 0, \tag{4}$$

where Φ, Ψ and Γ are polynomials in $f(\eta)$. In case f is one of the functions: rational, exp, tanh, coth, tan, cot, then $\Psi \equiv 0$ which means that the resulting equation is a polynomial equation in $f(\eta)$.

S3 Set all the coefficients of the different powers of T in Φ and Ψ of (4) to zero, and consequently get a system of polynomial equations whose variables include a_i, a_{-i} $(i = 1, \ldots, m)$, λ_j $(j = 1, \ldots, k+1)$, δ, and the parameters in *plist*.

S4 Solve the system of polynomial equations with respect to the variables using the Maple command *solve*. For the order of the variables, we sort the variables in ascending order according to their degrees in the system. It turns out that the Maple solver is very powerful. For polynomial equations, it outputs all the solutions.

S5 If *real* = *true*, keep the real solutions only.

S6 Substitute the solutions obtained into $U(\eta) = \sum_{i=-m}^{i=m} a_i T^i$ one by one to get the travelling wave solutions of f type.

Based on Algorithm 1, the main algorithm now is:

Algorithm 2. TWS.
Input:
pde: a PDE as described in (1);
flist: a list of functions. The default value is *funclist*;
plist: a list of parameters. The default value is the empty list;
real: a boolean value. The default value is *true*.
Output: The travelling wave solutions to *pde*.
Procedure:

Set *Soln* = { }. For each f in *flist* repeat

M1 Find the balancing number m of *pde* with respect to f. As in step S1 of Algorithm 1, we change *pde* into *ode* with dependent variable $U(\eta)$. Then we need to determine the degree of each term in *ode* with respect to $T = f(\eta)$ when substituting (3) into *ode* (but actual substitution is not necessary).
According to the list of formulas in step S2 of Algorithm 1, we can see that if f is one of the functions: rational, tanh, coth, tan, cot, csch, sech, sec, csc, JacobiCN and JacobiSN, then the degree of $\mathrm{d}^p U(\eta)/\mathrm{d}\eta^p$ with respect to T is $m + p$; if f is one of the functions: exp, sinh, cosh, sin, cos, then the degree of $\mathrm{d}^p U(\eta)/\mathrm{d}\eta^p$ with respect to T is m, and the degree of $U(\eta)^q$ with respect to T is qm. Based on these facts above, each term in *ode* has a degree with respect to T in the form of $cm + d$, and consequently we obtain a list of degrees for *ode*. Find the degree with maximum value of c and the degree with maximum value of d. Then by equating them we can solve the balancing number m.
As an example, let *pde* be $u_t - u_{xx} - u + u^2 = 0$ and f be tanh. Then the list of degrees is $[m+1, m+2, m, 2m]$. So from $m + 2 = 2m$, we get $m = 2$.

M2 If $m \in \mathbb{N}$, then set *Soln* = *Soln* $\cup$ TWSsolve(*pde*, *m*, *f*, *plist*, *real*).

M3 If $m \notin \mathbb{N}$ and $m \neq 0$, then substitute $u = v^m$ into *pde* to get another PDE *pde*2. It is easy to prove that the balancing number *m*2 of *pde*2 is a positive integer. *m*2 can be computed by step M1. Set *Soln*2 = TWSsolve(*pde*2, *m*2, *f*, *paramset*, *real*). Then raise every entry in *Soln*2 to the power of m. Finally, set *Soln* = *Soln* $\cup$ *Soln*2.

3. Examples

The algorithm in Section 2 has been programmed into a MAPLE package. The command for loading it to a Maple session is (suppose it is stored in C:\pde\TWS.mpl)

```
> read "C:/pde/TWS.mpl"
```

The calling sequence is: TWS(*pde*, function = *flist*, parameter = *plist*, *real*), where

- *pde*—a PDE as described in (1);
- *flist*—(optional) a list of functions to be used for finding the travelling wave solutions to *pde*, and the default value is: [rational, exp, sinh, csch, cosh, sech, tanh, coth, sin, csc, cos, sec, tan, cot, JacobiCN, JacobiSN];
- *plist*—(optional) a list of parameters occurred in *pde*, and the default value is the empty list;
- *real*—(optional) a boolean value which determines the output of the solutions: if it is *true* (the default value), only real solutions are returned, otherwise, all solutions are returned.

Before comparing the new package with other existing packages, we mention a key point when using it. If the given PDE *pde* contains parameters, and *pde* has travelling wave solutions only for special parameter values, then these parameters must be specified by the argument parameter = *plist* in the calling sequence above. Otherwise, the new package will treat these parameters as arbitrary parameters that allow any values. Consequently no nontrivial solutions would be returned by the package.

