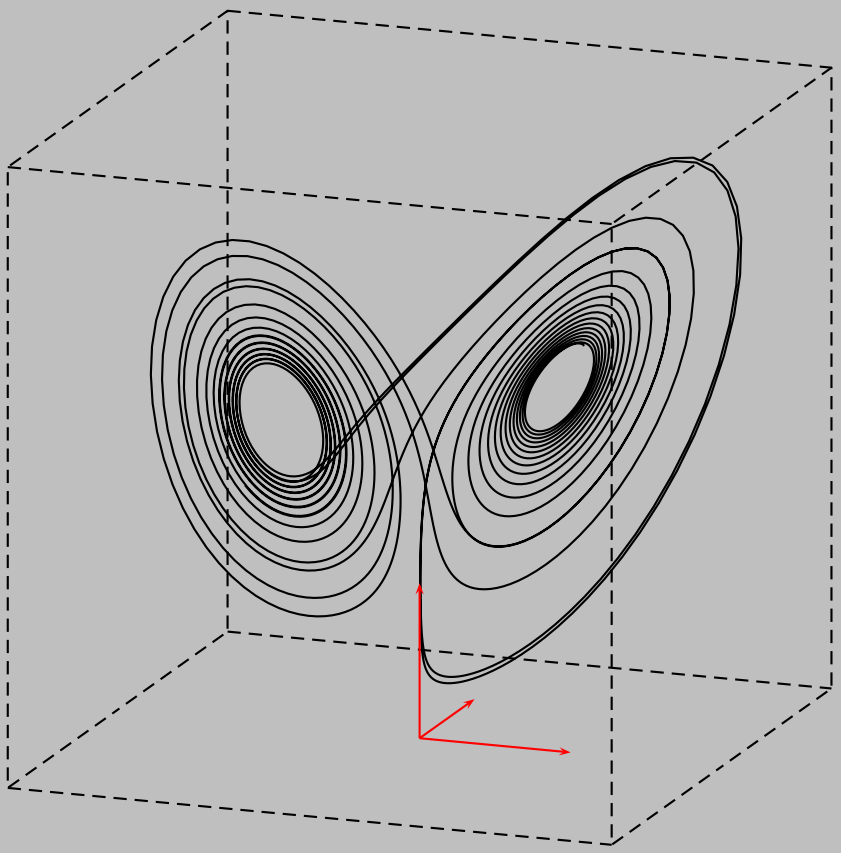
PSTricks

pst-ode

A PSTricks package for solving initial value problems for sets of Ordinary Differential Equations (ODE), v0.11

https://github.com/agrahn/pst-ode

15th August 2017



Package author(s):
Alexander Grahn

The `pstricks-add` package already provides `\psplotDiffEqn` for solving ODEs. However, as its name suggests, the macro always produces a plot of the computed result. While any number of coupled differential equations can be integrated simultaneously, only two-dimensional plots are supported. The user has to select the two components of the computed state vectors to be used in the plot.

Package `pst-ode` separates solving the equations from plotting the result. The result is stored as a `PostScript` object and can be plotted later using macros from other PSTricks packages, such as `\listplot` (`pst-plot`) and `\listplotThreeD` (`pst-3dplot`), or be further processed by user-defined `PostScript` procedures. Optionally, the computed state vectors can be written as a table to a text file.

Package `pst-ode` uses the Runge-Kutta-Fehlberg (RKF45) method with automatic step size control for integrating the differential equations. Thus, the precision of the result does not depend on the number of plot points specified, as it would be the case with the classical Runge-Kutta (RK4) method.

