Journal of Computational and Applied Mathematics 428 (2023) 115127

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

SODES: Solving ordinary differential equations step by step

José Luis Galán–García [a,*], Pedro Rodríguez–Cielos [a], María Ángeles Galán–García [a], Morgan le Goff [b], Yolanda Padilla–Domínguez [a], Pablo Rodríguez-Padilla [a], Iván Atencia [a], Gabriel Aguilera–Venegas [a]

[a] *Department of Applied Mathematics. University of Málaga, Spain*
[b] *Génie Mathématiques et Modélisation. Polytech Clermont Ferrand Université Clermont Auvergne, France*

A R T I C L E I N F O

Article history:
Received 31 October 2022
Received in revised form 26 January 2023

Keywords:
Ordinary Differential Equations (ODE)
ODE solvers
Computer Algebra Systems (CAS)
Programming with CAS
Stepwise solutions

A B S T R A C T

In this paper, we introduce SODES (Stepwise Ordinary Differential Equations Solver) a new solver for Ordinary Differential Equations (ODE). SODES can optionally provide the solution displaying all the steps needed to obtain it. This way, SODES is an important tool not only for researchers who need solving ODE but also constitutes an important tool for the teaching and learning process of ODE. SODES has been developed using programming with a Computer Algebra System (CAS). Specifically, we use the CAS Derive but it can be easily adapted to any other CAS supporting programming.

SODES provides, step by step, the solution of the following types of ODE: separable, homogeneous, exact, integrating factors, linear, Bernoulli, Riccati, first order ODE of *n*th degree, Cauchy's problems of first order ODE, higher order linear homogeneous equations with constant coefficients, Lagrange's method for particular solutions of higher order linear equations with constant coefficients, higher order linear equations with constant coefficients and Cauchy's problems of higher order linear equations with constant coefficients. SODES also deals with two generic programs which determine the type or types of a given ODE and provides the solution.

In this paper we will also introduce a draft of a Graphical User Interface (GUI) for SODES in a local web application using programming in Python (using its CAS module SymPy) which is a more portable and free CAS. This draft can be used in English, French and Spanish, and can be easily extended to other languages.

The code of SODES and the GUI are freely available so that it can be used by users who also will be able to adapt it to their needs.

1. Introduction

Ordinary Differential Equations (ODE) appears in many applications in Engineering and Sciences. Therefore, ODE is a very important topic to deal with for researchers and within the curriculum of Engineering and Sciences students.

In this paper we introduce a Stepwise Ordinary Differential Equations Solver (SODES) using a Computer Algebra System (CAS).

* Corresponding author.
E-mail addresses: jlgalan@uma.es (J.L. Galán–García), prodriguez@uma.es (P. Rodríguez–Cielos), magalan@ctima.uma.es (M.Á. Galán–García), morgan–legoff@live.fr (M. le Goff), ypadilla@ctima.uma.es (Y. Padilla–Domínguez), rodriguezpadillapablo@uma.es (P. Rodríguez-Padilla), iatencia@ctima.uma.es (I. Atencia), gabri@ctima.uma.es (G. Aguilera–Venegas).

https://doi.org/10.1016/j.cam.2023.115127
0377-0427/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

The main novelty is that SODES does not only provide the solution to an ordinary differential equation problem but also it can display the intermediate steps and explanations of how to obtain the solution. This way, SODES can be used in two different ways: firstly, SODES is a stepwise solver for ODE and secondly, it can be used as a tutorial for the teaching and learning process of this important topic in Engineering and Sciences helping both, the teacher and the student. Furthermore, with the stepwise option, SODES can be used as a self tutorial for researchers and students.

SODES has been developed using a CAS, specifically DERIVE [1]. A CAS is a mathematical software with two important differences from other mathematical software:

1. It deals with exact computations not with approximations.
   For example, the result of $\sum_{n=1}^{\infty} \frac{1}{n^2}$ using a CAS would be $\frac{\pi^2}{6}$, while using other mathematical software, an approximation of $\frac{\pi^2}{6}$, would be provided.
2. A CAS can handle algebraic expressions without the need of instantiating the values of the parameters involved.
   For example, the expanded result of $(a - bc)^4$ using a CAS is $a^4 - 4a^3bc + 6a^2b^2c^2 - 4ab^3c^3 + b^4c^4$, while using other mathematical software the values of a, b and c should be provided in order to get the final result.

These two features of CAS allow us to deal with generic parameters and with exact computations which will lead to a more widely use of the developed solver SODES.

In addition, using a CAS to develop SODES, we have been able to program the intermediate steps and add any text information on each step. This fact facilitates the use of SODES as a Pedagogical CAS (PeCAS [2]) since the solution can be obtained step by step and with information along all the process to reach the final result.

We have chosen the CAS DERIVE to develop SODES. The CAS to use is very important for SODES since not all CAS can deal with the needed computations. Specifically, we need, among others, to deal with these very important two topics that not all CAS support:

1. Since all ODE need to compute integrals symbolically in the process of resolutions, we need to use a CAS which supports computing primitives of functions symbolically and not with approximations (as a numerical software would do). This way, the developed programs in SODES will provide exact solutions to ODE instead of approximations.
2. In order to find particular solutions in Cauchy's problems and for linear ODE, we need to use a CAS which can handle systems of equations (where the unknowns can be symbolic functions) to provide the particular solutions needed to reach the final solutions in some ODE and Cauchy's problems.

The main goal of this paper is the introduction of the new stepwise solver SODES with the aim of serving as an important tool for researchers who need to deal with ODE. Another important goal of this paper is to increase the capabilities of the CAS DERIVE in particular and for other CAS in general (adapting the developed programs to the specific syntax of other CAS).

In addition, SODES can also be used in educational environments.

In the following sections we will describe SODES starting with Section 2 where the previous authors' backgrounds and some related works available worldwide are stated. Section 3 is devoted to describe SODES showing one by one all types of ODE available in SODES with the corresponding syntax and examples of use and executions. In Section 4 a prototype of Graphical User Interfaces (GUI) for SODES is introduced. Finally, Section 5 will describe the conclusions of the paper together with some ideas for future lines of related works.

2. Backgrounds and related works

Computer Algebra Systems (CAS) have been broadly used since their beginnings [3–5].

There has been an exponential increase in both: development of new CAS and the topics they can handle. In addition, nowadays it is a fact that many CAS include the capability of using their specific programming language to develop new applications. Normally, the available CAS do not have specific functions to deal with many mathematical topics or, in case a CAS can solve advances topics, the obtained solution is showed directly without any explanations of the intermediate steps to reach the solution. To achieve this lacks of dealing with advanced topics, programming new functions can be an interesting approach.

However, we are interested not only in programming new functions which allow CAS to deal with advanced mathematical topics and applications but also in programming with didactical approaches providing stepwise solutions and adding comments along all the process to obtain the final solution.

In the following subsections we describe the backgrounds of the authors (Section 2.1) and discuss some of the already works available worldwide (Section 2.2) on this topic of improving the capabilities of CAS providing stepwise solutions.

2.1. Author's backgrounds

The authors have been using CAS in research and in education for the last 25 years. The authors have developed most of the works using the CAS DERIVE [1]. The main reasons are:

- DERIVE has a very flexible syntax which make it easy to be used with few syntax errors.
- DERIVE has a very well designed menu line (with many shortcuts to frequent functions).
- DERIVE has a very wide number of available functions in its kernel and, in addition, many libraries can be loaded to increase the CAS capabilities.
- DERIVE supports new libraries built by users. Furthermore, DERIVE has a specific folder with libraries built by users all over the world.
- DERIVE has a international user group (Derive User Group [6]).
- Since the very beginnings of DERIVE, The Derive Newsletter [7] is a quarterly journal which supports publications of works dealing with DERIVE although in the last years also works with other CAS are considered.
- In addition, DERIVE is the CAS that the authors of this paper use in the computer lectures with engineering students because of its flexible syntax, which makes it easy for beginner students.