As an example, let *pde* be a generalized Kuramoto–Sivashinsky equation [19] $u_t + uu_x - u_{xx} + \alpha u_{xxx} + u_{xxxx} = 0$. If we run the following Maple command, then no nontrivial solutions are obtained.

```
> TWS(pde, function=[tanh]);
```

But if we run the following Maple command specifying the parameter α, then 3 tanh/coth type nontrivial solutions are returned for the special value $\alpha = 0$. Notice that, in the following examples, the outputs of the solutions have been edited in order to give a more pleasing layout.

```
> TWS(pde, function=[tanh],parameter=[alpha]);
```

- $\alpha = 0$, $u(x,t) = a_0 - \frac{45}{361}\sqrt{19}\tanh\eta + \frac{15}{361}\sqrt{19}(\tanh\eta)^3$, where $\eta = \frac{\sqrt{19}}{38}x - \frac{\sqrt{19}}{38}a_0 t + \delta$.
- $\alpha = 0$, $u(x,t) = \frac{15}{361}\frac{\sqrt{19}}{(\tanh\eta)^3} - \frac{45}{361}\frac{\sqrt{19}}{\tanh\eta} + a_0$, where $\eta = \frac{\sqrt{19}}{38}x - \frac{\sqrt{19}}{38}a_0 t + \delta$.
- $\alpha = 0$, $u(x,t) = \frac{15}{2888}\sqrt{19}(\tanh\eta)^{-3} - \frac{135}{2888}\sqrt{19}(\tanh\eta)^{-1} + a_0 - \frac{135}{2888}\sqrt{19}\tanh\eta + \frac{15}{2888}\sqrt{19}(\tanh\eta)^3$, where $\eta = \frac{\sqrt{19}}{76}x - \frac{\sqrt{19}}{76}a_0 t + \delta$.

Now we start to compare the new package TWS with other existing packages. Because MATHEMATICA is not available to us, we mainly compare TWS with the Maple packages TWSolutions [3] and RAEEM [15].

Example 1. (See [19].) A generalized Fisher equation $u_t - u_{xx} - u^{-1}u_x^2 = u(1-u^3)$.

This is a PDE whose balancing number is not a positive integer. It turns out that the new package TWS obtains 15 nontrivial solutions as follows.

```
> pde1:=diff(u(x,t),t)-diff(u(x, t),x$2)-diff(u(x,t),x)^2/u(x,t)
       =u(x,t)*(1-u(x,t))^3
```

$$pde1 := \frac{\partial}{\partial t}u(x,t) - \frac{\partial^2}{\partial x^2}u(x,t) - \frac{(\frac{\partial}{\partial x}u(x,t))^2}{u(x,t)} = u(x,t)\big(1 - \big(u(x,t)\big)^3\big)$$

```
> TWS(pde1);
```

- 1 tan/cot type solution:

$$u(x,t) = \frac{\sqrt[3]{28}}{4}\left(\frac{1 + (\tan(\frac{3}{8}\sqrt{2}x + \delta))^2}{\tan(\frac{3}{8}\sqrt{2}x + \delta)}\right)^{2/3}.$$

- 1 cos/sec type solution:

$$u(x,t) = \frac{\sqrt[3]{14}}{2}\left(\left(\cos\left(\frac{3}{4}\sqrt{2}x + \delta\right)\right)^{-1}\right)^{2/3}.$$

- 1 sin/csc type solution:

$$u(x,t)=\frac{\sqrt[3]{14}}{2}\left(\left(\sin\left(\frac{3}{4}\sqrt{2}x+\delta\right)\right)^{-1}\right)^{2/3}.$$

- 12 tanh/coth type solutions:

$$u(x,t)=\frac{\sqrt[3]{4}}{4}\left(-\frac{(-1+\tanh(\frac{3}{56}\sqrt{14}x-\frac{33}{56}t+\delta))^2}{\tanh(\frac{3}{56}\sqrt{14}x-\frac{33}{56}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{\sqrt[3]{2}}{2}\left(\frac{-1+\tanh(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta)}{\tanh(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(4+4\tanh\left(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta\right)\right)^{2/3},$$

$$u(x,t)=\frac{\sqrt[3]{2}}{2}\left(\frac{1+\tanh(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta)}{\tanh(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(\frac{4-4\tanh(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta)}{\tanh(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(-4-4\tanh\left(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta\right)\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(4-4\tanh\left(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta\right)\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(\frac{-4-4\tanh(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta)}{\tanh(\frac{3}{28}\sqrt{14}x+\frac{33}{28}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{1}{4}\left(-4+4\tanh\left(\frac{3}{28}\sqrt{14}x-\frac{33}{28}t+\delta\right)\right)^{2/3},$$

$$u(x,t)=\frac{\sqrt[3]{4}}{4}\left(\frac{(-1+\tanh(\frac{3}{56}\sqrt{14}x-\frac{33}{56}t+\delta))^2}{\tanh(\frac{3}{56}\sqrt{14}x-\frac{33}{56}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{\sqrt[3]{4}}{4}\left(-\frac{(1+\tanh(\frac{3}{56}\sqrt{14}x+\frac{33}{56}t+\delta))^2}{\tanh(\frac{3}{56}\sqrt{14}x+\frac{33}{56}t+\delta)}\right)^{2/3},$$

$$u(x,t)=\frac{\sqrt[3]{4}}{4}\left(\frac{(1+\tanh(\frac{3}{56}\sqrt{14}x+\frac{33}{56}t+\delta))^2}{\tanh(\frac{3}{56}\sqrt{14}x+\frac{33}{56}t+\delta)}\right)^{2/3}.$$

On the other hand, the package RAEEM obtains no solutions, while the package TWSolutions returns only trivial solutions.

```
> functionlist := [JacobiCN, JacobiDN, JacobiNC, JacobiND, JacobiNS,
JacobiSN, WeierstrassP, arcsinh, cos, cosh, cot, coth, csc, csch,
exp, identity, ln, sec, sech, sin, sinh, tan, tanh]:

> TWSolutions(pde1, function=functionlist);
```

$$u(x,t)=1,$$

$$u(x,t)=-\frac{1}{2}+\frac{1}{2}i\sqrt{3},$$

$$u(x,t)=-\frac{1}{2}-\frac{1}{2}i\sqrt{3}.$$

Example 2. (See [6].) A nonlinear reaction-diffusion equation $u_t-(u^2)_{xx}-pu+qu^2=0$ where $q>0$.

