Differential Equations in Maple 16

Summary

Maple 16 continues to push the frontiers in differential equation solving and extends its lead in computing closed-form solutions to differential equations, adding in even more classes of problems that can be handled. The numeric ODE, DAE, and PDE solvers also continue to evolve. Maple 16 shows significant performance improvements for these solvers, as well as enhanced event handling.

- Maple 16 integrates new solving methods for 1st, 2nd, and higher order nonlinear ODEs. The new methods can solve additional 1st order Abel and other families of equations, and a number of 2nd and higher order families of equations not admitting point symmetries.
- For both ordinary and partial differential equations, all symmetry algorithms have been extended to automatically handle problems involving anti-commutative variables, making all this DE functionality easily available for problems that involve non-commutative variables, as occur frequently, for example, in physics.
- Event handling for the numeric ODE and DAE solvers has been significantly enhanced to avoid triggering on spurious events as well as to increase performance.
- The numeric PDE solvers are now able to take advantage of the compiler, yielding a tremendous performance boost and allowing you to handle larger problems.

Ordinary Differential Equations (ODEs)

Using new algorithms, the dsolve command can solve new nonlinear ODE families of 1st, 2nd, 3rd and 4th order, all of them parameterized by arbitrary functions of the independent variable.

New solvable 1st order nonlinear ODE families

For 1st order ODEs, the simplest problems known to be beyond the reach of

complete solving algorithms are known as Abel equations. These are equations of the form

$$y' = \frac{f_3 y^3 + f_2 y^2 + f_1 y + f_0}{g_1 y + g_0}$$

where $y \equiv y(x)$ is the unknown and the $f_i \equiv f_i(x)$ and $g_j \equiv g_j(x)$ are arbitrary functions of x. The biggest subclass of Abel equations known to be solvable, the AIR 4-parameter class, was discovered by our research team. New in Maple 16, two additional 1-parameter classes of Abel equations, beyond the AIR class, are now also solvable.

Examples

> $PDEtools[declare](y(x), prime = x)$

y(x) will now be displayed as y

derivatives with respect to x of functions of one variable will now be displayed with '

This equation, of type Abel 2nd kind, depending on one parameter alpha, is now solved in terms of hypergeometric functions

> $ode_1 := \frac{d}{dx} y(x) = -(y(x)\,(-3+x)\,((-9+2\alpha)\,y(x)-9)) \Big/ (x\,(54-36x+6x^2-3\alpha+2\alpha x)\,y(x)+9\alpha+81-54x+9x^2)$

$$ode_1 := y' = -\frac{y\,(-3+x)\,((-9+2\alpha)\,y-9)}{x\,(54-36x+6x^2-3\alpha+2\alpha x)\,y+9\alpha+81-54x+9x^2}$$

> $dsolve(ode_1)$

$$_C1 + \frac{1}{128}\left(128\left(\alpha - \frac{9}{2}\right)^5 y^4\left(-y + \frac{2}{9}y\alpha - 1\right)^{\frac{-9+8\alpha}{-9+2\alpha}}(xy + 3)^2\,\text{hypergeom}\left(\left[5, \frac{-9+8\alpha}{-9+2\alpha}\right], \left[\frac{-18+10\alpha}{-9+2\alpha}\right], y - \frac{2}{9}y\alpha + 1\right) - 69984\left(-\frac{9}{2} + \left(\alpha - \frac{9}{2}\right)y\right)\left(\alpha - \frac{9}{8}\right)\left(\frac{3}{2} + \left(-\frac{3}{2} + x\right)y\right)\left(-y + \frac{2}{9}y\alpha - 1\right)^{\frac{6\alpha}{-9+2\alpha}}\right) \Big/ \left(\left(\alpha - \frac{9}{2}\right)^4 y^4\left(\alpha - \frac{9}{8}\right)(xy + 3)^2\right) = 0$$

The related *class* of Abel equations that is now *entirely* solvable consists of the set of equations that can be obtained from equation **(2)** by changing variables

$$x \to F(x), y \to \frac{P_1(x)\,y + Q_1(x)}{P_2(x)\,y + Q_2(x)}$$

where $G(x)$ and the four $P_i(x), Q_j(x)$ are arbitrary rational functions of x; this is the most general transformation that preserves the form of Abel equations and thus generates Abel ODE classes.