The authors are experts in DERIVE in the development of new libraries to increase the capabilities of the CAS and in the development of applications to different fields in Engineering, Science and Education. In addition, the authors have also develop different applications using other CAS, such as MAPLE [8], MAXIMA [9] or CoCoA [10,11] in combination with the development of Graphical User Interfaces (GUI) to display the results of the applications. The GUIs were developed using the programming language JAVA [12] or PHP + HTML [13] so that the applications can be run in different platforms.

The previous works developed by the authors can be grouped in two different fields:

1. Development of new libraries to improve the capabilities of the CAS.
2. Development of applications and GUIs to display the results.

The work developed in this paper, the introduction of the new solver SODES, can be classified in both fields since it is a new library which provides stepwise solutions of different types of ODEs, increasing this way the capabilities of the CAS DERIVE, and a GUI is being developed to ease the user to work with SODES. This prototype of GUI works locally in a web browser and it is described in Section 4.

It is important to remark that most of the previous developed works can be easily adapted to be used in different CAS, since in all cases we provide the code for the specific library or application developed.

In addition, as we will remark in the future work Section 5, we are migrating the different libraries and applications using the programming language PYTHON [14,15] using its CAS library SYMPY [16,17] which will provide more portable applications since PYTHON is a multi-platform programming language. Furthermore, the prototype of GUI considered for SODES is being developed in HTML + PYTHON.

The following is a non-exhaustive list of previous related works developed by the authors.

2.1.1. *Applications combining CAS and GUI*

- In [18] we introduced for the first time the combination of using programming with a CAS (specifically, DERIVE) and the development of a GUI (using JAVA). In this work we provide a counterpoint for a given melody (*cantus firmus*) introduced by the user in a virtual keyboard using the GUI.
- In [19] we developed the GRAM model to help in the control of traffic in a motorway. The model was implemented using the CAS DERIVE and a GUI developed in the program language JAVA, which shows graphically the accelerated-time simulations of the movement of vehicles in a motorway.
- In [20] we introduced the model ATISMART, an accelerated-time simulator for traffic control in a smart city. ATISMART was developed using the CAS MAXIMA together with a GUI in JAVA. In [21] we introduced the improved model ATISMART$^+$ (also using MAXIMA + JAVA) including a new probabilistic extension of Dijkstra's algorithm, which provides more realistic simulations for traffic control in smart cities.
- In [22] an accelerated-time simulator for controlling the baggage traffic in an airport terminal was introduced. Again, the combination MAXIMA + JAVA was considered for developing the algorithms of the model and the GUI respectively.
- In [23] we developed a knowledge-based system for helping a car driver when a dashboard light goes on. In this case, the CAS used is CoCoA and the GUI is developed in PHP + HTML.
- In [24] we introduced the PCAEGOL model, a accelerated-time simulator for cells using a probabilistic cellular automata. As an example of applications we applied PCAEGOL to describe the evolution of cancer cells.

2.1.2. *Improving CAS capabilities*

- In [25] we developed a library in the CAS DERIVE for automatic theorem proving using the Semantic Tableaux method [26].
- In [5,27] we developed new rules for computing improper integrals. These rules can be introduced in all CAS which support programming, improving this way their capabilities.
- In [28] we developed SMIS, a stepwise solver for multiple integration in DERIVE.
- [29] can be considered the main related work to the one introduced in this paper. We introduced SFOPDES, a stepwise solver for first order partial differential equations. SFOPDES needs to solve ODE, therefore, in this previous work we developed some programs to deal with ODE but only obtained the final result. In this work, we extend the programs developed in SFOPDES related with ODE in two different ways: we introduce more types of ODE to solve and we provide stepwise results.

2.2. Related works

In this section we describe some of the nowadays available tools related with the work we introduce in this paper.

2.2.1. Online applications

There are lots of online applications which deals with solving some ODE. The following is not an exhaustive list but a list of the most popular and most used ones:

- WolframAlpha [30] which uses Wolfram Language [31] built initially for the CAS Mathematica [32]. This is a very complete online site for different mathematics topics but for advanced uses and stepwise solutions requires the non-free pro version.
- MapleCloud [33] is the online version of part of the CAS Maple [8]. It does not provide stepwise solutions.
- Symbolab [34] is also another interesting website which deals with different mathematics topics but it requires a non-free pro version for advanced and stepwise solutions.
- MathDF [35] provides stepwise solutions for the different types of ODE that it can handle. Our approach provides extended explanations in the stepwise solutions. Moreover, at it will be stated in the description of SODES, since we provide the code, the user can easily adapt the stepwise solution to the specific needs.
- Mathforyou [36] is a website designed for students to help them in the study of different mathematics topics by solving problems. It has different and specific calculators to deal with the different topics they can handle. Specifically, the calculator dealing with ODE can be found at [37].
- ΣMh (eMathHelp) [38] can deal with different calculators related with ODE such as Laplace and Inverse Laplace calculators, ODE calculators and calculators with numeric approaches using different methods. Regarding the ODE calculator, only the final solution is provided without the option of getting a stepwise solution.
- Math24.pro [39] provides not very exhaustive stepwise solutions for the different types of ODE it can handle. In addition, if the ODE to be solved belongs to different types, the solution provides these types and the user can chose the solution method.
- snapXam [40] is another online site which solves different types of ODE but the stepwise option requires the non-free pro version.

2.2.2. Commercial CAS

Among the commercial CAS available nowadays which deals with ODE the most widely used are Mathematica [32] and Maple [8].

- Mathematica has an online solver (WolframAlpha [30]) already mentioned in the previous section. The main problem is that the pro version is required to obtain stepwise solutions.
- Maple has a solver for differential equations which can be downloaded at [41]. It has an online version available at [42], but it does not provide stepwise solutions.

2.2.3. Free CAS

Maxima [9] and SageMath [43] are probably the most widely used free CAS in Mathematics. The increasing use of Python [15] and its powerful CAS library Sympy [17] makes this combination be one of the best free option used nowadays. Even more, other applications are extending their capabilities by allowing using programming in Python in their systems.

- Maxima deals with ordinary differential equations but it does not provide stepwise solutions. A tutorial on how to solve ODE can be found at [44]
- The topics that the CAS library Sympy of Python can handle are continuously increasing. Specifically, it has a very important module for solving ODE. Its description and examples of use can be found at [45]. It does not provide stepwise solutions.
- SageMath also can solve differential equations but it uses the programs from Maxima. Therefore, the same solutions are obtained as with Maxima (with no stepwise solutions). A tutorial is available at [46]. In addition, SageMath incorpores the CAS library Sympy and, therefore, it can provide the same solutions provided by Sympy.

3. Description of SODES

SODES is a stepwise ordinary differential equations solver which can solve step by step (optionally) the following types of ODE:

- Separable.
- Homogeneous.
- Exact.
- Integrating factors.
- Linear.

- Bernoulli.
- Riccati.
- First order ODE of nth degree.
- Higher order linear equations with constant coefficients.
- Cauchy's problems of higher order linear equations with constant coefficients.

In addition, SODES also deals with two generic programs which determines the type or types of a given ODE and provides the solution.

In the following subsections, we describe the programs within SODES by means of their syntax, their description and examples of use.

The library `SODES.mth` developed in Derive containing all the programs of SODES can be freely downloaded at http://u.uma.es/c4U/SODES/. In addition, in the same link, the tutorial `SODES.dfw` can also be downloaded which shows different examples of use.

3.1. Separable equations

- **Function name**: `Separable`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It solves $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$ if it is a separable equation. If the ODE is not separable, the program looks for a function $f(x,y)$ such that $\frac{P(x,y)}{f(x,y)}\,\mathrm{d}x + \frac{Q(x,y)}{f(x,y)}\,\mathrm{d}y = 0$ is a separable ODE, in which case, it solves the new ODE.

Example: Solve $x(1+y^2)\,\mathrm{d}x + y(1+x^2)\,\mathrm{d}y = 0$ step by step.

Solution with Derive

`Separable(x(1+y`^{2}`),y(1+x`^{2}`),true)`

$P(x,y)\mathrm{d}x + Q(x,y)\mathrm{d}y = 0$ is a separable equation if there exists a function $f(x,y)$ such as $P(x,y)/f(x,y)dx + Q(x,y)/f(x,y) = 0$ is a equation with both variables isolated.