This is another PDE whose balancing number is not a positive integer. The new package TWS obtains six tanh/coth type solutions when the argument function $=$ *flist* is specified.

```
> pde2:=diff(u(x,t),t)-(diff(u(x,t)^2,x$2)-p*u(x,t)+q*u(x,t)^2=0;
```

$$pde2 := \frac{\partial}{\partial t}u(x,t) - 2\left(\frac{\partial}{\partial x}u(x,t)\right)^2 - 2\left(\frac{\partial^2}{\partial x^2}u(x,t)\right)u(x,t) - pu(x,t) + q\big(u(x,t)\big)^2 = 0$$

```
> TWS(pde2,function=[tanh]);
```

$$u(x,t) = \frac{2p\tanh(\frac{1}{4}\sqrt{q}x - \frac{1}{4}pt + \delta)}{q(-1+\tanh(\frac{1}{4}\sqrt{q}x - \frac{1}{4}pt + \delta))},$$
$$u(x,t) = \frac{-4p\tanh(\frac{1}{8}\sqrt{q}x - \frac{1}{8}pt + \delta)}{q(1 - 2\tanh(\frac{1}{8}\sqrt{q}x - \frac{1}{8}pt + \delta) + (\tanh(\frac{1}{8}\sqrt{q}x - \frac{1}{8}pt + \delta))^2)},$$
$$u(x,t) = \frac{2p\tanh(\frac{1}{4}\sqrt{q}x + \frac{1}{4}pt + \delta)}{q(1+\tanh(\frac{1}{4}\sqrt{q}x + \frac{1}{4}pt + \delta))},$$
$$u(x,t) = \frac{4p\tanh(\frac{1}{8}\sqrt{q}x + \frac{1}{8}pt + \delta)}{q(1 + 2\tanh(\frac{1}{8}\sqrt{q}x + \frac{1}{8}pt + \delta) + (\tanh(\frac{1}{8}\sqrt{q}x + \frac{1}{8}pt + \delta))^2)},$$
$$u(x,t) = \frac{2p}{q(1+\tanh(\frac{1}{4}\sqrt{q}x + \frac{1}{4}pt + \delta))},$$
$$u(x,t) = \frac{-2p}{q(-1+\tanh(\frac{1}{4}\sqrt{q}x - \frac{1}{4}pt + \delta))}.$$

Again, the package RAEEM obtains no solutions, while the package TWSolutions returns three nontrivial tanh type solutions.

```
> TWSolutions(pde2,function=[tanh],remove_redundant=true);
```

$$u(x,t) = \frac{p}{q},$$
$$u(x,t) = \frac{-2p}{q(-1+\tanh(-\_C1 + \frac{1}{4}\sqrt{q}x - \frac{1}{4}pt))},$$
$$u(x,t) = \frac{-2p}{q(\tanh(\_C1 + \frac{1}{4}\sqrt{q}x - \frac{1}{4}pt) - 1)},$$
$$u(x,t) = \frac{2p}{q(1+\tanh(-\_C1 + \frac{1}{4}\sqrt{q}x + \frac{1}{4}pt))}.$$

Example 3. (See [6].) A $(2+1)$-dimensional KdV–Burgers equation $(u_t + uu_x + pu_{xxx} - qu_{xx})_x + ru_{yy} = 0$.

For this nonlinear PDE, the new package TWS obtains 17 nontrivial solutions. They are as follows.

- 2 rational type solutions:

$$u(x,y,t) = \frac{a_{-2}}{(\lambda_3 t+\delta)^2} + \frac{a_{-1}}{\lambda_3 t + \delta} + a_0 + a_1(\lambda_3 t+\delta) + a_2(\lambda_3 t+\delta)^2,$$
$$u(x,y,t) = a_0 + a_1(\lambda_2 y + \lambda_3 t + \delta).$$

- 1 exponential type solution:

$$u(x,y,t) = \frac{a_{-2}}{(e^{\lambda_3 t+\delta})^2} + \frac{a_{-1}}{e^{\lambda_3 t+\delta}} + a_0 + a_1 e^{\lambda_3 t+\delta} + a_2(e^{\lambda_3 t+\delta})^2.$$

- 1 tan/cot type solution:

$$u(x,y,t) = \frac{a_{-2}}{(\tan(\lambda_3 t+\delta))^2} + \frac{a_{-1}}{\tan(\lambda_3 t+\delta)} + a_0 + a_1\tan(\lambda_3 t+\delta) + a_2\big(\tan(\lambda_3 t+\delta)\big)^2.$$

- 1 sec/cos type solution:

$$u(x,y,t) = \frac{a_{-2}}{(\sec(\lambda_3 t+\delta))^2} + \frac{a_{-1}}{\sec(\lambda_3 t+\delta)} + a_0 + a_1\sec(\lambda_3 t+\delta) + a_2\big(\sec(\lambda_3 t+\delta)\big)^2.$$

