Maple 14 における微分方程式 (DE) ソルバの更新

要約

- Maple 14 における常微分方程式の厳密および記号解に関するテーマは、既存の手法の範囲を超えている、2 階非線形 ODE を解く新しいアルゴリズムの開発です。これらはたとえば、点対称ではなく、 $\{x, y, y'\}$ 内の 2 点のみで決まる積分因子も存在しない非線形 ODE 族の求解を指します。このような、以前は処理することができなかった多くの ODE 族は新しいアプローチを使用して 1 階 ODE に約することで解が求められるようになりました。同様に、積分因子を探索して特定の解を計算する新しいアプローチを使用することによって、一般解が求められない ODE についてもさまざまな ODE の初期値問題が解けるようになりました。Maple 14 では、これらの機能は標準 DE ソルバ (dsolve) に組み込まれていますが、Maple 13 では、上記目的のために別途導入された DEtools コマンドを介して利用可能になっています。
- 偏微分方程式については、PDEtools に含まれている対称性に関するコマンドの機能が多くの点で拡張されました。2 つの新しいコマンドが追加された結果、pdsolve は このバージョンから限られた PDE 型 (将来はもっと多くの PDE 型が対応される予定) について境界条件付き厳密解を求めることができます。さらに、Maple 13 から提供されている内部 PDEtools Library には当初の 45 のプログラミング用専用ルーチンに 5 つの新しいプログラミング用ルーチンが追加されました。
- ODE の数値解については、nonstiff (硬くない) 微分および微分代数問題のイベント処理および解法に対する改良が dsolve/numeric になされました。

厳密解

常微分方程式 (ODE)

新しいアルゴリズムの採用により、 dsolve コマンドはこのバージョンから、既存解法の範囲外であったさまざまな 2 階非線形 ODE 族や、以前は計算できなかった ODE-IVP 問題に対して新たな解を求めることができるようになりました。この手法は既存のアルゴリズムでは一般解が求められない場合や ODE が点対称でない場合に使用されます (Maple 13 でも dsolve は ODE-IVP を解くための試みとして点対称性を探索する場合がありました)。

新しく解けるようになった 2 階非線形 ODE 族

新しいアプローチを使用することにより、点対称性がなく、 $\{x, y, y'\}$ 内の 2 点のみで決まる積分因子も存在しない非線形型の方程式がいくつか解けるようになりました。2 階の非線形問題を既知型の1 階の可解な 問題に写像する方法、または 2 階の線形 ODE に写像し、その独立解が元の非線形問題の 2 つの積分因子となる方法を探索し、問題解決の二重の指数削減が可能になります。

アプリケーションと例題

```
> 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 ' (2.1.1.1)

次は、新しいアルゴリズムを使用して非線形問題を非定常な変数式である 1 次のアーベル方程式 (これ自体が難解な問題) に約してから解く例です。

```
> ode[1] := diff(y(x),x,x) = -(diff(y(x),x)^3*x+y(x)*diff(y
  (x),x)^2-diff(y(x),x)*y(x)-x*diff(y(x),x)^2)/x/y(x);
```

(2.1.1.2)

$$ode_1 := y'' = -\frac{y^3 x + y y^2 - y' y - x y^2}{x y} \tag{2.1.1.2}$$

```
> sol[1] := dsolve(ode[1]);
```

$$sol_1 := \_C2 + \frac{1}{y}\left(-2\sqrt{\pi}\, y \operatorname{erf}\left(\frac{1}{2}\frac{(-y+x)\sqrt{2}}{\_C1}\right) + 2\,e^{-\frac{1}{2}\frac{y^2}{\_C1^2}}\, e^{\frac{xy}{\_C1^2}}\, e^{-\frac{1}{2}\frac{x^2}{\_C1^2}}\, \_C1\sqrt{2}\right) = 0 \tag{2.1.1.3}$$

- 次は、別の種類の非線形問題の例です。この例は、2 つの独立解が非線形 ODE の 2 つの独立した積分因子となるよう非線形 ODE を線形 ODE に写像して解きます。