In this case, dividing by $(x^2+1)(y^2+1)$ we obtain the equation:

$(x/((x^2)+1))\mathrm{d}x + (y/((y^2)+1))\mathrm{d}y = 0$

which has both variables isolated. Thus, the solution of this equation is:

$$\frac{\mathrm{LN}((x^2+1)\cdot(y^2+1))}{2} = C$$

3.2. Line-dependent equations

- **Function name**: `LineDependence`
- **Arguments**: `f`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It solves $y' = f(ax+by+c)$ with $a,b,c \in \mathbb{R}$.

Example: Solve $y' = (x+y+3)^2$ step by step.

Solution with Derive

`LineDependence((x + y + 3)`^{2}`),true)`

This equation depends on the line $x+y+3$. Thus, by means of the variable change $z = x+y+3$, a separable equation is obtained. Solving this separable equation and considering back the variable change, the following general solution is obtained:

$$\mathrm{ATAN}(x+y+3) - x = -C$$

3.3. Homogeneous equations

- **Function name**: `Homogeneous`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.

- **Description**: It checks if the equation $P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$ is an homogeneous one. If so, it provides the solution.

Example: Solve $(x^2 - y^2)\,\mathrm{d}x + 2xy\,\mathrm{d}y = 0$ step by step.

Solution with Derive

`Homogeneous(x²-y², 2xy,true)`

$f(x, y)$ is an homogeneous function of grade k if $f(ax, ay) = a^k f(x, y)$.

P(x,y) dx + Q(x,y) dy = 0 is an homogeneous differential equation if P and Q are homogeneous functions of same grade.

In this case, the differential equation is an homogeneous one. Therefore, by means of the variable change $y = tx$ ($\mathrm{d}\,y = t\mathrm{d}\,x + x\mathrm{d}\,t$) we obtain a separable differential equation. After solving this separable equation and considering back the variable change, we obtain:

$$\mathrm{LN}\left(\frac{x^2 + y^2}{x}\right) = C$$

3.4. Reducible to homogeneous equations

- **Function name**: `ReducibleToHomogeneous`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It checks if the equation $P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$ is reducible to an homogeneous ODE. If so, it provides the solution.

Example: Solve $(2x + y + 1)\,\mathrm{d}x + (x + 2y - 1)\,\mathrm{d}y = 0$ step by step.

Solution with Derive

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

Both lines intersects in point $(-1, 1)$. Therefore, by means of translation $x = X - 1$; $y = Y + 1$ we obtain an homogeneous differential equation. Solving this equation and considering back the variable changes, we obtain:

$$\mathrm{LN}\left(x^2 + x(y + 1) + y^2 - y - 1\right) = 2C$$

3.5. Exact equations

- **Function name**: `ExacT`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It checks if the equation $P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$ is an exact ODE. If so, it provides the solution.

Example: Solve $(2x + y + 1)\,\mathrm{d}x + (x + 2y - 1)\,\mathrm{d}y = 0$ step by step.

Solution with Derive

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

$P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$ is an exact differential equation if $\partial(P, y) = \partial(Q, x)$. In this case: $\partial(P, y) = 1$ and $\partial(Q, x) = 1$. Since the differential equation is an exact one, its general solution is $U(x, y) = C$ where

$U(x, y) = \int_0^x P(t, 0)\,\mathrm{d}t + \int_0^y Q(x, t)\,\mathrm{d}t$ is the potential function of the exact differential. That is:

$$x^2 + x(y + 1) + y(y - 1) = C$$

3.6. Integrating factors

- **Function name**: `IntegratingFactor`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It looks for an integrating factor for $P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$. If an integrating factor is found, it provides the solution of the exact differential equation obtained.

Example: Find and integration factor for $(x+y^2)\,\mathrm{d}x - 2xy\,\mathrm{d}y = 0$ and solve the equation step by step.

Solution with DERIVE

`IntegratingFactor(x + y`^{2}`, - 2xy, true)`

A function $\mu(x,y)$ is an integrating factor for a given differential equation $P(x,y)\,\mathrm{d}x + Q(x,y)\,\mathrm{d}y = 0$, if the equation $\mu(x,y)P(x,y)\,\mathrm{d}x + \mu(x,y)Q(x,y)\,\mathrm{d}y = 0$ is an exact differential equation.

This differential equation admits the following integrating factor:
$\mu(x,y) = 1/(x^2)$.

Thus, multiplying the original equation by such integrating factor, the following exact differential equation is obtained:
$((x+(y^2))/(x^2))\,\mathrm{d}x + (-(2y)/x)\,\mathrm{d}y = 0$

Solving this exact differential equation the following general solution is obtained:

$$\mathrm{LN}(x) - \frac{y^2}{x} = C$$

3.7. Linear equations

- **Function name**: `Linear`
- **Arguments**: `P, Q`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the linear ODE $y' + P(x)y = Q(x)$.

Example: Solve $y' + \frac{y}{x} = 1$ step by step.

Solution with DERIVE

`Linear(`$\frac{1}{x}$`, 1, true)`

A differential equation is linear if it presents the following form: $y' + p(x)y = q(x)$. The solution of a linear equation is given by $y = y_h + y_p$ where y_h is the general solution of the homogeneous associated equation: $y' + p(x)y = 0$, which it is a separable equation, and y_p is a particular solution of the linear equation which can be found using the constant variation method.

In this case, this is a linear differential equation which homogeneous associated equation is $y' + \frac{y}{x} = 0$, which general solution is: $y = \frac{C}{x}$.

A particular solution of the linear equation is given by: $y = \frac{x}{2}$.

Thus, the general solution of this linear differential equation is:

$$y = \frac{x^2+2C}{2x}$$

3.8. Bernoulli equations

- **Function name**: `Bernoulli`
- **Arguments**: `P, Q, n`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the ODE $y' + P(x)y = Q(x)y^n$.

Example: Solve $y' - \frac{y}{x} = -y^2$ step by step.

Solution with DERIVE

`Bernoulli(`$-\frac{1}{x}$`, −1, 2, true)`

A differential equation is a Bernoulli's one if it presents the following form: $y' + p(x)y = q(x)y^n$.

By means of the variable change $z = y^{(1-n)} = \frac{1}{y^{n-1}}$ a linear differential equation is obtained. Solving this linear equation and considering back the change of variable, the following general solution is obtained:

$$\frac{1}{y} = \frac{x^2+2C}{2x}$$

3.9. Riccati equations

- **Function name**: `Riccati`
- **Arguments**: `P, Q, R, yp`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the ODE $y' + P(x)y + Q(x)y^2 = R(x)$ where yp is a particular solution.

Example: Solve $y' + \frac{y}{x} + y^2 = \frac{1}{x^2}$ step by step with $yp = -\frac{1}{x}$.

Solution with Derive

`Riccati(`$\frac{1}{x}$, 1, $\frac{1}{x^2}$, $-\frac{1}{x}$, `true)`

A differential equation is a Riccati's one if it presents the following form: $y' + p(x)y + q(x)y^2 = r(x)$.

In order to solve a Riccati differential equation the following variable change can be done: $y = z + y_p$ where y_p is a particular solution.

In this case: by means of the variable change: $y = z - \frac{1}{x}$ a Bernoulli differential equation is obtained. Solving such Bernoulli equation, considering back the variable change and taking into account the singular solution, the following singular and general solutions are obtained:

$$\left[\texttt{Singular solution}, y = -\frac{1}{x}, \texttt{General solution}, y = \frac{2x}{x^2+2C} - \frac{1}{x}\right]$$

3.10. Generic programs to solve ODE

- **Function name**: `DifferentialEquationPQ`
- **Arguments**: `P, Q`
- **Description**: It classifies the equation $P(x, y)\,\mathrm{d}x + Q(x, y)\,\mathrm{d}y = 0$ in different types. If needed, it manipulates the equation in order to find a specific type of ODE. The program provides all types found together with the corresponding solutions.

Example: Classify the ODE $x\,\mathrm{d}x + y\,\mathrm{d}y = 0$ and solve the equation.