Contents

1 Introduction

An initial value problem involves finding the solution $\mathbf{x}(t)$ of a set of first order differential equations

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{x}) \tag{1}$$

by integrating them with respect to the independent variable t starting at t_0 up to t_e. If the set consists of n differential equations, a vector of initial conditions

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{2}$$

of the same length n is required. For $n = 1$ the initial value problem is one-dimensional:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x) \quad \text{for } t \in [t_0, t_\mathrm{e}], \text{ where} \tag{3}$$

$$x(t_0) = x_0. \tag{4}$$

Instead of producing analytical expressions of the solution functions $\mathbf{x}(t)$, the numerical method gives only approximate values $\mathbf{x}_i$ at N discrete points t_i of the integration interval $I = [t_0, t_\mathrm{e}]$:

$$\mathbf{x}_i \approx \mathbf{x}(t_i). \tag{5}$$

The computed approximations $\mathbf{x}_i$ of the solution as well as the initial condition vector $\mathbf{x}_0$ are called 'state vectors'. In the case of a single equation problem, Eqs. (3), (4), the state vectors have only one component.

2 Commands

`\pstODEsolve [Options]` {*result*}{*output format*}{t_0}{t_e}{N}{$\mathbf{x}_0$}{$\mathbf{f}(t, \mathbf{x})$}

is the main user command for solving initial value problems.

The first mandatory argument *result* is a simple identifier composed of letters and possibly numbers. It is the name of a `PostScript` object that takes the computed state vectors $\mathbf{x}_i$, formatted according to the second argument *output format*, as a long list of values. *result* can be directly used as the *data* argument of `\listplot{`*data*`}` (package `pst-plot`) or `\listplotThreeD{`*data*`}` (package `pst-3dplot`). When put on the `PostScript` operand stack, *result* is immediately executed, that is, the list of values contained in *result* is pushed onto the operand stack. The scope of *result* is global and thus its content survives page breaks.

The second argument *output format* determines which of the components of the state vectors $\mathbf{x}_i$ and possibly the independent variable t are stored into *result*. *output format* can be specified in two different formats, depending on the setting of the command option `algebraicOutputFormat`. If `algebraicOutputFormat` is set, calculations can be made on the components of the computed state vectors before writing them to *result*. Without option `algebraicOutputFormat` the following applies: The keyword `(t)` (parentheses required) inserts the integration parameter value t_i into the result list; numbers (`0`, `1`, `2`, …, $n-1$) in arbitrary order specify the components of vector $\mathbf{x}_i$ to be inserted, as well as their order of insertion. The elements of *output format* are to be separated by spaces. If option `algebraicOutputFormat` is set, *output format* is a `|`-separated list of algebraic expressions (infix notation) according to which the components of the output vector are to be calculated. In these algebraic expressions, the n current state vector components can be referred to as `x[0]`, `x[1]`, …, `x[`$n-1$`]` or `y[0]`, `y[1]`, …, `y[`$n-1$`]`, and the current independent variable value as '`t`'. In either case, there is no upper limit of the output vector length. It must have at least one element though.

Arguments t_0 and t_e define the interval of integration $I = [t_0, t_e]$. Both arguments accept expressions in infix or `PostScript` (postfix, reverse polish) notation. Infix notation requires option `algebraicT`.

N is the number of *equally* spaced output points, including t_0 and t_e; it must be ≥ 2. In order to divide the interval of integration into K output steps, N must be set to $K + 1$. Note that the precision of the solution does *not* depend on N; internal integration steps are automatically inserted and resized according to the changes in the solution.

$\mathbf{x}_0$ is a list of n space separated initial values, one for each differential equation. Alternatively, $\mathbf{x}_0$ can be given as a `PostScript` procedure pushing the initial values on the stack, or as an algebraic expression in infix notation where the elements are separated by '`|`'. Infix notation requires option `algebraicIC`. This argument can be left empty. In that case, the last computed state vector of a preceding `\pstODEsolve` call. Of course, the number of equations n must be the same as in the preceding calculation.

$\mathbf{f}(t, \mathbf{x})$ is the right-hand side of the differential equations. Equations can be entered in either infix or `PostScript` (postfix, reverse polish) notation. Infix notation requires option `algebraic`, and equations have to be separated by '`|`'. The n current state vector components can be referred to as `x[0]`, `x[1]`, …, `x[`$n-1$`]` or `y[0]`, `y[1]`, …, `y[`$n-1$`]`, and the current independent variable value as '`t`'. If given in `PostScript` notation, the provided procedure must first pop the current state vector components in reverse order(!) from the operand stack and then push the first derivatives in regular order back to it. Again, the independent variable value can be accessed using '`t`'.

