SFOPDES: A Stepwise First Order Partial Differential Equations Solver with a Computer Algebra System

José Luis Galán-García*, Gabriel Aguilera-Venegas, Pedro Rodríguez-Cielos,
Yolanda Padilla-Domínguez, María Ángeles Galán-García

Escuela de Ingenierías Industriales
University of Málaga, Málaga, Spain

Abstract

Partial Differential Equations (PDE) appear in multiple Physic and Engineering applications. Normally, when modeling an application, the use of well-known and already solved PDE are considered. But what happens if a new PDE is used? Solving a new PDE is not an easy task. In this paper, we use a Computer Algebra System (Cas) in order to find the solution of PDE of first order.

Specifically, we deal with *Pfaff Equations*, *Quasilinear PDE* and *general first order PDE* (using *Lagrange-Charpit Method*).

To solve these PDE, we combine the power of a Cas with the flexibility of programming with it. Furthermore, the developed programs do not only provide the final result but also display all the intermediate steps which lead to find the solution of the PDE. This way, we introduce SFOPDES, a new Stepwise First Order PDE Solver which serves as a tutorial showing, step by step, the way to deal with PDE.

Keywords: Partial Differential Equations (PDE), PDE Solver, Pfaff Equations, Quasilinear PDE, First order PDE, Computer Algebra Systems

---

*Corresponding author

Email addresses: `jlgalan@uma.es` (José Luis Galán-García), `gabri@citma.uma.es` (Gabriel Aguilera-Venegas), `prodriguez@uma.es` (Pedro Rodríguez-Cielos), `ypadilla@ctima.uma.es` (Yolanda Padilla-Domínguez), `magalan@ctima.uma.es` (María Ángeles Galán-García)

Preprint submitted to Computers & Mathematics with Applications *April 22, 2019*

1. Introduction

Working with Partial Differential Equations (PDE) is a common task in Engineering and Sciences applications. Many phenomena can be modeled with PDE. Solving PDE normally requires difficult computation and in many situations, the way to find the solution is by means of an approximation algorithm from Numerical Analysis theory. But what happens if the PDE to solve depends on one or more general parameters? Applying numerical methods normally requires giving specific values for these parameters. Finding a solution of a PDE with parameters is therefore a difficult task that normally needs to be solved analytically. To achieve this goal, we will use a Computer Algebra System (Cas).

Cas are computer software that can deal with exact computations and manipulate algebraical expressions. For example, a Cas can expand $(x+y+z)^3$ and return the result $x^3+3x^2y+3x^2z+3xy^2+6xyz+3xz^2+y^3+3y^2z+3yz^2+z^3$ while a numerical software needs the values of x, y and z to obtain the result. In addition, $\displaystyle\int_0^1 \sqrt{1-x^2}\,\mathrm{d}x = \frac{\pi}{4}$ in a Cas while a numerical software will return an approximation of the value with some significative digits.

There are a lot of Cas available in the market (see [1] for a complete list and description of Cas from their origins) but we are interested in those ones which allow the use of programming and with a large amount of built-in functions. Among the Cas satisfying these characteristic, there are both, proprietary and free software. For example, some proprietary ones that can be considered are Derive [2], Mathematica [3] or Maple [4] while Maxima [5] or Sage [6, 7] are free software that could also be used. Due to the expertise of the authors, the chosen one is Derive since we have a long experience in using this software both in Engineering education [8, 9] and research [10, 11].

The existing Cas PDE solvers use the Cas as a black box [12, 13, 14, 15, 16]. That is, the Cas only provide the solution of the PDE without any further information. We are interested in this work in the use of a Cas for solving PDE as a white box, providing not only the solution but also all the procedure of solving the PDE.

Therefore, the main goal of this paper is the development of `SFOPDES`, a new stepwise solver using the software DERIVE to solve first order PDE step by step. We will describe this new solver and will provide the code freely so that anyone can use and modify it attending to the specific necessities. SFOPDES is a novelty since we do not know of any other free PDE solver which provides so many detailed steps of the solving procedure. This fact makes SFOPDES to be very useful also in the teaching and learning process in a PDE course for Engineering or Mathematics students.

In the process of solving PDE, first order Ordinary Differential Equations (ODE) are needed. Therefore, SFOPDES does not only solve PDE but ODE also. The programs in SFOPDES can be grouped within the following blocks:

- **First-order ODE**: separable equations and equations reducible to them, homogeneous equations and equations reducible to them, exact differential equations and equations reducible to them (integrating factor technique), linear equations, the Bernoulli equation, the Riccati equation, first-order differential equations and nth degree in y', and generic programs to solve first order ordinary differential equations.
- **First-order PDE**: Pfaff Differential Equations, Quasilinear PDE and Lagrange-Charpit Method for First-order PDE.

In order to make the paper self-contained, section 2 is devoted to explain the definitions and theory needed to solve the PDE considered. In section 3 the syntax of the programs developed in the library of functions of SFOPDES is described. Some examples of use are shown in section 4. Finally, section 5 is devoted to the conclusions and the enumeration of some ideas for future related work.

2. Theoretical framework

This section is included in order the article to be self-contained and can be skipped by a reader familiar with the subject. In this section, the main ideas for solving the considered PDE are stated. For a deeper study, there are many text books which deal with PDE such as [17, 18].

2.1. Pfaff equations

Let P, Q and R be scalar fields: $P, Q, R : \mathbb{R}^3 \longrightarrow \mathbb{R}$. The equation

$$P(x, y, z)\,\mathrm{d}x + Q(x, y, z)\,\mathrm{d}y + R(x, y, z)\,\mathrm{d}z = 0 \tag{1}$$

is called a total differential equation, also known as *Pfaff equation.* The first term (in short, $P\,\mathrm{d}x + Q\,\mathrm{d}y + R\,\mathrm{d}z$) is called a *Pfaff differential form.*

If a Pfaff differential form is an exact differential form, then:

$$\frac{\partial P}{\partial y} = \frac{\partial Q}{\partial x} \quad ; \quad \frac{\partial P}{\partial z} = \frac{\partial R}{\partial x} \quad ; \quad \frac{\partial Q}{\partial z} = \frac{\partial R}{\partial y}$$

If the differential form is not an exact one, it is sometimes possible to find an integrating factor $\mu(x, y, z)$ which makes the form $\mu(x, y, z)\Big(P(x, y, z)\,\mathrm{d}x + Q(x, y, z)\,\mathrm{d}y + R(x, y, z)\,\mathrm{d}z\Big)$ to be an exact one. In this case, there exist a so called *potential function* $U(x, y, z)$ such that $\mathrm{d}U(x, y, z) = \mu(x, y, z)\Big(P(x, y, z)\mathrm{d}x + Q(x, y, z)\,\mathrm{d}y + R(x, y, z)\,\mathrm{d}z\Big)$.