Solution with Derive

`DifferentialEquationPQ(x, y, true)`

$$\begin{bmatrix} \texttt{Type} & & \texttt{Solution :} \\ \texttt{SEPARABLE EQUATION} & & \dfrac{x^2}{2} + \dfrac{y^2}{2} = C \\ \texttt{EXACT} & & \dfrac{x^2}{2} + \dfrac{y^2}{2} = C \\ \texttt{HOMOGENEOUS} & & \mathrm{LN}\left(x^2+y^2\right) = 2C \\ \texttt{INTEGRATING FACTOR} & \mu = 1 & \dfrac{x^2}{2} + \dfrac{y^2}{2} = C \\ \texttt{BERNOULLI} & & y^2 = C - x^2 \end{bmatrix}$$

We introduce now another general program to find out the type or types of an ODE:

- **Function name**: `ExplicitDifferentialEquation`
- **Arguments**: `f`
- **Description**: It finds the types and solution of the ODE $y' = f(x, y)$.

Example: Classify the equation $y' = 1 - \frac{y}{x}$ and find the solution.

ExplicitDifferentialEquation(1-y/x, true)

$$\left[\begin{array}{ccc} \text{Type} & & \text{Solution :} \\ \text{HOMOGENEOUS} & & \dfrac{\text{LN}\left(\dfrac{2y-x}{x}\right)}{2} + \text{LN}(x) = C \\ \text{INTEGRATING FACTOR} & \mu = x & \dfrac{x^2}{2} - xy = C \\ \text{LINEAR} & & y = \dfrac{x^2+2C}{2x} \\ \text{BERNOULLI} & & y = \dfrac{x^2+2C}{2x} \\ \text{RICCATI} & & \left[\begin{array}{c} \text{Parameters :} \\ \dfrac{1}{x} \\ 0 \\ 1 \\ y_p \end{array}\right] \end{array}\right]$$

3.11. First order ODE of nth degree

- **Function name**: FirstOrderDegreeN
- **Arguments**: r^n + P1(x,y) r^(n−1) + P2(x,y) r^(n−2) + ··· + Pn(x,y)
- **Optional argument**: true or false to set on or off the stepwise option.
- **Description**: It finds the solutions of the ODE

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

Example: Solve $(y')^2 - (x+y)y' + xy = 0$ step by step.

Solution with Derive

FirstOrderDegreeN(r^2-(x+y)r+xy, true)

When factorizing the differential equation, the following differential equations are obtained:
y' = x
y' = y

The differential equation $y' = x$ belongs to the following types:

SEPARABLE EQUATION EXACT INTEGRATING FACTOR LINEAR BERNOULLI RICCATI

and the solution is given by: $x^2/2 - y = C1$

The differential equation $y' = y$ belongs to the following types:

SEPARABLE EQUATION INTEGRATING FACTOR LINEAR RICCATI

and the solution is given by: $\text{LN}(y) - x = -C2$

Thus, the solution to the differential equation is given by the following general solutions family:

$$\left[\frac{x^2}{2} - y = C1, \text{LN}(y) - x = -C2\right]$$

3.12. Cauchy's problems for first order ODE

- **Function name**: `CauchyProblem`
- **Arguments**: `sol, x0, y0`
- **Description**: It finds the solution of: $\left\{\begin{array}{l} y' = f(x, y) \\ y(x_0) = y_0 \end{array}\right.$ where `sol` is the general solution of $y' = f(x, y)$.

Example: If $x^2 + y^2 = Cx$ is the general solution of $y' = f(x, y)$ find the solution of
$(a)\ \left\{\begin{array}{l} y' = f(x, y) \\ y(1) = 2 \end{array}\right.$ $(b)\ \left\{\begin{array}{l} y' = f(x, y) \\ y(0) = 0 \end{array}\right.$ $(c)\ \left\{\begin{array}{l} y' = f(x, y) \\ y(0) = 1 \end{array}\right.$

Solution with Derive

(a) `CauchyProblem(x^2 + y^2 = Cx, 1, 2)`

$$x^2 + y^2 = 5x$$

(b) `CauchyProblem(x^2 + y^2 = Cx, 0, 0)`

$$x^2 + y^2 = Cx$$

(c) `CauchyProblem(x^2 + y^2 = Cx, 0, 1)`

```
There is no solution for this problem.
```

3.13. Higher order linear homogeneous equations with constant coefficients

- **Function name**: `LinearHomogeneousOrderN`
- **Arguments**: `a0 r^n + a1 r^(n−1) + a2 r^(n−2) + ··· + an`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the Linear equation:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_n y = 0.$$

Example: Solve $y'' - 2y' + y = 0$ step by step.

Solution with Derive

`LinearHomogeneousOrderN(r^2 - 2r + 1, true)`

The characteristic polynomial roots together with their multiplicity grades are:

[[Root, Multiplicity], [1, 2]]

Thus, a solutions fundamental system for the homogeneous equation is given by:

$[e^x, xe^x]$

and the general solution is

$$y = e^x(C2x + C1)$$

3.14. Lagrange's method for particular solutions of higher order linear equations with constant coefficients

- **Function name**: `LagrangeParticularSolution`
- **Arguments**: `a0 r^n + a1 r^(n−1) + a2 r^(n−2) + ···+ an, f`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds a particular solution of the Linear equation:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_n y = f(x).$$

using the Lagrange's method.

Example: Find, step by step, a particular solution of $y'' - 2y' + y = e^x$ using Lagrange's method.

Solution with DERIVE

`LagrangeParticularSolution(`r^2 - 2r + 1, e^x, `true)`

The characteristic polynomial roots together with their multiplicity grades are:

[[Root, Multiplicity], [1, 2]]

Thus, a solutions fundamental system for the homogeneous equation is given by:

$[e^x, xe^x]$

Thus, there exists a particular solution of the form:

$y = e^x(C2(x)x + C1(x))$

Solving the corresponding system to find out such solution, the particular solution is:

$$y = \frac{x^2 e^x}{2}$$

3.15. Higher order linear equations with constant coefficients

- **Function name**: `LinearOrderN`
- **Arguments**: `a0 r^n + a1 r^(n−1) + a2 r^(n−2) + ··· + an, f`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the Linear equation:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_n y = f(x).$$

Example: Solve $y'' - 2y' + y = e^x$ step by step.

Solution with DERIVE

`LinearOrderN(`r^2 - 2r + 1, e^x, `true)`

The characteristic polynomial roots together with their multiplicity grades are:

[[Root, Multiplicity], [1, 2]]

Thus, a solutions fundamental system for the homogeneous equation is given by:

$[e^x, xe^x]$

and the general solution is

$y_h = e^x(C2x + C1)$

The particular solution, using Lagrange's method, is given by:

$y_p = x^2 e^x/2$

Since the general solution of the equation is given by $y = y_h + y_p$, the solution is:

$$y = e^x\left(\frac{x^2}{2} + C2x + C1\right)$$

3.16. Cauchy's problems of higher order linear equations with constant coefficients

- **Function name**: `CauchyProblemOrderN`
- **Arguments**: `a0 r^n + a1 r^(n−1) + a2 r^(n−2) + ··· + an, f, x0, [y0, y1, ..., yn−1]`
- **Optional argument**: `true` or `false` to set on or off the stepwise option.
- **Description**: It finds the solution of the Cauchy's problem:

$$\begin{cases} a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_n y = f(x) \\ y(x_0) = y0 \\ y'(x_0) = y1 \\ \ldots \\ y^{(n-1)}(x_0) = y_{n-1} \end{cases}$$

Example: Solve $\begin{cases} y^{(IV)} - 2y'' + y = e^x \\ y(0) = 1 \\ y'(0) = 2 \\ y''(0) = 3 \\ y'''(0) = 4 \end{cases}$ step by step

Solution with Derive

`CauchyProblemOrderN(`r^4 - $2r^2$ + 1, e^x, 0, [1, 2, 3, 4], `true)`

To solve a Cauchy's problem, the general solution of the differential equation is needed. The characteristic polynomial roots together with their multiplicity grades are:

[[Root, Multiplicity], [1, 2], [−1, 2]]

Thus, a solutions fundamental system for the homogeneous equation is given by:

$[e^x, xe^x, e^{-x}, xe^{-x}]$

and the general solution is

$y_h = e^x(C2x + C1) + e^{-x}(C4x + C3)$

The particular solution, using Lagrange's method, is given by:

$y_p = e^x(2x^2 - 4x + 3)/16$

Since the general solution of the equation is given by $y = y_h + y_p$, the solution is:

$y = e^x(2x^2 + 4x(4C2 - 1) + 16C1 + 3)/16 + e^{-x}(C4x + C3)$

Considering the initial conditions, an equation system to determinate the value of the generic constant is obtained. Solving such system, the solution of the Cauchy's problem comes to be:

$$y = \frac{e^x(2x^2 + 12x + 19)}{16} - e^{-x}\left(\frac{x}{8} + \frac{3}{16}\right)$$

4. Prototype of GUI for SODES

In order to make SODES more portable and usable, we have developed a prototype of this new stepwise solver in Python [14,15], a more portable and free programming language, using its CAS package SymPy [16,17] and Html. Furthermore, we have developed a prototype of GUI to deal with the solver using a web browser locally. This prototype deals with the first order ODE considered in SODES and will be extended in future to deal with higher order linear equations with constant coefficients and Cauchy problems so that it will deal with all types included in SODES.