```
> ode[2] := diff(y(x),x,x) = x/y(x)^2/(x*y(x)+1)*diff(y(x),x)^3-(1/y(x)-x)/(x*y(x)+1)*diff(y(x),x)^2+y(x)/(x*y(x)+1)*diff(y(x),x);
```

$$ode_2 := y'' = \frac{x y^3}{y^2 (x y + 1)} - \frac{\left(\frac{1}{y} - x\right) y^2}{x y + 1} + \frac{y y'}{x y + 1} \tag{2.1.1.4}$$

```
> sol[2] := dsolve(ode[2]);
```

$$sol_2 := y = e^{\frac{1}{2}\frac{2\,\mathrm{LambertW}\left(\frac{x\,e^{-\frac{1}{2}\frac{x^2}{\_C1}}\,\frac{\_C2}{\_C1}}{\_C1\,e^{\frac{\_C2}{\_C1}}}\right)\_C1 + x^2 + 2\,\_C2}{\_C1}} \tag{2.1.1.5}$$

以前のリリースでは、 ode_1 and ode_2 は両方とも積分変数に対して反転している未計算積分の項としてのみ解かれていました (解は RootOf で構成された入れ子 Intatとして表現)。

- 次の ODE は、未知数およびその導関数の多項式で、パラメータも点対称性もなく、 $\{x, y, y'\}$ の内の 2 つだけによる積分因子も存在しない、以前のリリースでは解けない族の代表的な例です。

```
> ode[3] := diff(y(x),x,x) = -diff(y(x),x)*(diff(y(x),x)^2*x^2+diff(y(x),x)*x^2+y(x)-x*diff(y(x),x))/x/y(x);
```

$$ode_3 := y'' = -\frac{y'\left(y'^2 x^2 + y' x^2 + y - y' x\right)}{x y} \tag{2.1.1.6}$$

```
> sol[3] := dsolve(ode[3]);
```

$$sol_3 := \_C2 + \frac{x\,\mathrm{KummerM}(-\_C1, 1 - \_C1, y) - \_C1\,e^{y}}{x\,\mathrm{KummerU}(-\_C1, 1 - \_C1, y)} = 0 \tag{2.1.1.7}$$

- ode_3 が点対称でないことを確認するには、対称変換の無限小について determining PDE for the infinitesimals を計算します：無限小 $\_\xi$ および $\_\eta$ に対する唯一の解は、ゼロです。

```
> PDEtools:- DeterminingPDE(ode[3]);
```

$$\{\_\eta_1(x, y) = 0, \_\xi_1(x, y) = 0\} \tag{2.1.1.8}$$

ODE 積分因子から得られる新しい ODE-IVP の解

- いくつかの非線形 ODE 初期値問題 (ODE-IVP) の一般解は既存のアルゴリズムの範囲外でした。Maple 13 の dsolve では、この状況を回避するために、自動的

に ODE の対称性が探索されていました。Maple 14 の dsolve では新しく、計算可能な ODE の積分因子も自動的に探索されます。これにより、ODE の一般解は得られなくても、多くの条件下で ODE-IVP 全体が解けるようになりました。

アプリケーションと例題

- 次に示す ode_4 の一般解は未知ですが、本バージョンから解けるようになった ODE-IVP 問題の例です。

```
> ode[4] := y(x)^p*p^2*diff(y(x),x)^2/y(x)^2+y(x)^p*p*diff(y(x),x,x)/y(x)-y(x)^p*p*diff(y(x),x)^2/y(x)^2 = y(x)+x*diff(y(x),x);
```

$$ode_4 := \frac{y^p p^2 y'^2}{y^2} + \frac{y^p p\, y''}{y} - \frac{y^p p\, y'^2}{y^2} = y + y' x \tag{2.1.2.1}$$

```
> ic[4] := y(0) = 1, D(y)(0) = 0;
```

$$ic_4 := y(0) = 1, \mathrm{D}(y)(0) = 0 \tag{2.1.2.2}$$

```
> sol[4] := dsolve([ode[4], ic[4]]);
```

$$sol_4 := y = \frac{\left(\frac{x^2 p - x^2 + 2p}{p}\right)^{\frac{1}{p-1}}}{2^{\frac{1}{p-1}}} \tag{2.1.2.3}$$

積分因子を使用する DEtools[particularsol]

- $[ode_4, ic_4]$ の ODE-IVP 解は、 DEtools[particularsol] に追加された新しいアルゴリズムで得られる 1 変数の特殊解から算出されます。