A Pfaff equation (1) is integrable if and only if an integrating factor exists. In this case, the general solution of (1) is given by $U(x, y, z) = C$ where U is a potential function and $C \in \mathbb{R}$ is an arbitrary constant. Therefore, in order to solve the general solution of (1), a potential function U must be found (if it exists).

Clairaut also established that a Pfaff equation (1) is integrable if and only if $\vec{F} \cdot \left(\vec{\nabla} \times \vec{F}\right) = 0$, where $\vec{F} = (P, Q, R)$, $\vec{\nabla} = \left(\frac{\partial}{\partial\, x}, \frac{\partial}{\partial\, y}, \frac{\partial}{\partial\, z}\right)$ and $\cdot$ and $\times$ are the dot and cross products respectively.

2.1.1. General method

Once checked that a Pfaff equation is integrable, the Clairaut's method for finding its general solution consists of the following procedure:

1. Consider one of the three variables x, y, z as a constant parameter. For example, let consider z as a constant parameter[1]. In this case, $\mathrm{d}z = 0$ and (1) simplifies to $P(x, y, z)\mathrm{d}x + Q(x, y, z)\mathrm{d}y = 0$ which is an ordinary differential equation (ODE) in $\mathbb{R}^2$ for which z is a constant.

   This first step in the general method consists of finding the solution of the previous ODE, obtaining an expression $G(x, y, z) = C$ as the solution of such ODE with C an arbitrary real constant.
2. This second steps consists of finding an integrating factor $\mu(x, y, z)$ from any of the following two equations:

   $$\frac{\partial\, G(x,y,z)}{\partial\, x} = \mu(x,y,z)\, P(x,y,z) \quad ; \quad \frac{\partial\, G(x,y,z)}{\partial\, y} = \mu(x,y,z)\, Q(x,y,z)$$
3. Solve the following ODE:

   $$\mathrm{d}G + K(G, z)\, \mathrm{d}z = 0$$

   where $K(G, z) = \mu(x, y, z)\, R(x, y, z) - \dfrac{\partial\, G(x,y,z)}{\partial\, z}$

   Once the solution of the previous ODE is solved, the value of G is replaced with the one computed in the first step obtaining, in this way, the general solution of the Pfaff equation (1).

2.1.2. Particular cases

The followings are some particular cases of Pfaff equations for which there exist specific and easier procedures for obtaining their general solution than the general method.

1. **Exact differential equation**

   If the Pfaff differential form is an exact one (and this happens if and only if $\vec{\nabla} \times \vec{F} = (0, 0, 0)$), then the corresponding Pfaff equation (1) is

---

[1] In case of considering x or y as the constant parameter, the procedure would be similar.

integrable and its general solution is $U(x,y,z)=C$ where $U(x,y,z)$ is the potential function.

2. **Separable equation**

   If the Pfaff equation can be transformed and expressed in the form $P(x)\mathrm{d}x+Q(y)\,\mathrm{d}y+R(z)\,\mathrm{d}z=0$, then it is integrable and its general solution is given by $\displaystyle\int P(x)\,\mathrm{d}x+\int Q(y)\,\mathrm{d}y+\int R(z)\,\mathrm{d}z=C$.

3. **One-separated variable equation**

   A Pfaff equation is said to have the variable z separated[2] if it can be transformed to $P(x,y)\mathrm{d}x+Q(x,y)\mathrm{d}y+R(z)\mathrm{d}z=0$. In this case, the Pfaff equation is integrable if and only if the differential form in $\mathbb{R}^2$ $P(x,y)\,\mathrm{d}x+Q(x,y)\,\mathrm{d}y$ is an exact one and its general solution is given by $U(x,y)+\displaystyle\int R(z)\,\mathrm{d}z=C$ where $U(x,y)$ is a potential function of $P(x,y)\,\mathrm{d}x+Q(x,y)\,\mathrm{d}y$.

2.2. Quasilinear PDE

Let P,Q and R be scalar fields: $P,Q,R:\mathbb{R}^3\longrightarrow\mathbb{R}$. The equation

$$P(x,y,z)\,\frac{\partial z}{\partial x}+Q(x,y,z)\,\frac{\partial z}{\partial y}=R(x,y,z) \tag{2}$$

is called a *quasilinear partial differential equation* of first order. We will denote the above equation as $P\,p+Q\,q=R$, with $p=\dfrac{\partial z}{\partial x}$ and $q=\dfrac{\partial z}{\partial y}$.

2.2.1. General solution

The general solution of (2) is $\varphi(f,g)=0$ with φ and arbitrary function and $f(x,y,z)=C_1$; $g(x,y,z)=C_2$ are two linearly independent first integrals of:

$$\frac{\mathrm{d}x}{P(x,y,z)}=\frac{\mathrm{d}y}{Q(x,y,z)}=\frac{\mathrm{d}z}{R(x,y,z)} \tag{3}$$

In many occasions, in order to find f or g, we have to consider the following property, looking for integral combinations using the fact that:

$$(3)=\frac{\alpha(x,y,z)\,\mathrm{d}x+\beta(x,y,z)\,\mathrm{d}y+\gamma(x,y,z)\,\mathrm{d}z}{\alpha(x,y,z)\,P(x,y,z)+\beta(x,y,z)\,Q(x,y,z)+\gamma(x,y,z)\,R(x,y,z)} \tag{4}$$

---

[2] Similarly, a Pfaff equation can be said to be with x or y separated and the general solution is obtained in a similar way.

for any scalar fields $\alpha(x,y,z), \beta(x,y,z)$ and $\gamma(x,y,z)$.