- 1 sech/cosh type solution:

$$u(x, y, t) = \frac{a_{-2}}{(\operatorname{sech}(\lambda_3 t + \delta))^2} + \frac{a_{-1}}{\operatorname{sech}(\lambda_3 t + \delta)} + a_0 + a_1 \operatorname{sech}(\lambda_3 t + \delta) + a_2\big(\operatorname{sech}(\lambda_3 t + \delta)\big)^2.$$

- 1 csc/sin type solution:

$$u(x, y, t) = \frac{a_{-2}}{(\csc(\lambda_3 t + \delta))^2} + \frac{a_{-1}}{\csc(\lambda_3 t + \delta)} + a_0 + a_1 \csc(\lambda_3 t + \delta) + a_2\big(\csc(\lambda_3 t + \delta)\big)^2.$$

- 1 csch/sinh type solutions:

$$u(x, y, t) = \frac{a_{-2}}{(\operatorname{csch}(\lambda_3 t + \delta))^2} + \frac{a_{-1}}{\operatorname{csch}(\lambda_3 t + \delta)} + a_0 + a_1 \operatorname{csch}(\lambda_3 t + \delta) + a_2\big(\operatorname{csch}(\lambda_3 t + \delta)\big)^2.$$

- 7 tanh/coth type solutions:

$$u(x, y, t) = a_0 + \frac{6}{25} q^2 p^{-1} \tanh\eta - \frac{3}{25} q^2 p^{-1} (\tanh\eta)^2, \quad \text{where}$$

$$\eta = -\frac{qx}{10p} + \lambda_2 y + \frac{(-3q^4 + 2500 r \lambda_2^2 p^3 + 25 q^2 a_0 p)t}{250 p^2 q} + \delta.$$

$$u(x, y, t) = -\frac{3}{100} q^2 p^{-1} (\tanh\eta)^{-2} - \frac{3}{25} q^2 p^{-1} (\tanh\eta)^{-1} + a_0 - \frac{3}{25} q^2 p^{-1} \tanh\eta - \frac{3}{100} q^2 p^{-1} (\tanh\eta)^2, \quad \text{where}$$

$$\eta = \frac{qx}{20p} + \lambda_2 y - \frac{(-3q^4 + 20000 r \lambda_2^2 p^3 + 50 q^2 a_0 p)t}{1000 p^2 q} + \delta.$$

$$u(x, y, t) = -\frac{3}{100} q^2 p^{-1} (\tanh\eta)^{-2} + \frac{3}{25} q^2 p^{-1} (\tanh\eta)^{-1} + a_0 + \frac{3}{25} q^2 p^{-1} \tanh\eta - \frac{3}{100} q^2 p^{-1} (\tanh\eta)^2, \quad \text{where}$$

$$\eta = -\frac{qx}{20p} + \lambda_2 y + \frac{(-3q^4 + 20000 r \lambda_2^2 p^3 + 50 q^2 a_0 p)t}{1000 p^2 q} + \delta.$$

$$u(x, y, t) = -\frac{3}{25} q^2 p^{-1} (\tanh\eta)^{-2} - \frac{6}{25} q^2 p^{-1} (\tanh\eta)^{-1} + a_0, \quad \text{where}$$

$$\eta = \frac{qx}{10p} + \lambda_2 y - \frac{(-3q^4 + 2500 r \lambda_2^2 p^3 + 25 q^2 a_0 p)t}{250 p^2 q} + \delta.$$

$$u(x, y, t) = -\frac{3}{25} q^2 p^{-1} (\tanh\eta)^{-2} + \frac{6}{25} q^2 p^{-1} (\tanh\eta)^{-1} + a_0, \quad \text{where}$$

$$\eta = -\frac{qx}{10p} + \lambda_2 y + \frac{(-3q^4 + 2500 r \lambda_2^2 p^3 + 25 q^2 a_0 p)t}{250 p^2 q} + \delta.$$

$$u(x, y, t) = \frac{a_{-2}}{(\tanh\eta)^2} + \frac{a_{-1}}{\tanh\eta} + a_0 + a_1 \tanh\eta + a_2 (\tanh\eta)^2, \quad \text{where}$$

$$\eta = \lambda_3 t + \delta.$$

$$u(x, y, t) = a_0 - \frac{6}{25} q^2 p^{-1} \tanh\eta - \frac{3}{25} q^2 p^{-1} (\tanh\eta)^2, \quad \text{where}$$

$$\eta = \frac{qx}{10p} + \lambda_2 y - \frac{(-3q^4 + 2500 r \lambda_2^2 p^3 + 25 q^2 a_0 p)t}{250 p^2 q} + \delta.$$

- 2 Jacobi elliptic type solutions:

$$u(x, y, t) = \frac{a_{-2}}{(\operatorname{JacobiCN}(\eta, \omega))^2} + \frac{a_{-1}}{\operatorname{JacobiCN}(\eta, \omega)} + a_0 + a_1 \operatorname{JacobiCN}(\eta, \omega) + a_2\big(\operatorname{JacobiCN}(\eta, \omega)\big)^2, \quad \text{where}$$

$$\eta = \lambda_3 t + \delta.$$

$$u(x, y, t) = \frac{a_{-2}}{(\operatorname{JacobiSN}(\eta, \omega))^2} + \frac{a_{-1}}{\operatorname{JacobiSN}(\eta, \omega)} + a_0 + a_1 \operatorname{JacobiSN}(\eta, \omega) + a_2\big(\operatorname{JacobiSN}(\eta, \omega)\big)^2, \quad \text{where}$$

$$\eta = \lambda_3 t + \delta.$$

On the other hand, the package RAEEM returns 3 tanh type nontrivial solutions as follows. Notice that the third solution can be transferred into tanh type.