```
> particular_sol[4] := DEtools[particularsol](ode[4]);
```

$$particular\_sol_4 := y = 0, y = \mathrm{e}^{\frac{\ln\left(\frac{1}{2}\frac{x^2(p-1)}{p}\right)}{p-1}}, y = \frac{\left(\frac{x^2 p - x^2 + 2\,\_C1\, p}{p}\right)^{\frac{1}{p-1}}}{2^{\frac{1}{p-1}}} \tag{2.1.3.1}$$

- 同様に DEtools[IVPsol] の新機能は、一般解を特殊化する DEtools[particularsol] や、 dsolve の関連オプション solution (両方とも Maple 13 で追加) にも生かされます。

```
> DEtools[IVPsol]([ic[4]], particular_sol[4][-1]);
```

$$y = \frac{\left(\frac{x^2 p - x^2 + 2p}{p}\right)^{\frac{1}{p-1}}}{2^{\frac{1}{p-1}}} \tag{2.1.3.2}$$

DEtools[firint] による線形 ODE の 1 階積分の集合の計算

- DEtools[firint] コマンドは線形 ODE の 1 階積分の集合が計算できるようになりました。すなわち、互いに独立した、ODE の階数と同じ数だけの 1 階積分が計算できるようになりました。

```
> ode[5] := diff(y(x),x,x) = (-6*x-6)/x/(3*x+2)*diff(y(x),x)
  +6/x/(3*x+2)*y(x)-6/x/(3*x+2);
```

$$ode_5 := y'' = \frac{(-6\,x-6)\,y'}{x\,(3\,x+2)} + \frac{6\,y}{x\,(3\,x+2)} - \frac{6}{x\,(3\,x+2)} \tag{2.1.4.1}$$

```
> DEtools[firint](ode[5], method = formal);
```

$$-\frac{x^3\,((x+1)\,y'-y+1)}{3\,x+2},\ \frac{y'x+2\,y-2}{3\,x+2} \tag{2.1.4.2}$$

次は、典型的な例です：

```
> PDEtools[declare](g(x));
```

$g(x)$ *will now be displayed as* g (2.1.4.3)

```
> ode[6] := diff(y(x), x,x) = ((diff(g[1](x), x,x))*g[2](x)-g
[1](x)*(diff(g[2](x), x,x)))*(diff(y(x), x))/(g[2](x)*(diff
(g[1](x), x))-g[1](x)*(diff(g[2](x), x)))+((diff(g[1](x),
x))*(diff(g[2](x), x,x))-(diff(g[1](x), x,x))*(diff(g[2]
(x), x)))*y(x)/(g[2](x)*(diff(g[1](x), x))-g[1](x)*(diff(g
[2](x), x)))+((diff(g[0](x), x,x))*(-g[2](x)*(diff(g[1](x),
x))+g[1](x)*(diff(g[2](x), x)))+((diff(g[1](x), x))*g[0](x)
-g[1](x)*(diff(g[0](x), x)))*(diff(g[2](x), x,x))-((diff(g
[2](x), x))*g[0](x)-g[2](x)*(diff(g[0](x), x)))*(diff(g[1]
(x), x,x)))/(-g[2](x)*(diff(g[1](x), x))+g[1](x)*(diff(g[2]
(x), x)));
```

$$ode_6 := y'' = \frac{(g_1'' g_2 - g_1 g_2'')\,y'}{g_2 g_1' - g_1 g_2'} + \frac{(g_1' g_2'' - g_1'' g_2')\,y}{g_2 g_1' - g_1 g_2'} + \frac{g_0''\,(-g_2 g_1' + g_1 g_2') + (g_1' g_0 - g_1 g_0')\,g_2'' - (g_2' g_0 - g_2 g_0')\,g_1''}{-g_2 g_1' + g_1 g_2'} \tag{2.1.4.4}$$

このode_6は、斉次部分については解として基底 $g_1(x)$ および $g_2(x)$ を持ち、非斉次方程式全体の特殊解として $g_0(x)$ を持つ最も一般的な 2 階非斉次線形 ODE です。解の基底は以下のように表現されます。

```
> basis_sol[6] := [[g[1](x), g[2](x)], g[0](x)];
```

$$basis\_sol_6 := [[g_1, g_2], g_0] \tag{2.1.4.5}$$

$basis\_sol_6$ が ode_6 の完全基底であるか確認するには、 $basis\_sol_6$ から一般解を構築します。次に、odetest を使用して検証します。