2.2.2. Particular solutions

Let $\varphi(f,g)=0$ with φ an arbitrary function be the general solution of the quasilinear equation $P\,p+Q\,q=R$. In order to find a particular solution of the quasilinear equation which contents the curve Γ given by the intersection of surfaces $\mathcal{S}_1 \equiv h(x,y,z)=0$ and $\mathcal{S}_2 \equiv k(x,y,z)=0$, the following nonlinear system have to be solved:

$$\left\{\begin{array}{lcl} f(x,y,z) & = & C_1 \\ g(x,y,z) & = & C_2 \\ h(x,y,z) & = & 0 \\ k(x,y,z) & = & 0 \end{array}\right. \tag{5}$$

finding a relation between C_1 and C_2 free of variables x,y,z. That is, from the previous system, we obtain a solution of the form $\phi(C_1,C_2)=0$. Once this is obtained, a particular solution is given by $\phi\Big(f(x,y,z), g(x,y,z)\Big)=0$.

2.3. General first order PDE

The most important results on a first order PDE $F(x,y,z,p,q)=0$ with $p=\dfrac{\partial z}{\partial x}$; $q=\dfrac{\partial z}{\partial y}$ were provided by Lagrange [19] in 1772 when he developed the general theory on non-linear first order PDE. In 1774, Lagrange established the relations between singular, general and complete solutions.

Later, the resolution of a first order PDE was reduced to solve a system of ODE using the Lagrange-Charpit's method [20] merging the ideas from Lagrange's works in 1792 and 1799.

2.3.1. Lagrange-Charpit's general method

Given the first order PDE $F(x,y,z,p,q)=0$, in order to find a complete integral, that is, a biparametric surfaces family $f(x,y,z,C_1,C_2)=0$ satisfying the PDE, the Lagrange-Charpit's method [20] can be used. It consists of the following stepts:

1. Find a first integral $\phi(x,y,z,p,q)=C_1$ from the ODE system:
$$\frac{\mathrm{d}x}{F'_p}=\frac{\mathrm{d}y}{F'_q}=\frac{\mathrm{d}z}{pF'_p+qF'_q}=\frac{-\mathrm{d}p}{F'_x+pF'_z}=\frac{-\mathrm{d}q}{F'_y+qF'_z} \tag{6}$$
2. Solve $p=p(x,y,z,C_1)$ and $q=q(x,y,z,C_1)$ from the algebraic system
$$\begin{cases} \phi(x,y,z,p,q) &= C_1 \\ F(x,y,z,p,q) &= 0 \end{cases} \tag{7}$$
3. Solving the general solution of the Pfaff equation
$$\mathrm{d}z \;=\; p(x,y,z,C_1)\,\mathrm{d}x \;+\; q(x,y,z,C_1)\,\mathrm{d}y \tag{8}$$
   will provide the complete integral $f(x,y,z,C_1,C_2)=0$.

2.3.2. Particular cases

Some of the previous steps in the general method can be avoided or simplified in the followings particular cases:

1. If the PDE can be expressed as $F(p,q)=0$, then $q=C_1$ (or $p=C_1$) is a first integral of (6). Now, the system (7) is reduced to obtain $p=\psi(C_1)$ (or $q=\psi(C_1)$) from $F(p,q)=0$. Substituting p and q in (8) the complete integral $z=\psi(C_1)x+C_1y+C_2$ (or $z=C_1x+\psi(C_1)y+C_2$) is obtained.
2. If $F(x,y,z,p,q)=0$, can be expressed as $g_1(x,p)=g_2(y,q)$, the first and second steps of the general method is reduced to obtain p and q from $g_1(x,p)=g_2(y,q)=C_1$. This way, $p=\varphi_1(x,C_1)$ and $q=\varphi_2(y,C_1)$. Finally, solving $\mathrm{d}z=\varphi_1(x,C_1)\,\mathrm{d}x+\varphi_2(y,C_1)\,\mathrm{d}y$, the complete integral is $z=\displaystyle\int\varphi_1(x,C_1)\,\mathrm{d}x+\int\varphi_2(y,C_1)\,\mathrm{d}y+C_2$.
3. If $F(x,y,z,p,q)=0$, can be expressed as $z=px+qy+g(p,q)$, directly, the compete integral is $z=C_1x+C_2y+g(C_1,C_2)$.

3. Description of SFOPDES

SFOPDES is a new stepwise first order partial differential equation solver. The first versions (in Spanish) were aimed to be used as a tutorial for teaching a partial differential equation courses for Engineering students [21, 22].

SFOPDES has been developed using the CAS DERIVE and its code can be freely downloaded at `http://www.matap.uma.es/jlgalan/SFOPDES/` or by request to the corresponding author at `jlgalan@uma.es`.

The main characteristic of SFOPDES is the possibility of solving first order PDE with optional explanation of all the steps needed to find the solution.

In this section, we describe the syntax and usage of both, the main functions in SFOPDES and some of its auxiliar functions which can be used also to solve not only PDE but first order ODE.

3.1. Syntax of main functions for solving PDE

SFOPDES can solve the following first order PDE: General solution of a Pfaff Differential Equations, general and particular solutions for Quasilinear PDE and, complete integrals for PDE using the Lagrange-Charpit Method.

Optionally, together with the result, explained steps to get the solution of the PDE are provided when the boolean variable `Stepwise` is set to `True` (this is the default option).

If the user wants to get just only the final solution without any intermediate steps, the boolean variable `Stepwise` must be set to `False` using the following instruction in DERIVE: `Stepwise:=False`. At any time, the stepwise functionality can be set on once again with `Stepwise:=True`.

All main programs in SFOPDES have different mandatory arguments and may have also some optional arguments. The first optional one will be always the variable `comments_` (set by default to `Stepwise`) which will allow the user to get the answer to a program with or without the explanation of all steps or just the final solution.

Another important global variable is `Theory`. Setting the variable `Theory` to true or false allows the user to display or not the theory of the PDE being solved.