- $u(\xi) = a_0 + a_1 \tanh(\xi) + a_2(\tanh(\xi))^2, \xi = k(x - c_1 y - tc_2)$, where

$$a_2 = -\frac{3q^2}{25p}, \qquad a_0 = \frac{3q^2 + 25pc_2 - 25pc_1^2 r}{25p}, \qquad a_1 = \frac{6q^2}{25p}, k = -\frac{q}{10p} \quad \text{and}$$

$$a_2 = -\frac{3q^2}{25p}, \qquad a_1 = -\frac{6q^2}{25p}, \qquad k = \frac{q}{10p}, \qquad a_0 = \frac{3q^2 + 25pc_2 - 25pc_1^2 r}{25p}.$$

- $u(\xi) = a_0 \frac{a_1(\mathrm{sech}(\xi))^2}{\tanh(\xi)-1} + \frac{a_2(\mathrm{sech}(\xi))^4}{(\tanh(\xi)-1)^2}, \xi = k(x - yc_1 - tc_2)$, where

$$a_1 = 0, \qquad a_2 = -\frac{3q^2}{25p}, \qquad k = \frac{q}{10p}, \qquad a_0 = \frac{25pc_2 - 25pc_1^2 r + 6q^2}{25p}$$

The package TWSolutions obtains 8 nontrivial solutions of tan type, cot type, tanh type and coth type. Notice that, the tan type and cot type solutions are complex solutions.

- 2 cot type solutions:

$$u(x, y, t) = \frac{2500r\_C3^2 p^3 + 3q^4 + 250i\_C4qp^2}{25q^2 p} - \frac{6}{25} iq^2 (\cot\eta) p^{-1} + \frac{3}{25} q^2 (\cot\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = \_C1 + \frac{1/10iqx}{p} + \_C3y + \_C4t.$$

$$u(x, y, t) = \frac{2500r\_C3^2 p^3 + 3q^4 - 250i\_C4qp^2}{25q^2 p} - \frac{6}{25} iq^2 (\cot\eta) p^{-1} + \frac{3}{25} q^2 (\cot\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = -\_C1 + \frac{1/10iqx}{p} - \_C3y - \_C4t.$$

- 2 coth type solutions:

$$u(x, y, t) = -\frac{-3q^4 - 250_C4qp^2 + 2500r\_C3^2 p^3}{25q^2 p} - \frac{6}{25} q^2 (\coth\eta) p^{-1} - \frac{3}{25} q^2 (\coth\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = -\_C1 + \frac{qx}{10p} - \_C3y - \_C4t.$$

$$u(x, y, t) = -\frac{-3q^4 + 250\_C4qp^2 + 2500r\_C3^2 p^3}{25q^2 p} - \frac{6}{25} q^2 (\coth\eta) p^{-1} - \frac{3}{25} q^2 (\coth\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = \_C1 + \frac{qx}{10p} + \_C3y + \_C4t.$$

- 2 tan type solutions:

$$u(x, y, t) = \frac{2500r\_C3^2 p^3 + 3q^4 + 250i\_C4qp^2}{25q^2 p} + \frac{6}{25} iq^2 (\tan\eta) p^{-1} + \frac{3}{25} q^2 (\tan\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = \_C1 + \frac{1/10iqx}{p} + \_C3y + \_C4t.$$

$$u(x, y, t) = \frac{2500r\_C3^2 p^3 + 3q^4 - 250i\_C4qp^2}{25q^2 p} + \frac{6}{25} iq^2 (\tan\eta) p^{-1} + \frac{3}{25} q^2 (\tan\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = -\_C1 + \frac{1/10iqx}{p} - \_C3y - \_C4t.$$

- 2 tanh type solutions:

$$u(x, y, t) = -\frac{-3q^4 - 250\_C4qp^2 + 2500r\_C3^2 p^3}{25q^2 p} - \frac{6}{25} q^2 (\tanh\eta) p^{-1} - \frac{3}{25} q^2 (\tanh\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = -\_C1 + \frac{qx}{10p} - \_C3y - \_C4t.$$

$$u(x, y, t) = -\frac{-3q^4 + 250\_C4qp^2 + 2500r\_C3^2 p^3}{25q^2 p} - \frac{6}{25} q^2 (\tanh\eta) p^{-1} - \frac{3}{25} q^2 (\tanh\eta)^2 p^{-1}, \quad \text{where}$$

$$\eta = \_C1 + \frac{qx}{10p} + \_C3y + \_C4t.$$

Example 4. (See [1].) The Zakharov–Kuznetsov KdV-type equation $u_t + \alpha u u_x + u_{xxx} + u_{xyy} + u_{xzz} = 0$.

For this $(3+1)$-dimensional PDE, the new package TWS obtains 27 nontrivial solutions. They are as follows.