`\pstODEsolve` accepts a few `[Options]` :

`dt0=<number>`
By default, the output step $(t_e - t_0)/(N-1)$ is used as the intitial, tentative integration step size. A large initial step may cause the integration to crash early by numerical overflow, in particular if the right-hand side evaluates to large values. With option `dt0`, an arbitrary value of the initial step size can be specified.

`append`
If the initial condition argument $\mathbf{x}_0$ of `\pstODEsolve` is left empty, integration of an ODE continues from an already existing state, which was obtained by the most recent use of `\pstODEsolve` or which was re-instated from a previously saved state (`\pstODEsaveState`) by calling `\pstODErestoreState`. Keyword `append` tells `\pstODEsolve` to append the computed result to *result* which must already exist.

`saveData`
If option `saveData` is set, the formatted state vectors are written as a table to a textfile named '*result*`.dat`'. Note that `ps2pdf` must be called with option `-dNOSAFER` to enable writing of external files.

`algebraicOutputFormat`
With `algebraicOutputFormat`, the command argument *output format* is a `|`-separated list of algebraic expressions in infix notation, according to which the output vector components are to be assembled before storing them into *result*. Default is to *not* use algebraic infix expressions. For details, see the description of *output format* above.

`algebraicT`
With `algebraicT`, the integration interval limits t_0 and t_e can be entered as algebraic expressions in infix notation, otherwise `PostScript` (postfix, reverse polish) notation must be used. Of course, single rational numbers for t_0 and t_e always work.

`algebraicIC`
With `algebraicIC`, the initial condition vector $\mathbf{x}_0$ can be given in algebraic infix notation. Vector components have to be separated by '`|`'. Default is `PostScript` notation, i. e. space separated postfix expressions or rational numbers.

`algebraic`
With `algebraic`, the right-hand side of differential equations $\mathbf{f}(t, \mathbf{x})$ can be given in infix notation. Algebraic infix expressions are to be separated by '`|`'. Default is `PostScript` notation.

`algebraicAll`
Option `algebraicAll` is equivalent to setting all of `algebraicOutputFormat`, `algebraicT`, `algebraicIC`, `algebraic`.

`silent`
Option `silent` suppresses the terminal output of stepping information.

`varsteptol=<number>`
The tolerance for automatic calculation of the integration step size can be set with `varsteptol`. The default value is `1e-6`. Sometimes, it may be helpful to relax this value in order to cope with situations where integration stops with step size underflow. The occurence of step size underflow is indicated by writing '`!`' to the terminal and integration stops prematurely before reaching t_e. The current value of t and the current state vector $\mathbf{x}$ are

written to the terminal. Step size underflow may happen for stiff ODEs at some point along the integration interval $[t_0, t_e]$.

`\pstODEsaveState{`*state*`}`

is a user command to save the state of an integration at the end of a `\pstODEsolve` call. The saved state can be restored later, using `\pstODErestoreState{`*state*`}`, in order to continue the integration of an ODE. *state* is an identifier composed of letters and possibly numbers.

`\pstODErestoreState{`*state*`}`

is a user command to restore a previously saved state (`\pstODEsaveState{`*state*`}`). After restoring a state, `\pstODEsolve` can be called with an empty initial condition argument in order to continue integration of the ODE. Of course, the number of differential equations of the current and of the saved `\pstODEsolve` calls must be the same.