```
> sol[6] := y(x) = _C1*g[1](x) + _C1*g[2](x) + g[0](x);
```

$$sol_6 := y = \_C1\ g_1 + \_C1\ g_2 + g_0 \tag{2.1.4.6}$$

```
> odetest(sol[6], ode[6]);
```

$$0 \tag{2.1.4.7}$$

次は、ode_6 の 1 階積分の完全な集合を計算します。

```
> first_integrals[6] := DEtools[firint](ode[6], y(x), basis =
  basis_sol[6]);
```

$$first\_integrals_6 := \frac{(y-g_0)\,g_2' - g_2\,(y'-g_0')}{-g_2 g_1' + g_1 g_2'},\ \frac{(-y+g_0)\,g_1' + g_1\,(y'-g_0')}{-g_2 g_1' + g_1 g_2'} \tag{2.1.4.8}$$

結果を検証します。

```
> map(DEtools[firtest], [first_integrals[6]], ode[6], y(x));
```

$$[0, 0] \tag{2.1.4.9}$$

偏微分方程式 (PDE)

PDEtools パッケージについては、数多くの改良と追加により最先端の記号計算および偏微分方程式の解法に新たな基準を設けています。

境界条件付き PDE の厳密解

- コンピュータを使用した微分方程式の厳密解の解法開発の先頭を切って、Maple 14 の pdsolve では境界条件付き PDE の厳密解が計算できるようになりました (pdsolve[boundaryconditions]参照)。ルーチンは実験的であり、2 つの独立変数を持つ PDE を主な対象としていますが、標準的な問題もいくつか処理できます。処理可能な問題は今後増える予定です。

アプリケーションと例題

```
> sys[1] := [diff(u(x, t), t)+c*(diff(u(x, t), x)) = -lambda*u(x, t), u(x, 0) = phi(x)];
```

$$sys_1 := \left[u_t + c\,u_x = -\lambda\,u(x,t),\ u(x,0) = \phi(x)\right] \tag{2.2.1.1}$$

```
> pdsolve(sys[1]);
```

$$u(x,t) = \phi(x - t\,c)\,\mathrm{e}^{-\lambda t} \tag{2.2.1.2}$$

pdetestを使用してこの結果を検証します。

```
> pdetest((2.2.1.2), sys[1]);
```

$$[0, 0] \tag{2.2.1.3}$$

0 および Lで周期的な境界条件を持つ熱伝導方程式

```
> sys[2] := [diff(u(x, t), t) = kappa*(diff(u(x, t), x, x)), u(0, t) = 0, u(L, t) = 0];
```

$$sys_2 := \left[u_t = \kappa\,u_{x,x},\ u(0,t) = 0,\ u(L,t) = 0\right] \tag{2.2.1.4}$$

```
> pdsolve(sys[2]) assuming L > 0;
```

$$u(x,t) = \_C3\,\sin\left(\frac{\pi\,x\,|\_Z1\sim|}{L}\right)\_C1\,\mathrm{e}^{-\frac{\kappa\pi^2\,\_Z1\sim^2\,t}{L^2}} \tag{2.2.1.5}$$

```
> pdetest((2.2.1.5), sys[2]);
```

$$[0, 0, 0] \tag{2.2.1.6}$$

次は、波動方程式型の問題の例です。

```
> sys[3] := [diff(u(x,t), x, x) - (1/c^2)*diff(u(x,t), t, t) = 0, u(x,0) = f(x), D[2](u)(x, 0) = g(x)];
```

$$sys_3 := \left[u_{x,x} - \frac{u_{t,t}}{c^2} = 0,\ u(x,0) = f(x),\ \mathrm{D}_2(u)(x,0) = g\right] \tag{2.2.1.7}$$

```
> pdsolve(sys[3]);
```

$$u(x,t) = \frac{1}{2}\,\frac{f(x - t\,c)\,c - \left(\int_0^{x - t\,c} g(x1)\,\mathrm{d}x1\right) + f(t\,c + x)\,c + \int_0^{t\,c + x} g(x1)\,\mathrm{d}x1}{c} \tag{2.2.1.8}$$