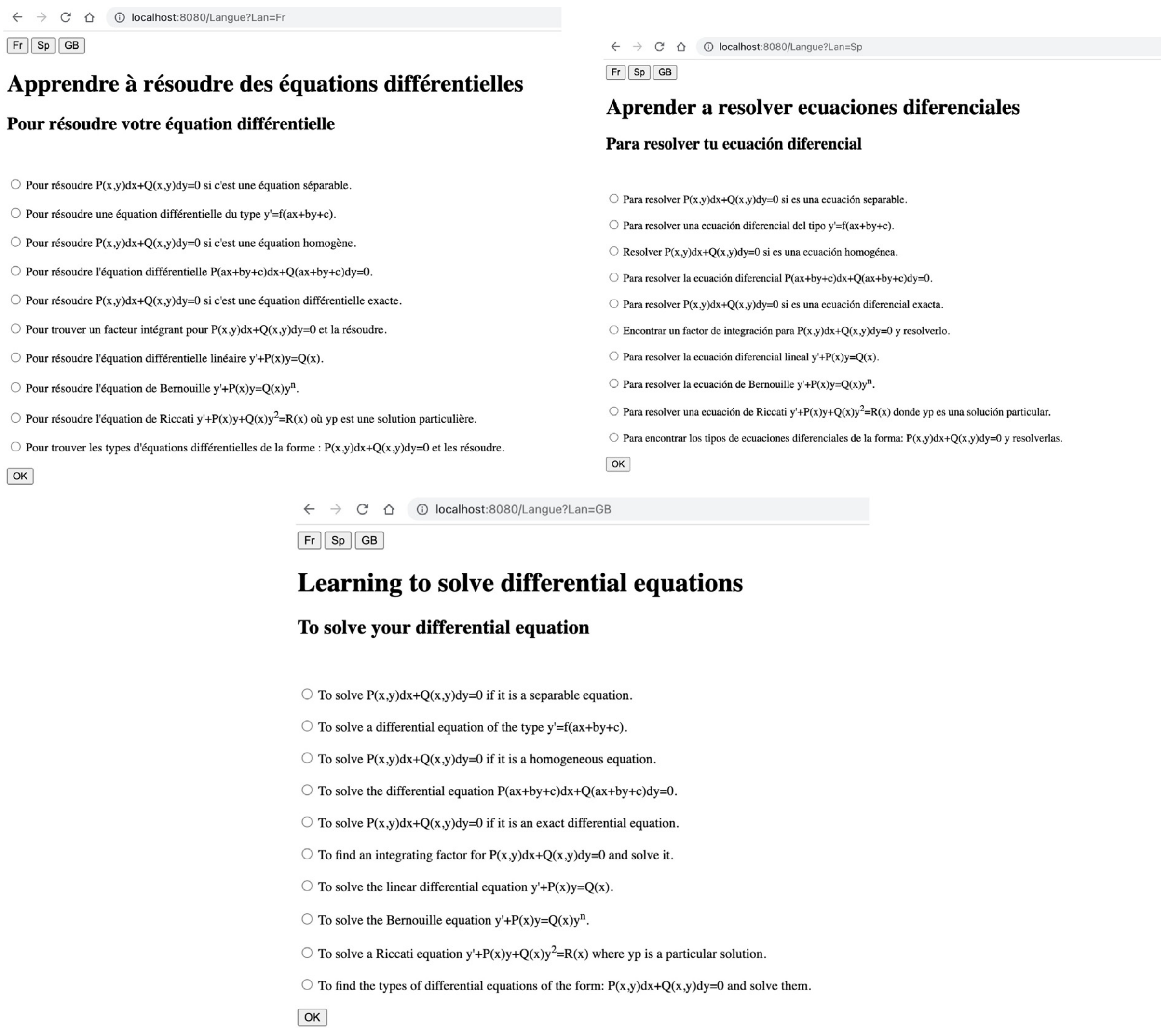
4.1. Languages

This prototype of local GUI for SODES can be used in English, Spanish and French but can be easily extended to other languages (see Fig. 1). Not only the frontend but also the stepwise solutions are provided in the selected language.

4.2. Examples of use

In order to show how to work with the GUI, let us consider the following two examples:

1. Given the differential equation $x(1 + y^2)\,dx + y(1 + x^2)\,dy = 0$, find out the different types of it and get a step by step solution.
   Solution
   We can start by finding the type or types of ODE of the above equation and we obtain the result shown in Fig. 2. Therefore, we have checked that the previous ODE is separable, exact and has an integrating factor (in fact, since the integrating factor is 1, it means that the ODE is already an exact one, as we already also know). Now, we could go back to the main screen and select, for example, the first option which will provide the solution step by step as a separable ODE (see Fig. 3).
2. Find an integrating factor in order to solve the differential equation
   $$\Big(x\sin(y) + y\cos(y)\Big)\,dx + \Big(x\cos(y) - y\sin(y)\Big)\,dy = 0$$
   Solution
   We just have to select the "finding the integrating factor" option in the main screen of the GUI and introduce function P and Q. The result provides the integrating factor, the exact differential equation to solve and the final solution (see Fig. 4).


localhost:8080/Langue?Lan=Fr
Fr
Sp
GB
OK
localhost:8080/Langue?Lan=Sp
localhost:8080/Langue?Lan=GB

Apprendre à résoudre des équations différentielles

Pour résoudre votre équation différentielle

○ Pour résoudre P(x,y)dx+Q(x,y)dy=0 si c'est une équation séparable.
○ Pour résoudre une équation différentielle du type y'=f(ax+by+c).
○ Pour résoudre P(x,y)dx+Q(x,y)dy=0 si c'est une équation homogène.
○ Pour résoudre l'équation différentielle P(ax+by+c)dx+Q(ax+by+c)dy=0.
○ Pour résoudre P(x,y)dx+Q(x,y)dy=0 si c'est une équation différentielle exacte.
○ Pour trouver un facteur intégrant pour P(x,y)dx+Q(x,y)dy=0 et la résoudre.
○ Pour résoudre l'équation différentielle linéaire y'+P(x)y=Q(x).
○ Pour résoudre l'équation de Bernouille y'+P(x)y=Q(x)y^n.
○ Pour résoudre l'équation de Riccati y'+P(x)y+Q(x)y^2=R(x) où yp est une solution particulière.
○ Pour trouver les types d'équations différentielles de la forme : P(x,y)dx+Q(x,y)dy=0 et les résoudre.

Aprender a resolver ecuaciones diferenciales

Para resolver tu ecuación diferencial

○ Para resolver P(x,y)dx+Q(x,y)dy=0 si es una ecuación separable.
○ Para resolver una ecuación diferencial del tipo y'=f(ax+by+c).
○ Resolver P(x,y)dx+Q(x,y)dy=0 si es una ecuación homogénea.
○ Para resolver la ecuación diferencial P(ax+by+c)dx+Q(ax+by+c)dy=0.
○ Para resolver P(x,y)dx+Q(x,y)dy=0 si es una ecuación diferencial exacta.
○ Encontrar un factor de integración para P(x,y)dx+Q(x,y)dy=0 y resolverlo.
○ Para resolver la ecuación diferencial lineal y'+P(x)y=Q(x).
○ Para resolver la ecuación de Bernouille y'+P(x)y=Q(x)y^n.
○ Para resolver una ecuación de Riccati y'+P(x)y+Q(x)y^2=R(x) donde yp es una solución particular.
○ Para encontrar los tipos de ecuaciones diferenciales de la forma: P(x,y)dx+Q(x,y)dy=0 y resolverlas.