3 Examples

3.1 Lorenz Attractor (lorenz.tex)

The Lorenz Attractor depicted on the title page is governed by

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \alpha(y - x)\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\beta - z) - y\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \gamma z.\end{aligned}$$

This system of differential equations is known to display chaotic behaviour due to the non-linear combination (products) of the dependent functions $x(t)$, $y(t)$ and $z(t)$. The trajectory of solution is susceptible to slight changes in the initial conditions and hence to slight discrepancies in the computed intermediate state vectors, which in turn can be regarded as initial conditions for the continuation of the solution. This is known as the 'butterfly effect', a term coined by Lorenz. Although an adaptive stepping algorithm is used, the solution of this initial problem *does* therefore depend on the number of output points chosen. To some extent, this fact contrasts with the statement made in the abstract of this documentation. However, for linear problems which only know one distinct solution it still holds. In the present case, the values $\alpha = 10$, $\beta = 28$, $\gamma = 8/3$ and the initial condition $\mathbf{x}_0 = (10, 10, 30)$ where chosen. The integration parameter t is running from 0 to 25 and the state vector is output at 2501 points of the integration interval.

```
\pstVerb{
  /alpha 10 def
  /beta 28 def
  /gamma 8 3 div def
}
\pstODEsolve[algebraic]{lorenzXYZ}{0 1 2}{0}{25}{2501}{10 10 30}{
  alpha*(x[1]-x[0]) |
  x[0]*(beta-x[2]) - x[1] |
  x[0]*x[1] - gamma*x[2]
```

```
}
\listplotThreeD{lorenzXYZ}
```

As the plot is three-dimensional, all three components of the state vectors are stored in the `PostScript` variable `lorenzXYZ` by setting the *output format* argument to '`0 1 2`'.

3.2 Charged particle movement in a vertical electrical field (particle.tex)

The trajectory $\mathbf{x}(t)$ of the particle shown below is governed by a set of three second order differential equations:

$$\ddot{x} = \omega\dot{y} - \frac{\dot{x}}{\tau} \tag{6a}$$

$$\ddot{y} = -\omega\dot{x} - \frac{\dot{y}}{\tau} \tag{6b}$$

$$\ddot{z} = -\frac{\dot{z}}{\tau}, \tag{6c}$$

where ω and τ are constants. An initial value problem of this type needs $3 \times 2 = 6$ initial conditions. These are given as the initial position $\mathbf{x}_0 = (x_0, y_0, z_0)$ and the initial velocity $\dot{\mathbf{x}}_0 = (\dot{x}_0, \dot{y}_0, \dot{z}_0) = (u_0, v_0, w_0) = \mathbf{v}_0$ of the particle.

In order to solve the equations above numerically, they have to be rewritten as a set of six first order differential equations:

$$\dot{x} = u \tag{7a}$$

$$\dot{y} = v \tag{7b}$$

$$\dot{z} = w \tag{7c}$$

$$\dot{u} = \omega v - \frac{u}{\tau} \tag{7d}$$

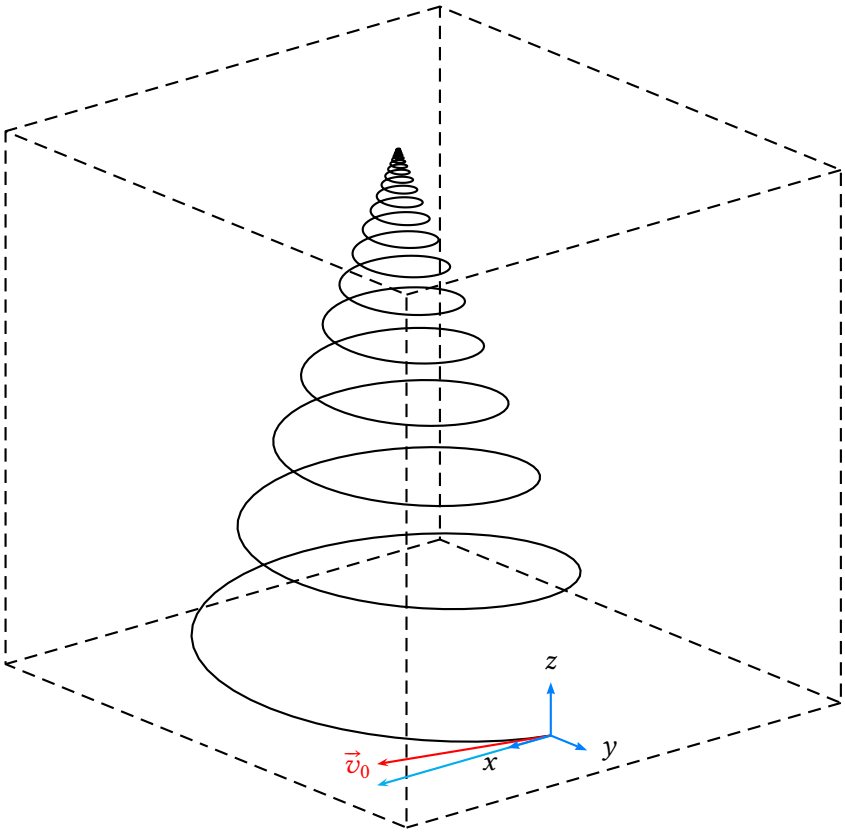
$$\dot{v} = -\omega u - \frac{v}{\tau} \tag{7e}$$