The following ODE family, which depends on an arbitrary function $G(x)$, is representative of the next difficult problem beyond Abel equations, that is, a problem involving 4th powers of $y(x)$ in the right-hand side.

> $ode_2 := \frac{d}{dx}y(x) = \frac{1\left(G(x)^2 y(x)^4 - \left(\frac{d}{dx}G(x)\right)y(x)^2 + x\right)}{2\,G(x)\,y(x)}$

$$ode_2 := y' = \frac{1}{2}\,\frac{G(x)^2 y^4 - G' y^2 + x}{G(x)\,y}$$

We solve it here in implicit form to avoid square roots obscuring the solution

> $sol_2 := dsolve(ode_2, y(x), implicit)$

$$sol_2 := \_C1 + \frac{-G(x)\,y^2\,\text{AiryBi}(-x) + \text{AiryBi}(1, -x)}{-G(x)\,y^2\,\text{AiryAi}(-x) + \text{AiryAi}(1, -x)} = 0$$

> $odetest(sol_2, ode_2, y(x))$

$$0$$

A generalization of the problem above, solvable in Maple 16, involving an arbitrary function $G(y(x))$

> $ode_3 := \frac{d}{dx} y(x) = \frac{2 G(y(x)) x}{2 - D(G)(y(x)) x^2 + G(y(x))^2 x^4 - 2 y(x) G(y(x)) x^2}$

$$ode_3 := y' = \frac{2 G(y) x}{2 - D(G)(y) x^2 + G(y)^2 x^4 - 2 y G(y) x^2}$$

> $sol_3 := dsolve(ode_3)$

$$sol_3 := \_C1 + \frac{-2y + G(y) x^2}{2 I e^{y^2} + (-2y + G(y) x^2) \sqrt{\pi}\ \mathrm{erf}(I y)} = 0$$

> $odetest(sol_3, ode_3)$

$$0$$

New solvable nonlinear ODE families of 2nd, 3rd, and 4th order

Using new algorithms developed by our research team, the dsolve command in Maple 16 can additionally solve two new nonlinear ODE families for each of the 2nd, 3rd and 4th order problems, with the ODE families involving arbitrary functions of the independent (x) or dependent ($y(x)$) variables.

Examples

A 4th order ODE family

> $ode_4 := \frac{d^4}{dx^4} y(x) = 5 y(x) \left(\frac{d^3}{dx^3} y(x)\right) + \left(10 \left(\frac{d}{dx} y(x)\right) - 10 y(x)^2 - \frac{1}{x}\right) \left(\frac{d^2}{dx^2} y(x)\right) - 15 y(x) \left(\frac{d}{dx} y(x)\right)^2 + \left(10 y(x)^3 + \frac{3 y(x)}{x}\right) \left(\frac{d}{dx} y(x)\right) - y(x)^5 - \frac{1 y(x)^3}{x}$

$$ode_4 := y'''' = 5 y y''' + \left(10 y' - 10 y^2 - \frac{1}{x}\right) y'' - 15 y y'^2 + \left(10 y^3 + \frac{3 y}{x}\right) y' - y^5 - \frac{y^3}{x}$$

This ODE has no point symmetries; the determining PDE for the symmetry infinitesimals only admits both of them equal to zero:

> $PDEtools:\text{-}DeterminingPDE(ode_4)$

$$\{\_\xi_x(x,y)=0,\_\eta_y(x,y)=0\}$$

Using new algorithms, this problem is nevertheless solvable in explicit form in terms of Bessel functions