```
> pdetest((2.2.1.8), sys[3]);
```

$$[0, 0, 0] \tag{2.2.1.9}$$

境界条件付き PDE システムの級数解

- pdsolve コマンドは Maple 14 から初期値を持つ PDE の級数解を計算できるようになりました (pdsolve[series]参照)。

アプリケーションと例題

次の 2 次元熱伝導方程式を検討します。

```
> pde[1] := diff(u(x, t), t,t) - diff(u(x, t), x) = 0;
```

$$pde_1 := u_{t,t} - u_x = 0 \tag{2.2.2.1}$$

古典的には、熱伝導方程式の解は 2 つの任意関数 $u(x, 0) = f(x)$ および $u_t(x, 0) = g(x)$ に依存します。説明のために、初期値として $f = \cos, g = \sin$を選択します。

```
> iv[1] := u(x, 0) = cos(x), D[2](u)(x, 0) = sin(x);
```

$$iv_1 := u(x, 0) = \cos(x), \mathrm{D}_2(u)(x, 0) = \sin(x) \tag{2.2.2.2}$$

```
> pdsolve([pde[1], iv[1]] , series, order = 3);
```

$$u(x, t) = -\frac{1}{2}x^2 + \frac{1}{6}(6t - 3t^2)x + 1 + \frac{1}{6}t^3 \tag{2.2.2.3}$$

pdsolve は新しい DifferentialAlgebra パッケージの PowerSeriesSolution コマンドを使用してこれらのべき級数解を計算します。その上、pdsolve は拡張的アプローチを使用して dpolyform で説明されている非多項式型微分方程式も処理できます。すなわち、PowerSeriesSolution と異なり pdsolve は多項式以外の項が存在しても級数解を計算できます。たとえば、$x = x0$ 前後の初期値を次の方程式に与えた場合を検討します。

```
> pde[2] := diff(u(x,t),x,x) = BesselJ(nu, sin(x))*diff(u(x, t), t);
```

$$pde_2 := u_{x,x} = \mathrm{BesselJ}(\nu, \sin(x))\, u_t \tag{2.2.2.4}$$

```
> iv[2] := u(x0, t) = f(t), D[1](u)(x0, t0)=1, D[2](u)(x0, t0) = 2;
```

$$iv_2 := u(x0, t) = f(t), \mathrm{D}_1(u)(x0, t0) = 1, \mathrm{D}_2(u)(x0, t0) = 2 \tag{2.2.2.5}$$

```
> pdsolve({pde[2], iv[2]}, u(x,t), series, order = 3);
```

$$u(x, t) = f(t0) + 2t - 2t0 + x - x0 + \mathrm{BesselJ}(\nu, \sin(x0))(x - x0)^2 + \frac{1}{3}\left(-\mathrm{BesselJ}(\nu+1, \sin(x0)) + \frac{\nu\,\mathrm{BesselJ}(\nu, \sin(x0))}{\sin(x0)}\right)\cos(x0)(x - x0)^3 \tag{2.2.2.6}$$

PDEtools パッケージの新規コマンド：Solve および InvariantEquation

- PDEtools には 2 つの新しいコマンドが追加されました。
  - Solve；このコマンドの目的は 2 つです：1 つ目の目的は solve、dsolve および pdsolve の機能を 1 つのコマンドで提供し、ユーザーの入力から実際に呼び出すコマンドを判断することで、2 つ目の目的は非多項式型の代数方程式システムと微分方程式システムに対し solve[identity] および、任意数の identity 変数で提供した機能を一般化し、指定した変数に依存しない解を計算する新規の関連機能を提供することです。
  - InvariantEquation ；与えられた対称群を持つことができる不変式を計算します。
- PDE システムの操作と PDEtools ライブラリのコマンドを使用したプログラム作成に役立つ 5 つの新しい PDEtools:-Library コマンドが追加されました。これらは次のとおりです：JetNumbersToJetVariables、JetVariablesToJetNumbers、NormalizeBoundaryConditions、SolveCase、ToSameVariables

アプリケーションと例題

```
> with(PDEtools):
```

次は、微分方程式ではない式の例です。

```
> eq[1] := a*x^2 + b*x + c;
```

$$eq_1 := a\,x^2 + b\,x + c \tag{2.2.3.1}$$

```
> Solve(eq[1], x);
```

$$x = \frac{1}{2}\,\frac{-b+\sqrt{b^2-4\,a\,c}}{a},\ x = -\frac{1}{2}\,\frac{b+\sqrt{b^2-4\,a\,c}}{a} \tag{2.2.3.2}$$

次は、ODE 問題とその級数解の例です。