$$\dot{w} = -\frac{w}{\tau}. \tag{7f}$$

Here, ω and τ are both set to the value of 5, the initial position of the particle is defined as $\mathbf{x}_0 = (0, 0, 0)$ and its initial velocity vector as $\mathbf{v}_0 = (20, 0, 2)$. The integration parameter t is running from 0 to 25 and the state vector is output at 1000 points of the integration interval.

```
\pstVerb{
  /wc 5 def
  /tau 5 def
}
\pstODEsolve[algebraic]{particleXYZ}{0 1 2}{0}{25}{1000}{0 0 0 20 0 2}{
  x[3] | x[4] | x[5] | wc*x[4] - x[3]/tau | -wc*x[3] - x[4]/tau | -x[5]/tau
}
\listplotThreeD{particleXYZ}
```

Since we are interested in plotting the particle positions, only the first three components of the state vectors are stored in `particleXYZ`.



3.3 One more first order differential equation (ode.tex)

The aim of the last example is to demonstrate that precision does not depend on the number of output points. We set only five output points and plot the analytical solution against the numerical one for comparison.

The ODE to be solved reads

$$y' = -2yt. \tag{8}$$

It should be noted that the right-hand side also depends the integration parameter t. With the initial condition

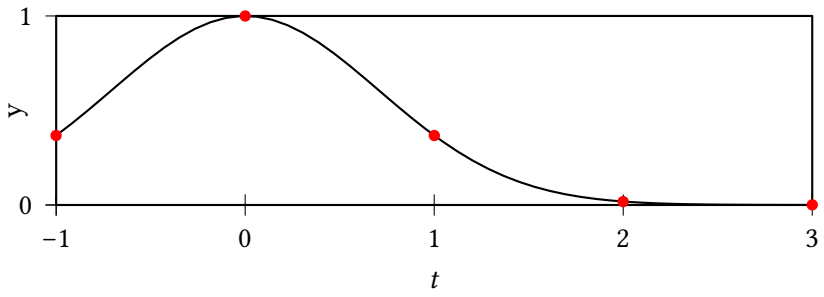
$$y(-1) = 1/e \tag{9}$$

the analytical solution of Eq. (8) is

$$y(t) = e^{-t^2}. \tag{10}$$

```
\pstODEsolve[algebraicIC,algebraic]{TY}{(t) 0}{-1}{3}{5}{1/Euler}{-2*t*y[0]}
\listplot[plotstyle=dots]{TY}
```

Note that in `\pstODEsolve`, '`x[..]`' and '`y[..]`' can be used interchangeably for representing the state vector in algebraic notation. The initial condition (9) is given as an algebraic expression, which requires option `algebraicIC`; in `PostScript` notation it would read '`1 Euler div`'. The integration parameter and the one available state vector component are stored into the `PostScript` object '`TY`' by setting *output format* to '`(t) 0`'.



The plot contains the analytical solution and the five output points of the numerical solution as red dots. They lie exactly on the analytic solution.

4 Acknowledgements

I'd like to thank Manuel Luque for the inspiring examples on his site http://pstricks.blogspot.fr, some of which I used as a basis for this documentation.