Therefore, the combination of variables `Theory` and `Stepwise`, allows the user to get the following four possibilities:

1. `Theory:=true` and `Stepwise:=true`: the theory and steps are displayed.

2. `Theory:=true` and `Stepwise:=false`: the theory is displayed but not the steps.
3. `Theory:=false` and `Stepwise:=true`: the steps are displayed but not the theory.
4. `Theory:=false` and `Stepwise:=false`: only the final solution is displayed.

The syntax of the main functions in SFOPDES are:

3.1.1. Pfaff equations

- `Pfaff(P, Q, R, comments_ := Stepwise, var_ := y)`

  to find the general solution of the Pfaff equation (1):

  $$P(x, y, z)\,\mathrm{d}x + Q(x, y, z)\,\mathrm{d}y + R(x, y, z)\,\mathrm{d}z = 0.$$

  The optional fifth argument (`var_`, by default equals to `y`) set the variable to be used as a constant parameter in case that the general method for solving the Pfaff equation needs to be used.

  This program first checks whether the Pfaff equation is an integrable one or not. If it is not integrable, the program ends returning this result. If it is integrable, the program checks if the Pfaff equation presents a particular case. Even more, if not, it tries to manipulate the Pfaff equation in order to transform it in an equivalent one that could present a particular case. If this is the case, the program solves the Pfaff equation attending to the particular case procedure. If not, the program will use the general method described in (2.1.1) to find out the solution.

Examples:

- `Pfaff(3xz+2y, x, x ^ 2, true, x)`

  to solve, step by step, the equation: $(3xz + 2y)\,\mathrm{d}x + x\,\mathrm{d}y + x^2\,\mathrm{d}z = 0$ and considering the variable x as a constant parameter when using the general method to solve the equation.

- `Pfaff(x, y, xyz)`

  to check that the equation: $x\,\mathrm{d}x + y\,\mathrm{d}y + xyz\,\mathrm{d}z = 0$ is not integrable.

- `Pfaff(yz, xz, xy, true)`

  to solve, step by step, the equation: $yz\ \mathrm{d}x + xz\ \mathrm{d}y + xy\ \mathrm{d}z = 0$ converting it in a separable one.

- `Pfaff(y ^ 2, -z, y, true)`

  to solve, step by step, the equation: $y^2\,\mathrm{d}x - z\,\mathrm{d}y + y\,\mathrm{d}z = 0$ converting it in a one-separated variable equation.

See section 4.1 for a detailed executions of these examples.

3.1.2. Quasilinear equations

- `Quasilinear(P, Q, R, comments_ := Stepwise)`

  to find the general solution of the quasilinear equation (2):

$$P(x, y, z)\,\frac{\partial z}{\partial x} + Q(x, y, z)\,\frac{\partial z}{\partial y} = R(x, y, z).$$

- `QuasilinearParticular(P, Q, R, S1, S2, comments_ := Stepwise)`

  to find a particular solution of the quasilinear equation (2)

$$P(x, y, z)\,\frac{\partial z}{\partial x} + Q(x, y, z)\,\frac{\partial z}{\partial y} = R(x, y, z),$$

  which contains the curve Γ given by the intersection of surfaces of equations S1 and S2.

  Both programs have to find two first integrals from the system (3):

$$\frac{\mathrm{d}x}{P(x, y, z)} = \frac{\mathrm{d}y}{Q(x, y, z)} = \frac{\mathrm{d}z}{R(x, y, z)}$$

  and sometimes, it is needed to use the following property (4):

$$(3) = \frac{\alpha(x, y, z)\,\mathrm{d}x + \beta(x, y, z)\,\mathrm{d}y + \gamma(x, y, z)\,\mathrm{d}z}{\alpha(x, y, z)\,P(x, y, z) + \beta(x, y, z)\,Q(x, y, z) + \gamma(x, y, z)\,R(x, y, z)}$$

  for any scalar fields $\alpha(x, y, z), \beta(x, y, z)$ and $\gamma(x, y, z)$.

In addition, `QuasilinearParticular` has to solve the system (5):

$$\begin{cases} f(x,y,z) & = & C_1 \\ g(x,y,z) & = & C_2 \\ h(x,y,z) & = & 0 \\ k(x,y,z) & = & 0 \end{cases}$$

to find a particular solution.

Examples:

- `Quasilinear(xy,-xy,(x-y)z,true)`

  to find the general solution of the quasilinear equation:

  $$xy(p-q) = (x-y)z$$

  showing the intermediate steps.

- `QuasilinearParticular(y,-x,0,x=0,z=y ˆ 2, true)`

  to find a particular of the quasilinear equation $yp - xq = 0$ which contains the curve Γ given by the intersection of plane $x = 0$ with $z = y^2$ showing the intermediate steps.

See section 4.2 for detailed executions of these examples.

3.1.3. Lagrange-Charpit method

- `LagrangeCharpit(F, comments_ := Stepwise)`

  to find a complete integral of the first order PDE: $F(x,y,z,p,q) = 0$ using the Lagrange-Charpit method.

  This program first checks if the PDE presents a particular case. If this is the case, the program solves the PDE attending to the particular case procedure. If not, the program will use the general method described in (2.3.1) to find out the solution.

Examples:

- `LagrangeCharpit(pxy+2pq+qy-yz,true)`

  to find, step by step, a complete integral of the PDE equation:

  $$pxy + 2pq + qy = yz$$

- `LagrangeCharpit(p ^ 2 - q ^ 2 - 1,true)`

  to find, step by step, a complete integral of the PDE equation:

  $$p^2 - q^2 = 1$$

- `LagrangeCharpit(pq-8xy,true)`

  to find, step by step, a complete integral of the PDE equation:

  $$pq = 8xy$$

- `LagrangeCharpit(p ^ 2+4px+4qy-4z-q ^ 2,true)`

  to find, step by step, a complete integral of the PDE equation:

  $$p^2 + 4px + 4qy = 4z + q^2$$

See section 4.3 for a detailed executions of these examples.

3.2. Syntax of auxiliary functions

In the process of solving some of the previous PDE, SFOPDES uses different auxiliary functions. The most important are those aimed to solve first order ODE. Therefore, SFOPDES can also be used to solve first order ODE.