- 2 rational type solutions:

$$u(x,y,z,t) = -12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\lambda_1 x+\lambda_2 y+\lambda_3 z-\alpha\lambda_1 a_0 t+\delta)^2}+a_0.$$

$$u(x,y,z,t) = \frac{a_{-2}}{\eta^2}+\frac{a_{-1}}{\eta}+a_0+a_1\eta+a_2(\eta)^2, \quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

- 1 exponential type solution:

$$u(x,y,z,t) = \frac{a_{-2}}{(e^{\lambda_2 y+\lambda_3 z+\delta})^2}+\frac{a_{-1}}{e^{\lambda_2 y+\lambda_3 z+\delta}}+a_0+a_1 e^{\lambda_2 y+\lambda_3 z+\delta}+a_2(e^{\lambda_2 y+\lambda_3 z+\delta})^2.$$

- 4 tan type solutions:

$$u(x,y,z,t) = -12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\tan(\lambda_1 x+\lambda_2 y+\lambda_3 z+(-8\lambda_3^2\lambda_1-8\lambda_2^2\lambda_1-8\lambda_1^3-\alpha\lambda_1 a_0)t+\delta))^2}+a_0.$$

$$u(x,y,z,t) = \frac{a_{-2}}{(\tan\eta)^2}+\frac{a_{-1}}{\tan\eta}+a_0+a_1\tan\eta+a_2(\tan\eta)^2, \quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

$$u(x,y,z,t) = a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\tan\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(-8\lambda_3^2\lambda_1-8\lambda_2^2\lambda_1-8\lambda_1^3-\alpha\lambda_1 a_0\right)t+\delta.$$

$$u(x,y,z,t) = -12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\tan\eta)^2}+a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\tan\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(-8\lambda_3^2\lambda_1-8\lambda_2^2\lambda_1-8\lambda_1^3-\alpha\lambda_1 a_0\right)t+\delta.$$

- 2 sec type solutions:

$$u(x,y,z,t) = \frac{a_{-2}}{(\sec\eta)^2}+\frac{a_{-1}}{\sec\eta}+a_0+a_1\sec\eta+a_2(\sec\eta)^2, \text{ where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

$$u(x,y,z,t) = a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\sec\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(-\alpha\lambda_1 a_0+4\lambda_1^3+4\lambda_3^2\lambda_1+4\lambda_2^2\lambda_1\right)t+\delta.$$

- 2 sech type solutions:

$$u(x,y,z,t) = \frac{a_{-2}}{(\operatorname{sech}\eta)^2}+\frac{a_{-1}}{\operatorname{sech}\eta}+a_0+a_1\operatorname{sech}\eta+a_2(\operatorname{sech}\eta)^2, \quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

$$u(x,y,z,t) = a_0+12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\operatorname{sech}\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(-\alpha\lambda_1 a_0-4\lambda_1^3-4\lambda_3^2\lambda_1-4\lambda_2^2\lambda_1\right)t+\delta.$$

- 4 tanh type solutions:

$$u(x,y,z,t) = -12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\tanh(\lambda_1 x+\lambda_2 y+\lambda_3 z+(8\lambda_1^3+8\lambda_3^2\lambda_1+8\lambda_2^2\lambda_1-\alpha\lambda_1 a_0)t+\delta))^2}+a_0.$$

$$u(x,y,z,t) = \frac{a_{-2}}{(\tanh\eta)^2}+\frac{a_{-1}}{\tanh\eta}+a_0+a_1\tanh\eta+a_2(\tanh\eta)^2, \quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

$$u(x,y,z,t) = a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\tanh\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(8\lambda_1^3+8\lambda_3^2\lambda_1+8\lambda_2^2\lambda_1-\alpha\lambda_1 a_0\right)t+\delta.$$

$$u(x,y,z,t) = -12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\tanh\eta)^2}+a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\tanh\eta)^2}{\alpha}, \quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\left(8\lambda_1^3+8\lambda_3^2\lambda_1+8\lambda_2^2\lambda_1-\alpha\lambda_1 a_0\right)t+\delta.$$

- 4 JacobiCN type solutions:

$$u(x,y,z,t)=\frac{a_{-2}}{(\mathrm{JacobiCN}(\eta,\omega))^2}+\frac{a_{-1}}{\mathrm{JacobiCN}(\eta,\omega)}+a_0+a_1\,\mathrm{JacobiCN}(\eta,\omega)+a_2\big(\mathrm{JacobiCN}(\eta,\omega)\big)^2,\quad \text{where}$$
$$\eta=\lambda_2 y+\lambda_3 z+\delta.$$
$$u(x,y,z,t)=a_0+12\frac{\omega^2(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\mathrm{JacobiCN}(\eta,\omega))^2}{\alpha},$$
$$u(x,y,z,t)=12\frac{-\lambda_1^2-\lambda_2^2-\lambda_3^2+\lambda_3^2\omega^2+\lambda_2^2\omega^2+\lambda_1^2\omega^2}{\alpha(\mathrm{JacobiCN}(\eta,\omega))^2}+a_0,\quad \text{and}$$
$$u(x,y,z,t)=12\frac{-\lambda_1^2-\lambda_2^2-\lambda_3^2+\lambda_3^2\omega^2+\lambda_2^2\omega^2+\lambda_1^2\omega^2}{\alpha(\mathrm{JacobiCN}(\eta,\omega))^2}+a_0+12\frac{\omega^2(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\mathrm{JacobiCN}(\eta,\omega))^2}{\alpha},\quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\delta+\big(-\alpha\lambda_1 a_0+4\lambda_3^2\lambda_1+4\lambda_1^3-8\lambda_2^2\lambda_1\omega^2-8\lambda_3^2\lambda_1\omega^2-8\lambda_1^3\omega^2+4\lambda_2^2\lambda_1\big)t.$$