```
> eq[2] := diff(y(x),x) = y(x);
```

$$eq_2 := y' = y \tag{2.2.3.3}$$

```
> Solve(eq[2], series);
```

$$y = y(0) + y(0)\,x + \frac{1}{2}\,y(0)\,x^2 + \frac{1}{6}\,y(0)\,x^3 + \frac{1}{24}\,y(0)\,x^4 + \frac{1}{120}\,y(0)\,x^5 + \mathrm{O}(x^6) \tag{2.2.3.4}$$

次は、境界条件付きの PDE 問題の例です。

```
> eq[3] := [diff(u(x, t), t)+c*(diff(u(x, t), x)) = -lambda*u
  (x, t), u(x, 0) = phi(x)];
```

$$eq_3 := \left[u_t + c\,u_x = -\lambda\,u(x,t),\ u(x,0) = \phi(x)\right] \tag{2.2.3.5}$$

```
> Solve(eq[3]);
```

$$u(x,t) = \phi(x - t\,c)\,\mathrm{e}^{-\lambda t} \tag{2.2.3.6}$$

上記例は solve、dsolve または pdsolve を 1 つのコマンドで呼び出し、入力および出力に対して統一された形式を持つ利点を示しています。Solve は追加機能も提供します：与えられた変数に依存しない解を計算できます。independentof が指定されていると、不等式が含まれていてもシステムを解くことができます。

```
> eq[4] := [k*a*c*(a+b)*exp(k*d*t)-2*a*exp(k*t)*k+Q*(-c+a)*x,
  a <> 0];
```

$$eq_4 := \left[k\,a\,c\,(a+b)\,\mathrm{e}^{k\,d\,t} - 2\,a\,\mathrm{e}^{k\,t}\,k + Q\,(-c+a)\,x,\ a \neq 0\right] \tag{2.2.3.7}$$

```
> Solve(eq[4], {a, b, c, d}, independentof = {t, x});
```

$$\left\{a = a,\ b = -\frac{a^2-2}{a},\ c = a,\ d = 1\right\} \tag{2.2.3.8}$$

微分方程式または微分方程式システムについても与えられた変数と独立した解を計算できます。

```
> eq[5] := diff(f(x,y),x)*diff(g(x,y),x) + diff(f(x,y),y)*
  diff(g(x,y),y) + g(x,y)*(diff(f(x,y), x,x) + diff(f(x,y),
  y,y)) = -1;
```

$$eq_5 := f_x\,g_x + f_y\,g_y + g(x,y)\,\left(f_{x,x} + f_{y,y}\right) = -1 \tag{2.2.3.9}$$

次は、x にも、y にも依存しない PDE の解の例です。

```
> Solve(eq[5], independentof = x);
```

$$\left\{f(x,y) = \_F1(y),\ g(x,y) = \frac{-y + \_C1}{\_F1_y}\right\} \tag{2.2.3.10}$$

```
> Solve(eq[5], independentof = y);
```

$$\left\{f(x,y) = \_F1(x),\ g(x,y) = \frac{-x + \_C1}{\_F1'}\right\} \tag{2.2.3.11}$$

- 新しい InvariantEquations コマンドは与えられた対称での不変式を計算します。以下に示す無限小の 1 次元対称群のリストについて検討します。

```
> S := [x, 1, u];
```

(2.2.3.12)

$$S := [x, 1, u] \tag{2.2.3.12}$$

これらの無限小の対称変換の基礎となっている 1 階偏微分の不変式は次のようになります。

```
> PDE := InvariantEquation(S, u(x, t), arbitraryfunctionname
  = Lambda);
```

$$PDE := \Lambda\left(-\ln(x) + t, \frac{u(x,t)}{x}, u_x, \frac{u_t}{x}\right) \tag{2.2.3.13}$$

S における 3 階 PDE の不変式は次のようになります。