Learning to solve differential equations

To solve your differential equation

○ To solve P(x,y)dx+Q(x,y)dy=0 if it is a separable equation.
○ To solve a differential equation of the type y'=f(ax+by+c).
○ To solve P(x,y)dx+Q(x,y)dy=0 if it is a homogeneous equation.
○ To solve the differential equation P(ax+by+c)dx+Q(ax+by+c)dy=0.
○ To solve P(x,y)dx+Q(x,y)dy=0 if it is an exact differential equation.
○ To find an integrating factor for P(x,y)dx+Q(x,y)dy=0 and solve it.
○ To solve the linear differential equation y'+P(x)y=Q(x).
○ To solve the Bernouille equation y'+P(x)y=Q(x)y^n.
○ To solve a Riccati equation y'+P(x)y+Q(x)y^2=R(x) where yp is a particular solution.
○ To find the types of differential equations of the form: P(x,y)dx+Q(x,y)dy=0 and solve them.

Fig. 1. Different languages supported by the local GUI.

4.3. Download and installation information

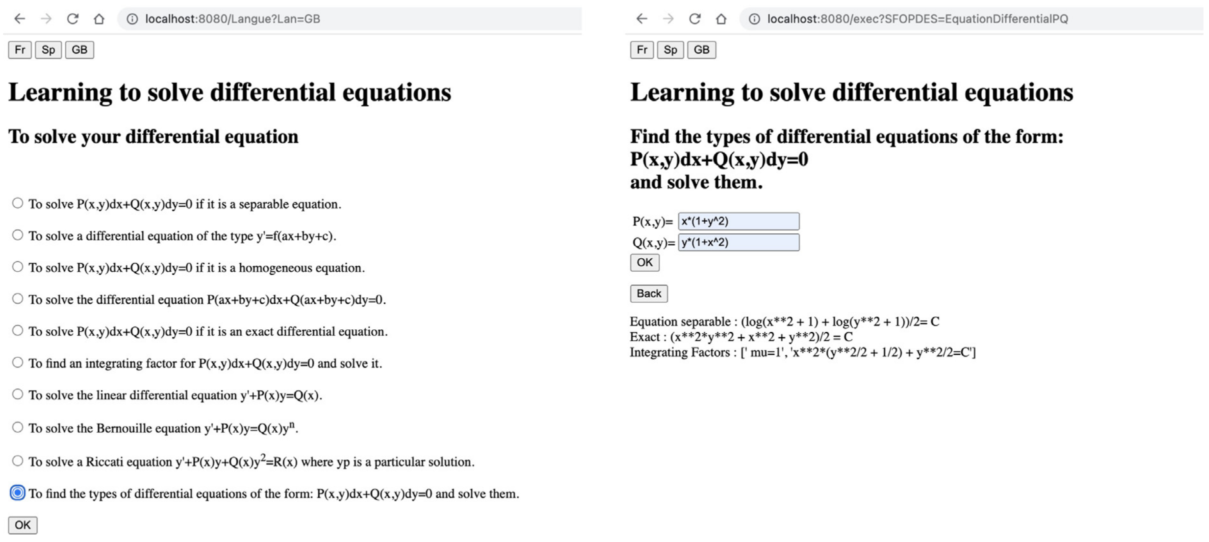
This prototype (Python code and GUI) can be freely downloaded in the folder `LocalGUI` at http://u.uma.es/c4U/SODES/. In the file `install.txt` the instructions to install and execute the GUI in a local web browser are given.

5. Conclusions and future work

5.1. Conclusions

We have introduce the new solver SODES for Ordinary Differential Equations which can optionally provides solutions step by step. SODES has been developed using programming with Derive. The ODE considered in SODES are: Separable, Homogeneous, Exact, Integrating factors, Linear, Bernoulli, Riccati, First order ODE of *n*th degree, Cauchy's problems of first order ODE, Higher order linear homogeneous equations with constant coefficients, Lagrange's method for particular solutions of higher order linear equations with constant coefficients, Higher order linear equations with constant coefficients and Cauchy's problems of higher order linear equations with constant coefficients. SODES also deals with two generic programs which determine the type or types of a given ODE and provides the solution.

The code of all programs in SODES can freely available at http://u.uma.es/c4U/SODES/ where two different files can be downloaded:


localhost:8080/Langue?Lan=GB

Fr Sp GB

Learning to solve differential equations

To solve your differential equation

- ○ To solve P(x,y)dx+Q(x,y)dy=0 if it is a separable equation.
- ○ To solve a differential equation of the type y'=f(ax+by+c).
- ○ To solve P(x,y)dx+Q(x,y)dy=0 if it is a homogeneous equation.
- ○ To solve the differential equation P(ax+by+c)dx+Q(ax+by+c)dy=0.
- ○ To solve P(x,y)dx+Q(x,y)dy=0 if it is an exact differential equation.
- ○ To find an integrating factor for P(x,y)dx+Q(x,y)dy=0 and solve it.
- ○ To solve the linear differential equation y'+P(x)y=Q(x).
- ○ To solve the Bernouille equation y'+P(x)y=Q(x)y^n.
- ○ To solve a Riccati equation y'+P(x)y+Q(x)y^2=R(x) where yp is a particular solution.
- [✓ To find the types of differential equations of the form: P(x,y)dx+Q(x,y)dy=0 and solve them.]

OK

localhost:8080/exec?SFOPDES=EquationDifferentialPQ

Fr Sp GB

Learning to solve differential equations

Find the types of differential equations of the form:
P(x,y)dx+Q(x,y)dy=0
and solve them.

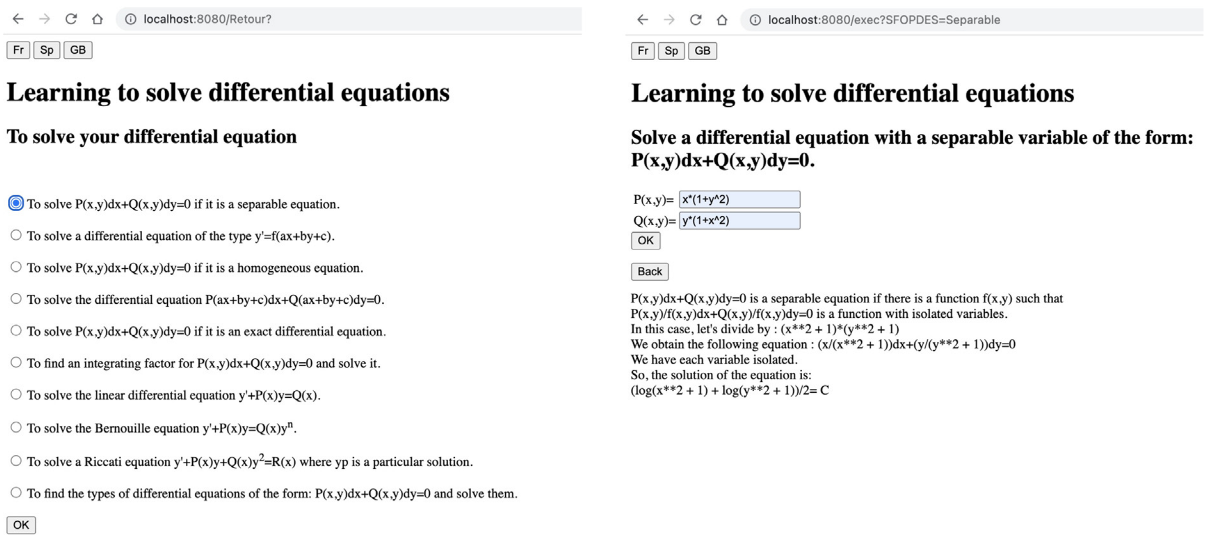
P(x,y)=	x*(1+y^2)
Q(x,y)=	y*(1+x^2)

OK

Back

```
Equation separable : (log(x**2 + 1) + log(y**2 + 1))/2= C
Exact : (x**2*y**2 + x**2 + y**2)/2 = C
Integrating Factors : [' mu=1', 'x**2*(y**2/2 + 1/2) + y**2/2=C']
```

Fig. 2. Finding the types of equation $x(1 + y^2)\,dx + y(1 + x^2)\,dy = 0.$


localhost:8080/Retour?