As for the PDE, the step by step option will be on or off by setting to true or false the first optional parameter in the programs or the variable `Stepwise`.

The syntax to solve first order ODE are:

3.2.1. Specific types of ODE

- `Separable(P, Q, comments_ := Stepwise)`

  to find the solution of $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$ if it is a separable ODE. If the ODE is not separable, the program checks if there exists a function $f(x,y)$ such that $\dfrac{P(x,y)}{f(x,y)}\,\mathrm{d}x + \dfrac{Q(x,y)}{f(x,y)}\,\mathrm{d}y = 0$ is a separable ODE, and solves the new ODE.

Example: `Separable(x(1+y ^ 2),y(1+x ^ 2),true)`

to find, step by step, the solution of $x(1+y^2)\,\mathrm{d}x + y(1+x^2)\,\mathrm{d}y = 0.$

- `LineDependence(f, comments_ := Stepwise)`

  to find the solution of $y' = f(ax+by+c)$ with $a, b, c \in \mathbb{R}$.

  Example: `LineDependence((x+y-3) ^ 2,false)`

  to find the solution of $y' = (x+y-3)^2$ without showing the steps.

- `Homogeneous(P, Q, comments_ := Stepwise)`

  to find the solution of $P(x,y)\mathrm{d}x + Q(x,y)\mathrm{d}y = 0$ if it is a homogeneous ODE.

  Example: `Homogeneous(x ^ 2 - y ^ 2,2xy)`

  to find the solution of $(x^2-y^2)\,\mathrm{d}x + 2xy\,\mathrm{d}y = 0.$ The step by step option will depend on the value of `Stepwise`.

- `ReducibleToHomogeneous(P, Q, comments_ := Stepwise)`

  to find the solution of $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$ if it is reducible to a homogeneous ODE.

  Example: `ReducibleToHomogeneous(2x+y+1,x+2y-1,true)`

  to find, step by step, the solution of $(2x+y+1)\,\mathrm{d}x + (x+2y-1)\,\mathrm{d}y = 0.$

- `ExacT(P, Q, comments_ := Stepwise)`

  to find the solution of $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$ if it is an exact ODE.

  Example: `ExacT(2x+y+1,x+2y-1,true)`

  to find, step by step, the solution of $(2x+y+1)\,\mathrm{d}x + (x+2y-1)\,\mathrm{d}y = 0.$

- `IntegratingFactor(P, Q, comments_ := Stepwise)`

  to find an integrating factor for $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$ and solve the obtained exact ODE.

  Example: `IntegratingFactor(x + y ˆ 2, - 2xy, true)`

  to find an integrating factor for $(x+y^2)\,\mathrm{d}x - 2xy\,\mathrm{d}y = 0$ and solve the obtained exact ODE.

- `Linear(P, Q, comments_ := Stepwise)`

  to find the solution of the linear ODE $y' + P(x)\,y = Q(x)$.

  Example: `Linear(1/x,1,true)`

  to find, step by step, the solution of $y' + \dfrac{y}{x} = 1$.

- `Bernouille(P, Q, n, comments_ := Stepwise)`

  to find the solution of the Bernouille ODE $y' + P(x)\,y = Q(x)\,y^n$.

  Example: `Bernouille(-1/x,-1,2,true)`

  to find, step by step, the solution of $y' - \dfrac{y}{x} = -y^2$.

- `Riccati(P, Q, R, yp, comments_ := Stepwise)`

  to find the solution of the Riccati ODE $y' + P(x)\,y + Q(x)\,y^2 = R(x)$ with y_p a particular solution.

  Example: `Riccati(1/x,1,1/x ˆ 2,-1/x,true)`

  to find, step by step, the solution of $y' + \dfrac{y}{x} + y^2 = \dfrac{1}{x^2}$ with $y_p = -\dfrac{1}{x}$

3.2.2. Generic programs to solve ODE

- `DifferentialEquationPQ(P, Q)`

  to find the solution of the first order ODE $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$

  This program first checks the types of ODE and solves it. If needed, the program manipulates the equation trying to convert it to a known type. The program returns the different types of the equation together with the solution provided by each type.

Example: `DifferentialEquationPQ(2x+y+1,x+2y-1)`

to find the types and solution of $(2x + y + 1)\,\mathrm{d}x + (x + 2y - 1)\,\mathrm{d}y = 0$

- `ExplicitDifferentialEquation(f)`

  to find the solution of the first order ODE $y' = f(x, y)$

  As in the previous case, this program first checks the types of ODE and solves it. If needed, the program manipulates the equation trying to convert it to a known type. The program returns the different types of the equation together with the solution provided by each type.

  Example: `ExplicitDifferentialEquation(1-y/x)`

  to find the types and solution of $y' = 1 - \dfrac{y}{x}$

3.2.3. First order ODE of grade n

- `FirstOrderGradeN(r ˆ n + P1(x,y) r ˆ (n-1) + P2(x,y) r ˆ (n-2) + ··· + Pn(x,y), comments_ := Stepwise)`

  to find the solutions of the first order ODE of grade n:

  $$(y')^n + P_1(x,y)(y')^{(n-1)} + P_2(x,y)(y')^{(n-2)} + \cdots + P_n(x,y) = 0.$$

  Example: `FirstOrderGradeN(r ˆ 2 - (x+y) r + xy,true)`

  to find, step by step, the solutions of $(y')^2 - (x + y)y' + xy = 0.$

4. Examples of use

In order to show how SFOPDES solves step by step the PDE, we include in this section the execution of different examples. All these examples have been included in the file `SFOPDES.dfw` which is available at

`http://www.matap.uma.es/jlgalan/SFOPDES/`

The file `SFOPDES.dfw` is a tutorial of how to use SFOPDES. It contains the description and syntax of all programs (for solving ODE and PDE) and examples, step by step, of how to use them.

4.1. Examples of Pfaff equations

- Solve, step by step, the Pfaff equation $(3xz + 2y)\,\mathrm{d}x + x\,\mathrm{d}y + x^2\,\mathrm{d}z = 0$. If the general method is used, consider x as a constant parameter.

```
#   Pfaff(3xz+2y, x, x ˆ 2, true, x)
```

The equation is integrable since F·rot(F) = 0. The equation is not a particular case. Therefore, the general method must be used.