> $dsolve(ode_4)$

$$y=\left(-\_C2-2\_C3\,x-\_C4\,x^{3/2}\,\mathrm{BesselJ}\left(3,2\sqrt{x}\right)-x^{3/2}\,\mathrm{BesselY}\left(3,2\sqrt{x}\right)\right)\Big/\left(\_C1+\_C2\,x+\_C3\,x^2+\_C4\,x^2\,\mathrm{BesselJ}\left(4,2\sqrt{x}\right)+x^2\,\mathrm{BesselY}\left(4,2\sqrt{x}\right)\right)$$

Another problem not admitting point symmetries, of 3rd order

> $$ode_5:=\frac{d^3}{dx^3}y(x)=\left(4y(x)-\frac{1}{x}\right)\left(\frac{d^2}{dx^2}y(x)\right)+3\left(\frac{d}{dx}y(x)\right)^2+\left(-6y(x)^2+\frac{3y(x)}{x}-x\right)\left(\frac{d}{dx}y(x)\right)+y(x)^4-\frac{1y(x)^3}{x}+y(x)^2x-y(x)$$

$$ode_5:=y'''=\left(4y-\frac{1}{x}\right)y''+3y^2+\left(-6y^2+\frac{3y}{x}-x\right)y'+y^4-\frac{y^3}{x}+y^2x-y$$

> $dsolve(ode_5)$

$$y = \frac{1}{240}\left(-240\,\_C2\,x\,\text{hypergeom}\left(\left[\frac{1}{3},\frac{1}{3}\right],\left[\frac{2}{3},\frac{2}{3},\frac{4}{3}\right],-\frac{1}{9}x^3\right)\right.$$
$$+15\,\_C2\,x^4\,\text{hypergeom}\left(\left[\frac{4}{3},\frac{4}{3}\right],\left[\frac{5}{3},\frac{5}{3},\frac{7}{3}\right],-\frac{1}{9}x^3\right)$$
$$-480\,\_C3\,x^2\,\text{hypergeom}\left(\left[\frac{2}{3},\frac{2}{3}\right],\left[1,\frac{4}{3},\frac{5}{3}\right],-\frac{1}{9}x^3\right)$$
$$+16\,\_C3\,x^5\,\text{hypergeom}\left(\left[\frac{5}{3},\frac{5}{3}\right],\left[2,\frac{7}{3},\frac{8}{3}\right],-\frac{1}{9}x^3\right)$$
$$\left.-720\,\text{MeijerG}\left([[1],[]],\left[\left[\frac{2}{3},\frac{2}{3},\frac{1}{3}\right],[]\right],\frac{1}{9}x^3\right)\right)\Big/$$
$$\left(x\left(\_C1+\_C2\,x\,\text{hypergeom}\left(\left[\frac{1}{3},\frac{1}{3}\right],\left[\frac{2}{3},\frac{2}{3},\frac{4}{3}\right],-\frac{1}{9}x^3\right)\right.\right.$$
$$+\_C3\,x^2\,\text{hypergeom}\left(\left[\frac{2}{3},\frac{2}{3}\right],\left[1,\frac{4}{3},\frac{5}{3}\right],-\frac{1}{9}x^3\right)$$
$$\left.\left.+\text{MeijerG}\left([[1,1],[]],\left[\left[\frac{2}{3},\frac{2}{3},\frac{1}{3}\right],[0]\right],\frac{1}{9}x^3\right)\right)\right)$$

A 2nd order nonlinear ODE problem for which point symmetries exist but are of no use for integration purposes (they involve an unsolved 4th order linear ODE).