Fr Sp GB

Learning to solve differential equations

To solve your differential equation

- [✓ To solve P(x,y)dx+Q(x,y)dy=0 if it is a separable equation.]
- ○ To solve a differential equation of the type y'=f(ax+by+c).
- ○ To solve P(x,y)dx+Q(x,y)dy=0 if it is a homogeneous equation.
- ○ To solve the differential equation P(ax+by+c)dx+Q(ax+by+c)dy=0.
- ○ To solve P(x,y)dx+Q(x,y)dy=0 if it is an exact differential equation.
- ○ To find an integrating factor for P(x,y)dx+Q(x,y)dy=0 and solve it.
- ○ To solve the linear differential equation y'+P(x)y=Q(x).
- ○ To solve the Bernouille equation y'+P(x)y=Q(x)y^n.
- ○ To solve a Riccati equation y'+P(x)y+Q(x)y^2=R(x) where yp is a particular solution.
- ○ To find the types of differential equations of the form: P(x,y)dx+Q(x,y)dy=0 and solve them.

OK

localhost:8080/exec?SFOPDES=Separable

Fr Sp GB

Learning to solve differential equations

Solve a differential equation with a separable variable of the form:
P(x,y)dx+Q(x,y)dy=0.

P(x,y)=	x*(1+y^2)
Q(x,y)=	y*(1+x^2)

OK

Back

```
P(x,y)dx+Q(x,y)dy=0 is a separable equation if there is a function f(x,y) such that
P(x,y)/f(x,y)dx+Q(x,y)/f(x,y)dy=0 is a function with isolated variables.
In this case, let's divide by : (x**2 + 1)*(y**2 + 1)
We obtain the following equation : (x/(x**2 + 1))dx+(y/(y**2 + 1))dy=0
We have each variable isolated.
So, the solution of the equation is:
(log(x**2 + 1) + log(y**2 + 1))/2= C
```

Fig. 3. Solving $x(1 + y^2)\,dx + y(1 + x^2)\,dy = 0$ step by step.

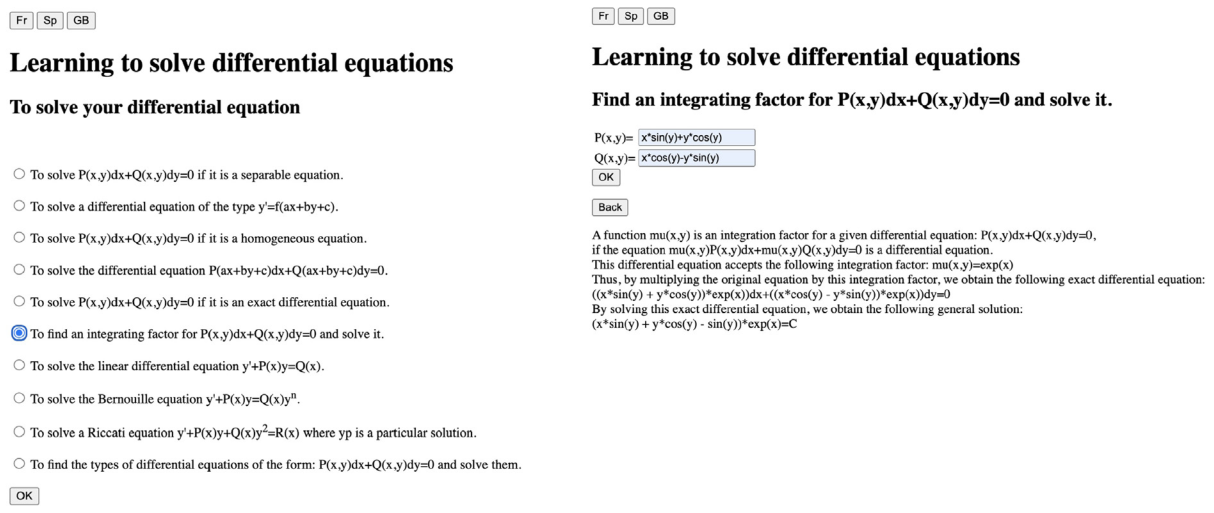
1. `SODES.mth` which is a library with all the programs of SODES. Using this file, we increase the capabilities of the CAS Derive which was one of the main goals of the paper. In addition, since the code is freely available, the user can migrate SODES to any other CAS increasing this way the capabilities of the CAS.
2. `SODES.dfw` which is a tutorial of SODES with examples developed step by step. This can be used in the teaching and learning process of ODE which is a very important topic in advanced subjects aim to Math, Engineering and other Sciences students. Therefore, we achieve the other main goal of the paper: we provide an important tool for education. Note that users can adapt the code in order to get the stepwise solutions according with their expectations.

With the development of SODES we have done one of the future works stated in [29] where we indicated that a stepwise solver for ODE was within the future lines of that previous work.

5.2. Future work

The future work related with this paper can be enumerated in the following lines of continuation:

1. We will consider more types of ODE to include in SODES.



Learning to solve differential equations

To solve your differential equation

- To solve P(x,y)dx+Q(x,y)dy=0 if it is a separable equation.
- To solve a differential equation of the type y'=f(ax+by+c).
- To solve P(x,y)dx+Q(x,y)dy=0 if it is a homogeneous equation.
- To solve the differential equation P(ax+by+c)dx+Q(ax+by+c)dy=0.
- To solve P(x,y)dx+Q(x,y)dy=0 if it is an exact differential equation.
- [✓ To find an integrating factor for P(x,y)dx+Q(x,y)dy=0 and solve it.]
- To solve the linear differential equation y'+P(x)y=Q(x).
- To solve the Bernouille equation y'+P(x)y=Q(x)y^n.
- To solve a Riccati equation y'+P(x)y+Q(x)y^2=R(x) where yp is a particular solution.
- To find the types of differential equations of the form: P(x,y)dx+Q(x,y)dy=0 and solve them.

Learning to solve differential equations

Find an integrating factor for P(x,y)dx+Q(x,y)dy=0 and solve it.

P(x,y)= x*sin(y)+y*cos(y)
Q(x,y)= x*cos(y)-y*sin(y)

A function mu(x,y) is an integration factor for a given differential equation: P(x,y)dx+Q(x,y)dy=0,
if the equation mu(x,y)P(x,y)dx+mu(x,y)Q(x,y)dy=0 is a differential equation.
This differential equation accepts the following integration factor: mu(x,y)=exp(x)
Thus, by multiplying the original equation by this integration factor, we obtain the following exact differential equation:
((x*sin(y) + y*cos(y))*exp(x))dx+((x*cos(y) - y*sin(y))*exp(x))dy=0
By solving this exact differential equation, we obtain the following general solution:
(x*sin(y) + y*cos(y) - sin(y))*exp(x)=C

Fig. 4. Finding an integrating factor to solve $\Big(x\sin(y)+y\cos(y)\Big)\,\mathrm{d}x+\Big(x\cos(y)-y\sin(y)\Big)\,\mathrm{d}y=0$.

2. We will extend the prototype of GUI to these new types of SODES and will add more languages to the local GUI.
3. We will develop other stepwise solvers to deal with other mathematical topics as Complex analysis, Random variable generation and Automatic theorem provers.
4. In the long term, we will develop an online stepwise powerful math calculator considering the previously developed solvers for Partial differential equations (SFOPDES [29]), Multiple integration (SMIS [28]), Improper integrals computations [5,27] together with the present work introduced in this paper: SODES. The calculator will be freely available online and will deal with different languages. It will also be continuously updated with the new solvers we will be developing in future.

Data availability

No data was used for the research described in the article.