Let x be a constant parameter which leads to dx=0. Thus, the equation is: Q dy + R dz = 0 where Q = 1 and R = x which type is: SEPARABLE and its general solution is: C = xz + y. Therefore, G = xz + y.

Obtaining μ from any of the following two expressions:

∂(G,y) = μQ ; ∂(G,z) = μR we get μ = 1/x

The general solution of the Pfaff equation will be the general solution of the differential equation: dG + K dx = 0 where K(G,x) = μP - ∂(G,x). Operating we get: K = 2(xz + y)/x. Solving K in terms of G and x, we get: K = 2G/x

The type of the differential equation dG + K dx = 0 is SEPARABLE and its general solution is: 2LN(x) + LN(G) = C. Replacing G with its value in the previous expression, we get the general solution of the Pfaff equation:

```
#                              x^2(xz + y) = C
```

- Check that the Pfaff equation $x\mathrm{d}x + y\mathrm{d}y + xyz\mathrm{d}z = 0$ is not integrable.

```
#   Pfaff(x, y, xyz)

#       The equation is not integrable since F·rot(F) ≠ 0.
```

- Solve, step by step, the Pfaff equation $yz\,\mathrm{d}x + xz\,\mathrm{d}y + xy\,\mathrm{d}z = 0$.

```
#   Pfaff(yz, xz, xy, true)
```

The equation is integrable since F·rot(F) = 0. Dividing by: xyz we get an equation with separated variables and its solution is obtained by integrating each individual summands and equating it to an arbitrary constant C. In this case, the solution is

```
#                          xyz = C
```

Note that the program has manipulated the original equation to transform it in a particular case.

- Solve, step by step, the Pfaff equation $y^2\,\mathrm{d}x - z\,\mathrm{d}y + y\,\mathrm{d}z = 0.$

```
#   Pfaff(y ^ 2, -z, y, true)
```

The equation is integrable since F·rot(F) = 0. Dividing the equation by: y^2 we get an equation which has x as a separated variable and its solution is obtained by equating to an arbitrary constant the sum of the integral of P with respect to x and the potential function of Qdy+Rdz=0. In this case, the result is

```
                                 z
#                          x  +  -  = C
                                 y
```

Note that the program has manipulated the original equation to transform it in a particular case.

4.2. Examples of quasilinear equations

- Find the general solution of the quasilinear equation $xy(p-q) = (x-y)z$

```
#   Quasilinear(xy,-xy,(x-y)z,true)
```

The general solution of the quasilinear equation

P(x,y,z)p + Q(x,y,z)q = R(x,y,z)

is given by ϕ(f,g) = 0, with ϕ an arbitrary function, where f=C1 ; g=C2 is a curve congruence from the characteristic system:

$$\frac{\mathrm{d}x}{P(x,y,z)} = \frac{\mathrm{d}y}{Q(x,y,z)} = \frac{\mathrm{d}z}{R(x,y,z)}$$

Sometimes, in order to find f=C1 and g=C2 the following property is used which consist of equating the previous characteristic system to an expression of the form:

$$\frac{\alpha(x,y,z)\,\mathrm{d}x + \beta(x,y,z)\,\mathrm{d}y + \gamma(x,y,z)\,\mathrm{d}z}{\alpha(x,y,z)\,P(x,y,z) + \beta(x,y,z)\,Q(x,y,z) + \gamma(x,y,z)\,R(x,y,z)}$$

In this case, two first integrals are [x + y, xyz] obtained respectively from:

The first and second fractions of the characteristic system and using the integrable combination: $\alpha = 1/\mathrm{x}$; $\beta = 1/\mathrm{y}$; $\gamma = 1/\mathrm{z}$

Therefore, the solution is:

```
#         φ(x+y,xyz) = 0   with φ an arbitrary function
```

- Find, step by step, a particular solution of the quasilinear equation $yp - xq = 0$ which contains the curve Γ given by the intersection of plane $x = 0$ and $z = y^2$

```
#   QuasilinearParticular(y,-x,0,x=0,z=y^2, true)
```

In order to find the particular solution of a quasilinear equation that contains a given curve, the quasilinear equation is first solved, obtaining two first integrals f = C1; g = C2 that, together with the equations of the curve, provide us with the particular solution required. For this, from the system formed by f = C1, g = C2 and the equations of the curve, we must obtain an expression in which only constants intervene, eliminating the variables x, y, z. Once an expression of this type is obtained, the original equations f are replaced by C1 and g by C2, obtaining as a result the particular solution sought. In this case, the solution of the quasilinear equation leads to the following two first integrals:

(1) C1 = z

(2) C2 = x^2/2 + y^2/2

that, together with the curve equations:

(3) x=0

(4) z=(y^2)

constitute the system to be solved. From equations (1), (2) and (3) we obtain:

$\mathrm{x} = 0 \quad ; \quad \mathrm{y} = \sqrt{2}\,\sqrt{C2} \quad ; \quad \mathrm{z} = \mathrm{C1}$

When substituting these values in the equation (4) we get the relation between constants:

C1 = 2C2

Substituting C1 and C2 by their original values, we obtain the particular solution:

```
#                    z = x^2 + y^2
```

4.3. Examples of Lagrange-Charpit method

- Find, step by step, a complete integral of $pxy + 2pq + qy = yz$

```
#   LagrangeCharpit(pxy+2pq+qy-yz,true)
```

In order to solve a complete integral of the PDE F(x,y,z,p,q)=0, we use the Lagrange-Charpit Method which consists of the following steps:

1.- Find an integral $\phi(x, y, z, p, q) = C1$ from the characteristic system:

$$\frac{\mathrm{d}x}{\partial(F,p)} = \frac{\mathrm{d}y}{\partial(F,q)} = \frac{\mathrm{d}z}{p\partial(F,p)+q\partial(F,q)} = \frac{-\mathrm{d}p}{\partial(F,x)+p\partial(F,z)} = \frac{-\mathrm{d}q}{\partial(F,y)+q\partial(F,z)}$$

2.- Given the equations system:

F(x,y,z,p,q) = 0 ; ϕ(x,y,z,p,q)=C1

p and q are solved, obtaining the expressions:

p = p(x,y,z,C1) ; q = q(x,y,z,C1)

3.- The following Pfaff equation is solved:

dz = p(x,y,z,C1)dx + q(x,y,z,C1)dy

The general solution of this Pfaff equation, f(x,y,z,C1,C2)=0, provides a complete integral solution of the PDE F(x,y,z,p,q)=0.