> $ode_6 := \frac{d^2}{dx^2}y(x) = x\left(\frac{d}{dx}y(x)\right)^3 y(x) + x^2(x-1)\left(\frac{d}{dx}y(x)\right)^3 + (-3x+1)\left(\frac{d}{dx}y(x)\right)^2$

$$ode_6 := y'' = x y^3 y + x^2(x-1)y^3 + (-3x+1)y^2$$

> $sol_6 := dsolve(ode_6)$

$$sol_6 :=$$
$$\left(\left(\_C2\,\text{AiryAi}\left(\frac{1}{4}-y\right)+\text{AiryBi}\left(\frac{1}{4}-y\right)\right)e^{-\frac{1}{2}y}\right)\Big/\left(\right.$$
$$\int^{y} e^{-\frac{1}{2}\_a}\left(\_C2\,\text{AiryAi}\left(\frac{1}{4}-\_a\right)+\text{AiryBi}\left(\frac{1}{4}-\_a\right)\right)d\_a$$
$$\left.+\_C1\right)+x=0$$

> $odetest(sol_6, ode_6)$

$$0$$

A 3rd order ODE problem illustrating improvements in existing algorithms: in previous releases this equation was only solved in terms of uncomputed integrals of unresolved RootOf expressions; now the solution is computed explicitly as a rational

function

> $ode_7 := \frac{d^3}{dx^3} y(x) + 4\,y(x)\left(\frac{d^2}{dx^2} y(x)\right) + 3\left(\frac{d}{dx} y(x)\right)^2 + 6\,y(x)^2\left(\frac{d}{dx} y(x)\right) + y(x)^4 = 0$

$$ode_7 := y''' + 4\,y''y + 3\,y'^2 + 6\,y'y^2 + y^4 = 0$$

> $sol_7 := dsolve(ode_7)$

$$sol_7 := y = \frac{3\,x^2\_C1 + 6\,\_C2\,x + 6\,\_C3}{x^3\,\_C1 + 3\,\_C2\,x^2 + 6\,\_C3\,x + 6}$$

> $odetest(sol_7, ode_7)$

$$0$$

Anticommutative Variables

For both ordinary and partial differential equations, all symmetry algorithms have been extended to automatically handle problems involving anti-commutative variables, making all this functionality easily available for problems that involve non-commutative variables. These problems often occur in advanced physics computations.

Examples: Ordinary Differential Equations (ODEs)

The dsolve command can now solve ODEs that involve anticommutative variables set using the Physics package.

> $with(Physics, Setup)$

$$[Setup]$$

Set first theta and Q as suffixes for variables of type/anticommutative

> $Setup(anticommutativepre = \{\theta, Q\})$

Partial match of 'anticommutativepre' against keyword 'anticommutativeprefix'

$$[anticommutativeprefix = \{Q, \_\lambda, \theta\}]$$

Consider this ordinary differential equation for the anticommutative function Q of a commutative variable x

> $\frac{d^2}{dx^2} Q(x) - Q(x)\left(\frac{d}{dx} Q(x)\right) = 0$

$$Q'' - Q(x)\,Q' = 0$$

Its solution using dsolve involves an *anticommutative constant* $_lambda1$, analogous to the commutative constants $_C1$

> $dsolve(\mathbf{(23)})$

$$Q(x) = \tan\left(\frac{1}{2}\sqrt{\_C1\ \_\lambda1}\ (x + \_C2)\sqrt{2}\right)\sqrt{\_C1\ \_\lambda1}\ \sqrt{2}$$

Examples: Partial Differential Equations (PDEs)

Many of the commands in PDEtools can now naturally handle anticommutative variables, these are: D_Dx, DeterminingPDE, dsubs, Eta_k, FromJet, FunctionFieldSolutions, InfinitesimalGenerator, Infinitesimals, InvariantSolutions, PolynomialSolutions, ReducedForm, and ToJet, as well as all the routines within the PDEtools[Library]. This makes it possible to tackle super PDE problems, that is, PDE systems involving anticommutative functions and variables. In addition, this permits the use of the PDE mathematical tools with the Physics package.

During the development of this generalization of PDEtools commands to handle anticommutative variables, the symmetry methods were also improved in a number of places. Together, this work sets a new benchmark for state-of-the-art symmetry analysis and the computation of exact solutions for partial differential equations.

Both dsolve and pdsolve can now solve PDEs that involve anticommutative variables set using the Physics package.