Acknowledgment

Funding for open access charge: Universidad de Málaga / CBUA.

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

References

[1] A.D. Rich, A brief history of the muMATH/DERIVE Cass, Deriv. Newsl. 40 (2000) 5.
[2] R. Rein Prank, M. Lepp, Conceptualizing a pedagogical CAS for algebraic manipulation of expression, R & E-Source (2014) Volume Special Issue of TIME 2014.
[3] J. Calmet J.A. van Hulzen, Computer algebra systems, in: B. Buchberger, G.E. Collins, R. Loos (Eds.), Computer Algebra. Symbolic and Algebraic Manipulation, Springer, Vienna, 1982, pp. 221–243.
[4] M.J. Wester, Computer Algebra Systems: A Practical Guide, Wiley, Chichester, 1999.
[5] J.L. Galán-García, G. Aguilera-Venegas, M.A. 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, http://dx.doi.org/10.1016/j.amc.2016.12.024.
[6] Derive User Group, myehosthttp://www.austromath.at/dug/. (Accessed on 29 October 2022).
[7] http://www.austromath.at/dug/. (Accessed on 29 October 2022),
[8] B.W. Char, et al., A tutorial introduction to maple, J. Symbolic Comput. 2 (2) (1986) 179–200, http://dx.doi.org/10.1016/S0747-7171(86)80021-9.
[9] Maxima, https://maxima.sourceforge.io/. (Accessed on 29 October 2022).
[10] J. Abbott, A. Bigatti, M. Caboara, L. Robbiano, CoCoA: Computations in commutative algebra, ACM Commun. Comput. Algebra 41 (3) (2007) 111–112, http://dx.doi.org/10.1145/1358190.1358204.
[11] CoCoA, https://cocoa.dima.unige.it/cocoa/. (Accessed on 29 October 2022).
[12] Java, https://www.java.com/. (Accessed on 29 October 2022).
[13] Php, https://www.php.net/. (Accessed on 29 October 2022).
[14] M.F. Sanner, A programming language for software integration and development, J. Mol. Graph. 17 (1999) 57–61, http://dx.doi.org/10.1016/S1093-3263(99)99999-0.
[15] Python, https://www.python.org/. (Accessed on 29 October 2022).
[16] D. Joyner, O. Čertík, A. Meurer, B.E. Granger, Open source computer algebra systems: SymPy, ACM Commun. Comput. Algebra 45 (3/4) (2012) 225–234, http://dx.doi.org/10.1145/2110170.2110185.

[17] SymPy, http://www.sympy.org/. (Accessed on 29 October 2022).
[18] 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, Math. Comput. Simulation 80 (6) (2010) 1200–1211, http://dx.doi.org/10.1016/j.matcom.2009.04.012.
[19] 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, Math. Comput. Simulation 104 (2014) 21–30, http://dx.doi.org/10.1016/j.matcom.2012.03.010.
[20] J.L. Galán-García, G. Aguilera-Venegas, P. Rodríguez-Cielos, An accelerated-time simulation for traffic flow in a smart city, J. Comput. Appl. Math. 270 (2014) 557–563, http://dx.doi.org/10.1016/j.cam.2013.11.020.
[21] J.L. Galán-García, G. Aguilera-Venegas, M.Á. Galán-García, P. Rodríguez-Cielos, A new probabilistic extension of Dijkstra's algorithm to simulate more realistic traffic flow in a smart city, Appl. Math. Comput. 267 (2015) 780–789, http://dx.doi.org/10.1016/j.amc.2014.11.076.
[22] G. Aguilera-Venegas, J.L. Galán-García, E. Mérida-Casermeiro, P. Rodríguez-Cielos, An accelerated-time simulation of baggage traffic in an airport terminal, Math. Comput. Simulation 104 (2014) 58–66, http://dx.doi.org/10.1016/j.matcom.2013.12.009.
[23] E. Roanes-Lozano, J.L. Galán-García, G. Aguilera-Venegas, A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. 267 (2015) 758–770, http://dx.doi.org/10.1016/j.amc.2014.12.001.
[24] G. Aguilera-Venegas, J.L. Galán-García, R. Egea-Guerrero, M.Á. Galán-García, P. Rodríguez-Cielos, Y. Padilla-Domínguez, M. Galán-Luque, A probabilistic extension to Conway's game of life, Adv. Comput. Math. 45 (2019) 2111–2121, http://dx.doi.org/10.1016/j.matcom.2013.12.009.
[25] 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, Int. J. Technol. Math. Educ. 22 (2) (2015) 85–92.
[26] R.M. Smullyan, First-Order Logic, Springer, Berlin, 1968.
[27] J.L. Galán-García, G. Aguilera-Venegas, M.Á. Galán-García, P. Rodríguez-Cielos, I. Atencia-Mc.Killop, Y. Padilla-Domínguez, R. Rodríguez-Cielos, Enhancing CAS improper integrals computations using extensions of the residue theorem, Adv. Comput. Math. 45 (2019) 1825–1841, http://dx.doi.org/10.1007/s10444-018-09660-y.
[28] J.L. Galán-García, P. Rodríguez-Cielos, Y. Padilla-Domínguez, M.Á. Galán-García, I. Atencia, P. Rodríguez-Padilla, G. Aguilera-Venegas, SMIS: A stepwise multiple integration solver using a CAS, Mathematics 9 (22) (2021) 2866, http://dx.doi.org/10.3390/math9222866.
[29] J.L. Galán-García, G. Aguilera-Venegas, P. Rodríguez-Cielos, Y. Padilla-Domínguez, M.Á. Galán-García, SFOPDES: A stepwise first order partial differential equations solver with a computer algebra system, Comput. Math. Appl. 78 (2019) 3152–3164, http://dx.doi.org/10.1016/j.camwa.2019.05.010.
[30] WolframAlpha Computational Intelligence, Available online: https://www.wolframalpha.com/. (Accessed on 29 October 2022).
[31] Wolfram Language website, Available online: https://www.wolfram.com/language/. (Accessed on 29 October 2022).
[32] S. Wolfram, Computer algebra: A 32-year update, in: Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation, Boston, MA, USA, 26–29, 2013, pp. 7–8.
[33] MapleCloud, Available online: https://maple.cloud/app/5691363796451328. (Accessed on 29 October 2022).
[34] Symbolab, Available online: https://www.symbolab.com/. (Accessed on 29 October 2022).
[35] MathDF, Available online: https://mathdf.com/dif/. (Accessed on 29 October 2022).
[36] Mathforyou, Available online: https://mathforyou.net/en/online/calculus/ode/. (Accessed on 29 October 2022).
[37] Mathforyou, ODE module, 2022, Available online: https://mathforyou.net/en/online/calculus/ode/. (Accessed on 29 October 2022).
[38] ΣMh (eMathHelp), Available online: https://www.emathhelp.net/calculators/differential-equations/. (Accessed on 29 October 2022).
[39] Math24.pro, Available online: https://math24.pro/differential_equation. (Accessed on 29 October 2022).
[40] snapXam, Available online: https://www.snapxam.com/calculators/first-order-differential-equations-calculator. (Accessed on 29 October 2022).
[41] Maple ODE solver, Available online: https://www.maplesoft.com/applications/detail.aspx?id=154102. (Accessed on 29 October 2022).
[42] Maple online solver, Available online: https://maple.cloud/app/5691363796451328. (Accessed on 29 October 2022).
[43] W. Stein, Sage: Creating a Viable Free Open Source Alternative To Magma, Maple, Mathematica, and MATLAB, in: Lond. Math. Soc. Lect. Note Ser., vol. 403, 2013, pp. 230–238.
[44] Maxima, ODE module, 2022, Available online: https://maxima.sourceforge.io/docs/tutorial/en/gaertner-tutorial-revision/Pages/ODE0001.htm. (Accessed on 29 October 2022).
[45] SymPy, ODE module, 2022, Available online: https://docs.sympy.org/latest/modules/solvers/ode.html. (Accessed on 29 October 2022).
[46] SageMath, ODE module, 2022, Available online: https://doc.sagemath.org/html/en/reference/calculus/sage/calculus/desolvers.html. (Accessed on 29 October 2022).