```
> PDE := InvariantEquation(S, u(x, t), arbitraryfunctionname
  = Lambda, order = 3);
```

$$PDE := \Lambda\left(-\ln(x) + t, \frac{u(x,t)}{x}, u_x, \frac{u_t}{x}, u_{t,x}, u_{x,x}x, \frac{u_{t,t}}{x}, u_{t,t,x}, u_{x,x,x}x^2, u_{t,x,x}x, \frac{u_{t,t,t}}{x}\right) \tag{2.2.3.14}$$

PDE の不変性はいくつかの方法で検証できます。最も単純な方法は SymmetryTestを使用して S と PDE が対称であると検証する方法だと思われます。

```
> SymmetryTest(S, PDE);
```

$$\{0\} \tag{2.2.3.15}$$

InvariantEquations を使用して n 次対称群による式の不変量を計算することもできます。たとえば、次の、2 つの従属変数 $u(x,t), v(x,t)$ を含む問題の 5 つの対称式のリストを検討します。

```
> G := [[_xi[x] = 0, _xi[t] = 1, _eta[u] = 0, _eta[v] = 0],
  [_xi[x] = 0, _xi[t] = 0, _eta[u] = 1, _eta[v] = 0], [_xi[x]
  = 1, _xi[t] = 0, _eta[u] = 2*t, _eta[v] = 0], [_xi[x] = t,
  _xi[t] = 0, _eta[u] = t^2+x, _eta[v] = 0], [_xi[x] = 1/2*
  x+3/2*t^2, _xi[t] = t, _eta[u] = t^3+3*t*x, _eta[v] = -v]];
```

$$G := \left[\left[\_\xi_x = 0, \_\xi_t = 1, \_\eta_u = 0, \_\eta_v = 0\right], \left[\_\xi_x = 0, \_\xi_t = 0, \_\eta_u = 1, \_\eta_v = 0\right], \left[\_\xi_x = 1, \_\xi_t = 0, \_\eta_u = 2\,t, \_\eta_v = 0\right], \left[\_\xi_x = t, \_\xi_t = 0, \_\eta_u = t^2 + x, \_\eta_v = 0\right], \left[\_\xi_x = \frac{1}{2}\,x + \frac{3}{2}\,t^2, \_\xi_t = t, \_\eta_u = t^3 + 3\,x\,t, \_\eta_v = -v\right]\right] \tag{2.2.3.16}$$

関連の 1 階不変 PDE (1st order invariant PDE) 族は次のようになります：

```
> PDE := InvariantEquation(G, [u, v](x, t),
  arbitraryfunctionname = Lambda);
```

$$PDE := \Lambda\left(\frac{v_x}{v(x,t)^{3/2}}, \frac{-4\,x + u_x^2 + 2\,u_t}{v(x,t)}, \frac{u_x\,v_x + v_t}{v(x,t)^2}\right) \tag{2.2.3.17}$$

```
> map(SymmetryTest, G, PDE, [u, v](x, t));
```

$$[\{0\}, \{0\}, \{0\}, \{0\}, \{0\}] \tag{2.2.3.18}$$

数値解

- ODE の初期値問題の数値解法が追加されました；Cash-Karp 対で、 ck45 法として提供されています。この対は、硬くない (nonstiff) ODE 問題を解くときに便利です。また、ck45_dae と呼ばれるこの手法の拡張が微分代数方程式問題を解く目的で追加されています。また、少し硬い (mildly stiff) 問題に関しては rkf45 法より ck45 法の方がパフォーマンスが若干良い傾向が見られます。
- 硬くない

- (nonstiff) ODE 向けのデフォルトの解法は rkf45 法ですが、オプション method を指定することで ck45 法 (とその DAE 向け拡張である ck45_dae) 法) を呼び出すことができます。：

```
> dsys1 := {diff(x(t),t)=y(t), diff(y(t),t)=x(t)+y(t),x(0)=2, y(0)=1};
```

$$dsys1 := \{x_t = y(t), y_t = x(t) + y(t), x(0) = 2, y(0) = 1\} \tag{3.1}$$

```
> dsn := dsolve(dsys1, numeric, method=ck45);
```

$$dsn := \mathbf{proc}(x\_ck45)\ \ldots\ \mathbf{end\ proc} \tag{3.2}$$

```
> dsn(2);
```

$$[t = 2., x(t) = 25.7240237105037, y(t) = 40.9727125531702] \tag{3.3}$$

- 詳細については、 dsolve/ck45 を参照してください。

関連項目

Index of New Maple 14 Features