> $with(PDEtools), with(Physics)$:

Set first theta; and Q; as suffixes for variables of type/anticommutative

> $Setup(anticommutativepre = \{\theta, Q\})$

** Partial match of 'anticommutativepre' against keyword 'anticommutativeprefix'*

$$[anticommutativeprefix = \{Q, \_\lambda, \theta\}]$$

Consider this partial differential equation for the anticommutative function Q of commutative and anticommutative variables x, theta

> $\frac{\partial^2}{\partial\theta\,\partial x}\,Q(x, y, \theta) = 0$

$$Q_{x,\,\theta} = 0$$

Its solution using pdsolve

> *pdsolve*(**(26)**)

$$Q(x, y, \theta) = \_F2(x, y)\ \_\lambda2 + \_F4(y)\ \theta$$

Note the introduction of an *anticommutative constant* $\_lambda2$, analogous to the commutative constants $\_Cn$ where n is an integer. The arbitrary functions $\_Fn$ introduced are all commutative as usual and the Grassmannian parity (on right-hand-side if compared with the one on the left-hand-side) is preserved

> *Physics*:-*GrassmannParity*(**(27)**)

$$1 = 1$$

Consider this PDE system with one unknown anticommutative function Q, which has of four variables: two commutative and two anticommutative. To avoid redundant typing and display of redundant information on the screen, use PDEtools:-declare, and PDEtools:-diff_table, which also handles anticommutative variables by automatically using Physics:-diff when Physics is loaded

> *PDEtools*:-*declare*$\left(Q\left(x, y, \theta_1, \theta_2\right)\right)$

$$Q\left(x, y, \theta_1, \theta_2\right) \textit{ will now be displayed as } Q$$

> $q := PDEtools\text{:-}diff\_table\left(Q\left(x, y, \theta_1, \theta_2\right)\right)$:

Now we can enter derivatives directly as the function's name indexed by the differentiation variables and see the display the same way; two PDEs

> $pde_1 := q_{x, y, \theta_1} + q_{x, y, \theta_2} - q_{y, \theta_1, \theta_2} = 0$

$$pde_1 := Q_{x, y, \theta_1} + Q_{x, y, \theta_2} - Q_{y, \theta_1, \theta_2} = 0$$

> $pde_2 := q_{\theta_1} = 0$

$$pde_2 := Q_{\theta_1} = 0$$

The solution to this system:

> *pdsolve*$\left(\left[pde_1, pde_2\right]\right)$

$$Q = \_F4(x, y)\ \_\lambda3 + (\_F9(x) + \_F8(y))\ \theta_2$$

The dsubs command also works with anticommutative variables, though natively without using PerformOnAnticommutativeSystem. By inspection, it is clear that the derivatives in pde_2 can be substituted in pde_1 reducing the problem to a simpler one:

> *dsubs*$\left(pde_2, pde_1\right)$

$$Q_{x, y, \theta_2} = 0$$

> $pdsolve(\mathbf{(33)})$

$$Q = \_F4(x,y)\,\_\lambda3 + \_F5(x,y)\,\theta_1 + (\_F9(x) + \_F8(y))\,\theta_2 + (\_F11(x) + \_F10(y))\,\_\lambda4\,\theta_1\,\theta_2$$

Substituting this result for Q back into pde_2, then multiplying by theta_1 and subtracting from the above also leads to the PDE system solution.

Using differential elimination techniques, ReducedForm in this example arrives at the same result as dsubs

> $PDEtools{:}\text{-}ReducedForm(pde_1, pde_2)$

$$\left[Q_{x,y,\theta_2}\right] \&\mathrm{where}\ [\ ]$$

Another command updated to handle anticommutative variables via PerformOnAnticommutativeSystem is casesplit; with this linear system in Q and its derivatives it returns

> $casesplit([pde_1, pde_2])$

$$\left[Q_{x,y,\theta_2} = 0, Q_{\theta_1} = 0\right] \&\mathrm{where}\ [\ ]$$

- With the core functionality in place, most the symmetry commands that work on top can now also handle anticommutative variables, most of them *natively*, without going through PerformOnAnticommutativeSystem.