- 4 JacobiSN type solutions:

$$u(x,y,z,t)=\frac{a_{-2}}{(\mathrm{JacobiSN}(\eta,\omega))^2}+\frac{a_{-1}}{\mathrm{JacobiSN}(\eta,\omega)}+a_0+a_1\,\mathrm{JacobiSN}(\eta,\omega)+a_2\big(\mathrm{JacobiSN}(\eta,\omega)\big)^2,\quad \text{where}$$
$$\eta=\lambda_2 y+\lambda_3 z+\delta.$$
$$u(x,y,z,t)=a_0-12\frac{\omega^2(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\mathrm{JacobiSN}(\eta,\omega))^2}{\alpha},$$
$$u(x,y,z,t)=-12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\mathrm{JacobiSN}(\eta,\omega))^2}+a_0-12\frac{\omega^2(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\mathrm{JacobiSN}(\eta,\omega))^2}{\alpha},\quad \text{and}$$
$$u(x,y,z,t)=-12\frac{\lambda_2^2+\lambda_1^2+\lambda_3^2}{\alpha(\mathrm{JacobiSN}(\eta,\omega))^2}+a_0,\quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\big(4\lambda_2^2\lambda_1+4\lambda_3^2\lambda_1\omega^2-\alpha\lambda_1 a_0+4\lambda_2^2\lambda_1\omega^2+4\lambda_3^2\lambda_1+4\lambda_1^3\omega^2+4\lambda_1^3\big)t+\delta.$$

- 2 csc type solutions:

$$u(x,y,z,t)=a_0-12\frac{(\lambda_2^2+\lambda_1^2+{\lambda_3}^2)(\csc\eta)^2}{\alpha},\quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\big(-\alpha\lambda_1 a_0+4\lambda_1^3+4\lambda_3^2\lambda_1+4\lambda_2^2\lambda_1\big)t+\delta.$$
$$u(x,y,z,t)=\frac{a_{-2}}{(\csc\eta)^2}+\frac{a_{-1}}{\csc\eta}+a_0+a_1\csc\eta+a_2(\csc\eta)^2,\quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$

- 2 csch type solutions:

$$u(x,y,z,t)=\frac{a_{-2}}{(\operatorname{csch}\eta)^2}+\frac{a_{-1}}{\operatorname{csch}\eta}+a_0+a_1\operatorname{csch}\eta+a_2(\operatorname{csch}\eta)^2,\quad \text{where } \eta=\lambda_2 y+\lambda_3 z+\delta.$$
$$u(x,y,z,t)=a_0-12\frac{(\lambda_2^2+\lambda_1^2+\lambda_3^2)(\operatorname{csch}\eta)^2}{\alpha},\quad \text{where}$$
$$\eta=\lambda_1 x+\lambda_2 y+\lambda_3 z+\big(-\alpha\lambda_1 a_0-4\lambda_1^3-4\lambda_3^2\lambda_1-4\lambda_2^2\lambda_1\big)t+\delta.$$

On the other hand, the package TWSolutions returns an error message if the argument function = *functionlist* (see Example 1) is specified, while the package RAEEM obtains 7 nontrivial solutions as follows.

- 1 sec type solution:

$$u(\xi)=a_0+a_1\sec(\xi)+a_2\big(\sec(\xi)\big)^2,\qquad \xi=-k(-x+yc_1+c_2z+tc_3),\quad \text{where}$$
$$a_0=\frac{4k^2c_2^2+4k^2c_1^2+4k^2+c_3}{\alpha},\qquad a_1=0,\qquad a_2=-12\frac{k^2(1+c_1^2+c_2^2)}{\alpha}.$$

- 1 sech type solution:

$$u(\xi)=a_0+a_1\operatorname{sech}(\xi)+a_2\big(\operatorname{sech}(\xi)\big)^2,\qquad \xi=-k(-x+yc_1+c_2z+tc_3),\quad \text{where}$$
$$a_0=\frac{-4k^2c_1^2-4k^2-4k^2c_2^2+c_3}{\alpha},\qquad a_2=12\frac{k^2(1+c_1^2+c_2^2)}{\alpha},\qquad a_1=0.$$

- 1 tan type solution:

$$u(\xi) = a_0 + a_1 \tan(\xi) + a_2\big(\tan(\xi)\big)^2, \qquad \xi = -k(-x + yc_1 + c_2 z + tc_3), \quad \text{where}$$
$$a_0 = \frac{-8k^2c_1^2 - 8k^2 - 8k^2c_2^2 + c_3}{\alpha}, \qquad a_1 = 0, \qquad a_2 = -12\frac{k^2(1 + c_1^2 + c_2^2)}{\alpha}.$$