Some of the previously described steps can be simplified if the equation F(x,y,z,p,q)=0 is any of the following particular cases:

1.- If the equation to solve presents the form F(p,q) = 0. In this case, the first step of the general method is reduced to set p = C1 or q = C1, easing, in addition, steps 2 and 3 of the general method.

2.- If the equation to solve presents the form g1(x,p) = g2(y,q). In this case, the two first steps of the general method are simplified getting p and q from: g1(x,p) = C1 and g2(y,q) = C1, easing, in addition, the last step of the general method.

3.- If the equation to solve presents the form z = px + qy + g(p,q). In this case, by direct substitution, a complete integral of the equation is
z = C1x + C2y + g(C1,C2)

The equation to solve is not a particular case. Therefore, it is solved using the general method. Following the steps of such general method, we obtain:

1.- From the fourth fraction we obtain the integral combination: C1=p

2.- From this relation and considering the equation to be solved, we obtain:

p=C1 ; q = y(z - C1x)/(y + 2C1)

3.- Solving the Pfaff equation dz = pdx + qdy, we obtain the complete integral:

```
#              LN(z - C1x) + 2C1LN(y + 2C1) - y = C2
```

In the following examples, we set the variable `Theory` to false so that the theory of Lagrange-Charpit Method is no longer displayed.

```
#   Theory := false
```

- Find, step by step, a complete integral of $p^2 - q^2 = 1$

```
#   LagrangeCharpit(p ^ 2 - q ^ 2 - 1,true)
```

The equation to solve presents the first particular case. Therefore, letting p = C1 and substituting in the equation to solve, we obtain:

$$\mathrm{p} = \mathrm{C1} \quad ; \quad \mathrm{q} = \sqrt{\mathtt{C1}^2-1}$$

Solving the Pfaff equation dz = pdx + qdy, we obtain the complete integral:

$\quad\quad \mathtt{C1x\ +\ y}\sqrt{\mathtt{C1}^2-1}\ \mathtt{-\ z\ =\ C2}$

- Find, step by step, a complete integral of $pq = 8xy$

```
#   LagrangeCharpit(pq-8xy,true)
```

The equation to solve presents the second particular case. Therefore, letting C1 = p/x and substituting in the equation to solve, we obtain:

$$p = C1x \quad ; \quad q = 8y/C1$$

Solving the Pfaff equation dz = pdx + qdy, we obtain the complete integral:

$$\frac{C1x^2}{2} + \frac{4y^2}{C1} - z = C2$$

Note that the program has manipulated the original equation to transform it in a particular case.

- Find, step by step, a complete integral of $p^2 + 4px + 4qy = 4z + q^2$

```
#   LagrangeCharpit(p^2+4px+4qy-4z-q^2,true)
```

The equation to solve presents the third particular case. Therefore, by direct substitution, we obtain the complete integral:

$$4C1x + 4C2y - 4z + C1^2 - C2^2 = 0$$

Note that the program has manipulated the original equation to transform it in a particular case.

5. Conclusions and future work

In this section, the conclusions of this work are described. Also, some ideas for future related work are presented.

5.1. Conclusions

In this work, SFOPDES, a new Stepwise First Order Partial Differential Equation Solver is introduced. The main novelty of SFOPDES is the possibility of providing the solutions of PDE step by step.

The types of equations that SFOPDES solves are:

- Pfaff equations (general method and some particular cases).
- Quasilinear first order partial differential equations (general solution and particular solutions) and

- General first order partial differential equations (complete integral, using the Lagrange-Charpit general method and some particular cases).

Since the solution of PDE requires the solution of ODE, SFOPDES also can be used as a stepwise first order ordinary differential equations solver. Specifically, the ODE that can be solved are:

- Separable equations and equations reducible to them.
- Homogeneous equations and equations reducible to them.
- Exact differential equations and equations reducible to them (integrating factor technique).
- Linear equations.
- The Bernoulli equation.
- the Riccati equation.
- Generic programs to solve first order ordinary differential equations, that is, generic programs which determine the type of an ODE and provides its general solution.
- First-order differential equations and nth degree in y'.

SFOPDES has been developed using the CAS DERIVE and its code can be freely downloaded at `http://www.matap.uma.es/jlgalan/SFOPDES/` or by request to the corresponding author at `jlgalan@uma.es`. The following three files can be downloaded in a zip file:

1. `SFOPDES.mth` which is the executable code to run the solver SFOPDES in DERIVE.
2. `SFOPDES_code.pdf` which is a pdf file with the code of SFOPDES.
3. `SFOPDES.dfw` which is a tutorial, developed in DERIVE, on how to use SFOPDES.

The optional stepwise option (set to on by default) makes SFOPDES be also a very useful tool for teaching a partial differential equation course for Engineering and Sciences degree since it can be used as a tutorial for the students. The tutorial `SFOPDES.dfw` included can be used to help in the teaching and learning process of PDE.

5.2. Future work

Directly related with the work developed in this paper, we will work in the following extensions:

- SFOPDES is a stepwise solver for first order partial differential equations. We will work also in solving second order partial differential equations.
- SFOPDES includes a solver for first order ordinary differential equations. We will develop a new stepwise solver for differential equations of first and higher order.
- SFOPDES has been developed using the CAS DERIVE but we will migrate it to SAGE and PYTHON and develop a web application so that SFOPDES will be able to be used freely on-line. The use of free software will allow all users to access the solver and different platforms will be available (since the proprietary software DERIVE only can be used in Windows system).

This work is part of a research line which consists of the development of libraries for different Computer Algebra Systems. As a future work, we are interested in improving our libraries by developing web applications in order to provide new stepwise solvers for different topics in Mathematic and Engineering. For example, among other topic, we will develop stepwise solvers for:

- Line, multiple and surface integrals computations.
- Automatic theorem provers.
- Improper integrals.
- Complex analysis exercises.