Set for instance the generic form of the infinitesimals for a PDE system like this one formed by pde_1 and pde_2. For this purpose, we need anticommutative infinitesimals for the dependent variable Q and two of the independent variables, theta_1 and theta_2; we use here the capital greek letters Xi and Eta for the anticommutative infinitesimal symmetry generators and the corresponding lower case greek letters for commutative ones

$$Q_{\theta_1} = 0$$

> $Setup(anticommutativepre = \{\Xi, \mathrm{H}\}, additionally)$

** Partial match of 'anticommutativepre' against keyword 'anticommutativeprefix'*

$$[anticommutativeprefix = \{\mathrm{H}, Q, \Xi, \_\lambda, \theta\}]$$

> $S := [\xi_1, \xi_2, \Xi_1, \Xi_2, \mathrm{H}](x, y, \theta_1, \theta_2)$

$$S := [\xi_1(x,y,\theta_1,\theta_2), \xi_2(x,y,\theta_1,\theta_2), \Xi_1(x,y,\theta_1,\theta_2), \Xi_2(x,y,\theta_1,\theta_2), \mathrm{H}(x,y,\theta_1,\theta_2)]$$

> *PDEtools:-declare*(S)

$$\mathrm{H}(x, y, \theta_1, \theta_2) \textit{ will now be displayed as } \mathrm{H}$$

$$\Xi(x, y, \theta_1, \theta_2) \textit{ will now be displayed as } \Xi$$

$$\xi(x, y, \theta_1, \theta_2) \textit{ will now be displayed as } \xi$$

The corresponding InfinitesimalGenerator

> *InfinitesimalGenerator*$\left(S, Q(x, y, \theta_1, \theta_2)\right)$

$$f \to \text{`*`}\left(\xi_1(x, y, \theta_1, \theta_2), \frac{\partial}{\partial x} f\right) + \text{`*`}\left(\xi_2(x, y, \theta_1, \theta_2), \frac{\partial}{\partial y} f\right) + \text{`*`}\left(\Xi_1(x, y, \theta_1, \theta_2), \frac{\partial}{\partial \theta_1} f\right) + \text{`*`}\left(\Xi_2(x, y, \theta_1, \theta_2), \frac{\partial}{\partial \theta_2} f\right) + \text{`*`}\left(\mathrm{H}(x, y, \theta_1, \theta_2), \frac{\partial}{\partial Q} f\right)$$

The table-function that returns the prolongation of the infinitesimal for Q is computed with Eta_k, assign it here to the lower case eta to use more familiar notation (recall $q_{[\,]} = Q(x, y, \mathrm{theta}_1, \mathrm{theta}_2)$)

> $\eta := \mathit{Eta\_k}\left(S, q_{[\,]}\right)$

$$\eta := \eta$$

The first prolongations of eta with respect to x and theta_1

> $\eta_{Q,\,[x]}$

$$\mathrm{H}_x - \xi_{1,x} Q_x - \xi_{2,x} Q_y - \Xi_{1,x} Q_{\theta_1} - \Xi_{2,x} Q_{\theta_2}$$

> $\eta_{Q,\,[\theta_1]}$

$$\mathrm{H}_{\theta_1} + Q_x \xi_{1,\theta_1} + Q_y \xi_{2,\theta_1} - \Xi_{1,\theta_1} Q_{\theta_1} - \Xi_{2,\theta_1} Q_{\theta_2}$$

The second mixed prolongations of eta with respect to x, y and x, theta_1

> $\eta_{Q,\,[x,y]}$

$$\mathrm{H}_{x,y} - \xi_{1,x,y} Q_x - \xi_{2,x,y} Q_y - \Xi_{1,x,y} Q_{\theta_1} - \Xi_{2,x,y} Q_{\theta_2} - \xi_{1,y} Q_{x,x} - \xi_{2,y} Q_{x,y} - Q_{x,\theta_1} \Xi_{1,y} - Q_{x,\theta_2} \Xi_{2,y} - \xi_{1,x} Q_{x,y} - \xi_{2,x} Q_{y,y} - \Xi_{1,x} Q_{y,\theta_1} - \Xi_{2,x} Q_{y,\theta_2}$$