- 2 tanh type solution:

$$u(\xi) = a_0 + a_1 \tanh(\xi) + a_2\big(\tanh(\xi)\big)^2, \qquad \xi = -k(-x + yc_1 + c_2 z + tc_3), \quad \text{where}$$
$$a_1 = 0, \qquad a_2 = -12\frac{k^2(1 + c_1^2 + c_2^2)}{\alpha}, \qquad a_0 = \frac{8k^2 + 8k^2c_2^2 + c_3 + 8k^2c_1^2}{\alpha}.$$
$$u(\xi) = a_0 + \frac{a_1(\operatorname{sech}(\xi))^2}{\tanh(\xi) - 1} + \frac{a_2(\operatorname{sech}(\xi))^4}{(\tanh(\xi) - 1)^2}, \qquad \xi = -k(-x + yc_1 + c_2 z + tc_3), \quad \text{where}$$
$$a_2 = -12\frac{k^2(1 + c_1^2 + c_2^2)}{\alpha}, \quad a_1 = -24\frac{k^2(1 + c_1^2 + c_2^2)}{\alpha}, \qquad a_0 = \frac{-4k^2c_2^2 - 4k^2 - 4k^2c_1^2 + c_3}{\alpha}.$$

- 1 JacobiCN type solution:

$$u(\xi) = a_0 + a_1 \operatorname{cn}(\xi) + a_2\big(\operatorname{cn}(\xi)\big)^2, \qquad \xi = -k(-x + yc_1 + c_2 z + tc_3), \quad \text{where}$$
$$a_2 = 12\frac{k^2R^2(1 + c_1^2 + c_2^2)}{\alpha}, \qquad a_0 = \frac{4k^2c_2^2 + c_3 - 8k^2c_2^2R^2 - 8k^2c_1^2R^2 - 8k^2R^2 + 4k^2c_1^2 + 4k^2}{\alpha}, \qquad a_1 = 0.$$

- 1 JacobiSN type solution:

$$u(\xi) = a_0 + a_1 \operatorname{sn}(\xi) + a_2\big(\operatorname{sn}(\xi)\big)^2, \qquad \xi = -k(-x + yc_1 + c_2 z + tc_3), \quad \text{where}$$
$$a_1 = 0, \qquad a_2 = -12\frac{k^2R^2(1 + c_1^2 + c_2^2)}{\alpha}, \qquad a_0 = \frac{4k^2 + 4k^2c_1^2R^2 + 4k^2c_2^2R^2 + 4k^2c_2^2 + 4k^2c_1^2 + 4k^2R^2 + c_3}{\alpha}.$$

4. Conclusions

We have described a new algorithm and its implementation in MAPLE 10 for automatically computing the travelling wave solutions to nonlinear PDEs. For single PDEs, this new package has two main advantages: first, for PDEs whose balancing numbers are not positive integers, TWS works much better than existing packages; second, for most cases, TWS obtains more solutions than existing packages. However, nothing is perfect. The main disadvantage of this package is that the current version cannot handle systems of PDEs. Theoretically, extending it for handling systems of PDEs is not a difficult problem, but substantial work has to be done before we can do so. Therefore, we leave it to the next version.

Acknowledgements

We are indebted to Professor Gregory J. Reid for his help at the first stage of this project. We would like to thank the anonymous referees for valuable comments which help us greatly improve the first version of the paper.

References

[1] D. Baldwin, Ü. Göktas, W. Hereman, L. Hong, R.S. Martino, J.C. Miller, J. Symb. Comp. 37 (2004) 669.
[2] F. Cariello, M. Tabor, Physica D 39 (1989) 77.
[3] E.S. Cheb-Terrab, K. von Bulow, Comput. Phys. Comm. 90 (1995) 102.
[4] P.G. Drazin, R.S. Johnson, Solitons: An Introduction, Cambridge University Press, 1989.
[5] B.R. Duffy, E.J. Parkes, Phys. Lett. A 214 (1996) 271.
[6] S.A. Elwakil, S.K. El-Labany, M.A. Zahran, R. Sabry, Phys. Lett. A 299 (2002) 179.
[7] S.A. Elwakil, S.K. El-Labany, M.A. Zahran, R. Sabry, Appl. Math. Comput. 161 (2005) 403.
[8] E. Fan, Phys. Lett. 277 (2000) 212.
[9] E. Fan, Z. Naturforsch. 56A (2001) 312.
[10] Y.T. Gao, B. Tian, Comput. Math. Appl. 33 (1997) 115.
[11] V.V. Gudkov, J. Math. Phys. 38 (1997) 4794.
[12] W. Hereman, A. Korpel, P.P. Banerjee, Wave Motion 7 (1985) 283.
[13] W. Hereman, M. Takaoka, J. Phys. A: Math. Gen. 23 (1990) 4805.
[14] Z.B. Li, Y.P. Liu, Comput. Phys. Comm. 148 (2002) 256.
[15] Z.B. Li, Y.P. Liu, Comput. Phys. Comm. 163 (2004) 191.

[16] Y.P. Liu, Z.B. Li, Chin. Phys. Lett. 19 (2002) 1228.
[17] W. Malfliet, Amer. J. Phys. 60 (1992) 650.
[18] W. Malfliet, W. Hereman, Phys. Scripta 54 (1996) 563.
[19] E.J. Parkes, B.R. Duffy, Comput. Phys. Comm. 98 (1996) 288.
[20] E.J. Parkes, B.R. Duffy, Phys. Lett. A 229 (1997) 217.