- Random variable generation.

Acknowledgements

We thank the anonymous reviewers for their suggestions and comments which have improved the quality of the paper.

References

[1] J. L. Galán–García, G. Aguilera–Venegas, M.Á. Galán–García, P. Rodríguez-Cielos, I. Atencia-Mc.Killop: Improving CAS capabilities: New rules for computing improper integrals. Appl. Math. Comput. 316 (2018) 525–540. `https://dx.doi.org/10.1016/j.amc.2016.12.024`

[2] A. D. Rich, A Brief History of the muMATH/DERIVE Cass, DERIVE Newsletter 40 (2000) 5.

[3] S. Wolfram, Computer algebra: a 32-year update, in: ISSAC '13 Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation (2013) 7–8.

[4] Bruce W. Char, Gregory J.Fee, Keith O.Geddes, Gaston H.Gonnet, Michael B. Monagan, A tutorial introduction to Maple, Journal of Symbolic Computation 2 (2) (1986) 179–200. `https://dx.doi.org/10.1016/S0747-7171(86)80021-9`

[5] Maxima, `http://maxima.sourceforge.net/` (accessed February 3, 2019).

[6] W. Stein, Sage: Creating a Viable Free Open Source Alternative to Magma, Maple, Mathematica, and MATLAB. In F. Cucker, T. Krick, A. Pinkus, and A. Szanto (Eds.), Foundations of Computational Mathematics, Budapest 2011 (London Mathematical Society Lecture Note Series, pp. 230-238). Cambridge: Cambridge University Press. `https://dx.doi.org/10.1017/CBO9781139095402.011`

[7] SageMath, `http://www.sagemath.org/` (accessed February 3, 2019).

[8] G. Aguilera-Venegas, J. L. Galán-García, M. Á. Galán-García, P. Rodríguez-Cielos, Teaching semantic tableaux method for propositional classical logic with a CAS, International Journal for Technology in Mathematics Education 22 (2) (2015) 85–92.

[9] G. Aguilera-Venegas, J. L. Galán-García, M.Á. Galán-García, G. Lobillo-Mora, J. Martínez-del-Castillo, S. Merino-Córdoba, Y. Padilla-Domínguez, P. Rodríguez-Cielos, R. Rodríguez-Cielos: Parametrization of curves and line integrals with a CAS. International Journal for Technology in Mathematics Education 24 (4), 179–190 (2017).

[10] G. Aguilera, J. L. Galán, R. Madrid, A. M. Martínez, Y. Padilla, P. Rodríguez, Automated generation of contrapuntal musical compositions using probabilistic logic in derive, Journal of Mathematic and Computer in Simulation 80 (6) (2010) 1200–1211. `https://dx.doi.org/10.1016/j.matcom.2009.04.012`

[11] G. Aguilera, J. L. Galán, J. M. García, E. Mérida, P. Rodríguez, An accelerated-time simulation of car traffic on a motorway using a CAS, Journal of Mathematic and Computer in Simulation 104 (2014) 21–30. `https://dx.doi.org/10.1016/j.matcom.2012.03.010`

[12] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T, Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, S., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., Scopatz,A., SymPy: Symbolic computing in Python. PeerJ Computer Science 3:e103 (2017). `https://doi.org/10.7717/peerj-cs.103`

[13] Ye Zhao, Lattice Boltzmann based PDE solver on the GPU. Visual Computer 24 (5) (2008). 323–333. `https://doi.org/10.1007/s00371-007-0191-y`

[14] Kurganov, A. Central Schemes: A Powerful Black-Box Solver for Nonlinear Hyperbolic PDEs. Handbook of Numerical Analysis 17 (2016). 525–548. `https://doi.org/10.1016/bs.hna.2016.09.008`

[15] MatLab, pdepe, a built-in PDE solver in MatLab. `https://www.mathworks.com/help/matlab/ref/pdepe.html` (accessed March 6, 2019)

[16] Hecht, F. FreeFem++, a finite element PDE solver. ECCOMAS 2004 - European Congress on Computational Methods in Applied Sciences and Engineering (2004).

[17] Marcelo Epstein, Partial Differential Equations Mathematical Techniques for Engineers. Springer International Publishing, first ed., Switzerland, 2017. ISBN 978-3-319-55211-8. `https://doi.org/10.1007/978-3-319-55212-5`

[18] Sandro Salsa, Partial Differential Equations in Action. From Modelling to Theory. Springer International Publishing, third ed., Switzerland, 2016. ISBN 978-3-319-31237-8. `https://doi.org/10.1007/978-3-319-31238-5`

[19] Luigi Pepe, Lagrange (1736–1813): a life in mathematics, Lettera Matematica 2 (1) (2014) 3–8. `https://dx.doi.org/10.1007/s40329-014-0038-0`

[20] Manuel Delgado, The Lagrange-Charpit Method, SIAM Review 2 (1997) 298–304. `https://doi.org/10.1137/S0036144595293534`

[21] G. Aguilera, J.L. Galán, M.Á. Galán, Y. Padilla, P. Rodríguez, R. Rodríguez. FOPDE.mth: Solving First-Order Partial Differential Equations with Derive 6 step by step. In proceedings of the 19th Conference on Applications of Computer Algebra - ACA 2013, J.L. Galán-García, G. Aguilera-Venegas and P. Rodríguez-Cielos (eds.), 237–237. Málaga (Spain), 2013, ISBN 978–84–616–4565–7.

[22] J.L. Galán-García, P. Rodríguez-Cielos, Y. Padilla-Domínguez, M.Á. Galán-García, G. Aguilera-Venegas, R. Rodríguez-Cielos. Teaching Partial

Differential Equations with CAS. In proceedings of the 24th Conference on Applications of Computer Algebra - ACA 2018, Francisco Botana, Felipe Gago and Manuel Ladra (eds.), 64–65. Servizo de Publicacións e Intercambio Científico Campus Vida, Santiago de Compostela (Spain), 2018, ISBN 978–84–16954–87–2.