> $\eta_{Q,\,[x,\theta_1]}$

$$\mathrm{H}_{x,\theta_1} + Q_x\,\xi_{1,x,\theta_1} + Q_y\,\xi_{2,x,\theta_1} - \Xi_{1,x,\theta_1}\,Q_{\theta_1} - \Xi_{2,x,\theta_1}\,Q_{\theta_2} + Q_{x,x}\,\xi_{1,\theta_1} + Q_{x,y}\,\xi_{2,\theta_1} - Q_{x,\theta_1}\,\Xi_{1,\theta_1} - Q_{x,\theta_2}\,\Xi_{2,\theta_1} - \xi_{1,x}\,Q_{x,\theta_1} - \xi_{2,x}\,Q_{y,\theta_1} - Q_{\theta_1,\theta_2}\,\Xi_{2,x}$$

- The DeterminingPDE for this system

> $DeterminingPDE([pde_1, pde_2], S)$

$$\{\mathrm{H}_{\theta_1} = 0, \mathrm{H}_{\theta_1,\theta_2} = 0, \mathrm{H}_{x,y,\theta_2} = 0, \Xi_{1,\theta_1} = -\theta_1\,\xi_{1,x,\theta_1} - \theta_2\,\xi_{1,x,\theta_2} + \xi_{1,x}, \Xi_{1,\theta_2} = \Xi_{2,\theta_2} + \theta_1\,\xi_{1,x,\theta_1} + \theta_2\,\xi_{1,x,\theta_2} - \xi_{1,x}, \Xi_{2,\theta_1} = 0, \Xi_{1,\theta_1,\theta_2} = 0, \Xi_{2,x,\theta_2} = 0, \Xi_{2,y,\theta_2} = 0, \Xi_{2,\theta_1,\theta_2} = 0, \xi_{1,\theta_1,\theta_2} = 0, \xi_{2,\theta_1,\theta_2} = 0, -\xi_{2,\theta_1} = 0, -\xi_{2,\theta_2} = 0, -\xi_{1,\theta_1} + \xi_{1,\theta_1,\theta_2}\,\theta_2 = 0, -\xi_{1,\theta_1,\theta_2}\,\theta_1 - \xi_{1,\theta_2} = 0, \Xi_{1,x} - \Xi_{1,x,\theta_2}\,\theta_2 - \Xi_{1,x,\theta_1}\,\theta_1 = 0, \Xi_{2,x} - \Xi_{2,x,\theta_2}\,\theta_2 - \Xi_{2,x,\theta_1}\,\theta_1 = 0, -\theta_1\,\xi_{1,y,\theta_1} - \theta_2\,\xi_{1,y,\theta_2} + \xi_{1,y} = 0, -\theta_1\,\xi_{2,x,\theta_1} - \theta_2\,\xi_{2,x,\theta_2} + \xi_{2,x} = 0, -\theta_1\,\xi_{1,x,x,\theta_1} - \theta_2\,\xi_{1,x,x,\theta_2} + \xi_{1,x,x} = 0\}$$

To compute now the exact form of the symmetry infinitesimals you can either solve this PDE system for the commutative and anticommutative functions using pdsolve, or directly pass the system to Infinitesimals that will perform all the steps automatically

> $pdsolve(\mathbf{(47)})$

$$\{\mathrm{H} = \_F8(x,y)\,\_\lambda5 + (\_F32(x) + \_F31(y))\,\theta_2, \Xi_1 = \_F28(y)\,\_\lambda7 + \_C2\,\theta_1 + (\_C1 - \_C2)\,\theta_2, \Xi_2 = \_F29(y)\,\_\lambda9 + \_C1\,\theta_2, \xi_1 = \_C2\,x + \_C3, \xi_2 = \_F30(y)\}$$

Substituting into S, you get the symmetry infinitesimal lists

> $subs(\mathbf{(48)}, S)$

$$[\_C2\,x + \_C3, \_F30(y), \_F28(y)\,\_\lambda7 + \_C2\,\theta_1 + (\_C1 - \_C2)\,\theta_2, \_F29(y)\,\_\lambda9 + \_C1\,\theta_2, \_F8(x,y)\,\_\lambda5 + (\_F32(x) + \_F31(y))\,\theta_2]$$

- Infinitesimals also works with anticommutative variables *natively*; the related

result comes specialized

> $Infinitesimals([pde_1, pde_2], q_{[\,]}, S)$

$$[1, \_F4(y), \_F2(y)\ \_\lambda7, \_F3(y)\ \_\lambda9, \_F1(x,y)\ \_\lambda5 + (\_F6(x) + \_F5(y))\ \theta_2], [x, \_F10(y), \_F8(y)\ \_\lambda7 + \theta_1 - \theta_2, \_F9(y)\ \_\lambda9, \_F7(x,y)\ \_\lambda5 + (\_F12(x) + \_F11(y))\ \theta_2], [0, \_F16(y), \_F14(y)\ \_\lambda7 + \theta_2, \_F15(y)\ \_\lambda9 + \theta_2, \_F13(x,y)\ \_\lambda5 + (\_F18(x) + \_F17(y))\ \theta_2]$$

To verify this result you can use SymmetryTest - it also handles anticommutative variables natively.

> $map(SymmetryTest, [(\mathbf{50})], [pde_1, pde_2])$

$$[\{0\}, \{0\}, \{0\}]$$

To see these three list of symmetry infinitesimals with a label in the left-hand-side, you can use map

> $map_3(zip, `=`, S, [(\mathbf{50})])$

$$[[\xi_1 = 1, \xi_2 = \_F4(y), \Xi_1 = \_F2(y)\ \_\lambda7, \Xi_2 = \_F3(y)\ \_\lambda9, \mathrm{H} = \_F1(x,y)\ \_\lambda5 + (\_F6(x) + \_F5(y))\ \theta_2], [\xi_1 = x, \xi_2 = \_F10(y), \Xi_1 = \_F8(y)\ \_\lambda7 + \theta_1 - \theta_2, \Xi_2 = \_F9(y)\ \_\lambda9, \mathrm{H} = \_F7(x,y)\ \_\lambda5 + (\_F12(x) + \_F11(y))\ \theta_2], [\xi_1 = 0, \xi_2 = \_F16(y), \Xi_1 = \_F14(y)\ \_\lambda7 + \theta_2, \Xi_2 = \_F15(y)\ \_\lambda9 + \theta_2, \mathrm{H} = \_F13(x,y)\ \_\lambda5 + (\_F18(x) + \_F17(y))\ \theta_2]]$$

Numerical PDE solutions with compile

The compile option has been added to the numeric pdsolve command, to allow more efficient computation of PDE numerical solutions.

Consider the following example (Klein-Gordon). Using the compile option, the solution is available 6 times faster than before. In many circumstances, even larger speed increases will be seen.

$$PDE := \frac{\partial^2}{\partial t^2} u(x,t) = \frac{\partial^2}{\partial x^2} u(x,t) + a\, e^{\beta u(x,t)} + b\, e^{2\beta u(x,t)}$$

$$u_{t,t} = u_{x,x} + a\, e^{\beta u(x,t)} + b\, e^{2\beta u(x,t)}$$

$$pden := pdsolve\left(PDE \middle| a = 1, b = \frac{1}{2}, \beta = -\frac{1}{2} \right), \left\{ u(0,t) = 0, u(1,t) = 0, u(x,0) = e^{-2x} \sin(\pi x), D_2(u)(x,0) = 0 \right\}, numeric, spacestep = \frac{1}{100}, timestep = \frac{1}{100}, compile = true \Big) :$$

$$pden:\text{-}plot3d(t = 0..10, grid = [40, 200]);$$