ODEs and PDEs

"Computer algebra systems have evolved into powerful solving environments for studying and solving differential equations."

Some polemical questions:

Can a Computer Algebra system compute numerical ODE solutions *as fast* as for instance C or FORTRAN code ?

Yes, including fast symbolic preprocessing that simplifies the system and is not possible in C. Both ODEs and PDEs can be solved numerically

Multiple-weight pendulum (50 weights)

Set parameters for system construction

```
> restart;
> nwgt := 50; # Number of weights
```

$$nwgt := 50 \tag{1.1.1.1}$$

```
> ndim := 2:  # Number of dimensions
  gdim := 2:  # Dimension in which gravity operates
  grav := -g: # Gravity value
```

Set up variable name mapping: position of weight 'i' will be $x_i(t), y_i(t)$ in cartesian coordinates

```
> vars := [x, y, z]:
  dvars := [xpp, ypp, zpp]:
  sbs := seq(seq(vars[i][j]=vars[i][j](t),i=1..ndim),j=1..
  nwgt),
       seq(seq(dvars[i][j]=diff(vars[i][j](t),t,t),i=1..ndim),
  j=1..nwgt):
```

Construct the constraints.
These correspond to a position constraint for each weight that enforces the length of the string connecting each pair of weights.
Note that these each have an implied velocity constraint (that the motion is perpendicular to the connecting string), but this constraint is figured out by dsolve.

```
> cons := NULL:
  for i to nwgt do
   if i=1 then
      cons := cons,add(vars[j][i]^2,j=1..ndim)-l[i]^2;
   else
      cons := cons,add((vars[j][i]-vars[j][i-1])^2,j=1..ndim)-l
  [i]^2;
   end if:
  end do:
  cons := subs(seq(l[i]=1,i=1..nwgt),[cons]):
```

Now construct the DEs, two second order DEs for each weight.

The method employed here is Lagrange multipliers (these are $\lambda_i(t)$)

```
> des := NULL:
  for i to nwgt do
   for j to ndim do
      de := m[i]*dvars[j][i]-`if`(j=gdim, m[i]*grav, 0);
      for k to nwgt do
         de := de + factor(lambda[k](t)*diff(cons[k], vars[j]
  [i])):
      end do:
      des := des,de:
   end do:
  end do:
  des := subs(seq(m[i]=1, i=1..nwgt), g=98/10, [des]):
```

The number of ODEs is then

```
> nops(des);
```

$$100 \tag{1.1.1.2}$$

Combine the constraints and DEs to form a DAE system, and compute the initial conditions. For initial conditions we are simply holding the string of masses stationary at the angle of θ_0 off vertical. A value of $\theta_0 = 0$ would have all masses in a vertical line.

```
> cons := subs(sbs,cons):
  des := subs(sbs,des):
  dsys := [op(des),op(cons)]:
  theta0 := Pi/7:
  ini := [seq(x[i](0)=i*evalf(sin(theta0)),i=1..nwgt),
        seq(y[i](0)=-i*evalf(cos(theta0)),i=1..nwgt),
        seq(D(x[i])(0)=0,i=1..nwgt),
        seq(D(y[i])(0)=0,i=1..nwgt)]:
```

Give a look at a sample equations and initial conditions - say for the second mass:

```
> op(3..4,des);
  cons[2];
  op(2,ini), op(2+nwgt,ini), op(2+2*nwgt,ini), op(2+3*nwgt,ini)
  ;
```

$$\ddot{x}_2(t) - 2\lambda_2(t)\left(-x_2(t) + x_1(t)\right) + 2\lambda_3(t)\left(-x_3(t) + x_2(t)\right), \frac{49}{5} + \ddot{y}_2(t) - 2\lambda_2(t)\left(-y_2(t) + y_1(t)\right) + 2\lambda_3(t)\left(-y_3(t) + y_2(t)\right)$$

$$\left(x_2(t) - x_1(t)\right)^2 + \left(y_2(t) - y_1(t)\right)^2 - 1$$

$$x_2(0) = 0.8677674786, y_2(0) = -1.801937736, \mathrm{D}(x_2)(0) = 0, \mathrm{D}(y_2)(0) = 0 \tag{1.1.1.3}$$

Now we call dsolve with numeric and compile options. Use the rosenbrock_dae method because the problem is very stiff: it takes approx 50 seconds to complete the compilation

```
> tt := time():
  dsn := dsolve({op(dsys), op(ini)}, numeric, implicit = true,
  method = rosenbrock_dae, optimize = true, compile = true):
  `Solution time` = 3*(time()-tt);
```

$$\textit{Solution time} = 54.984 \tag{1.1.1.4}$$

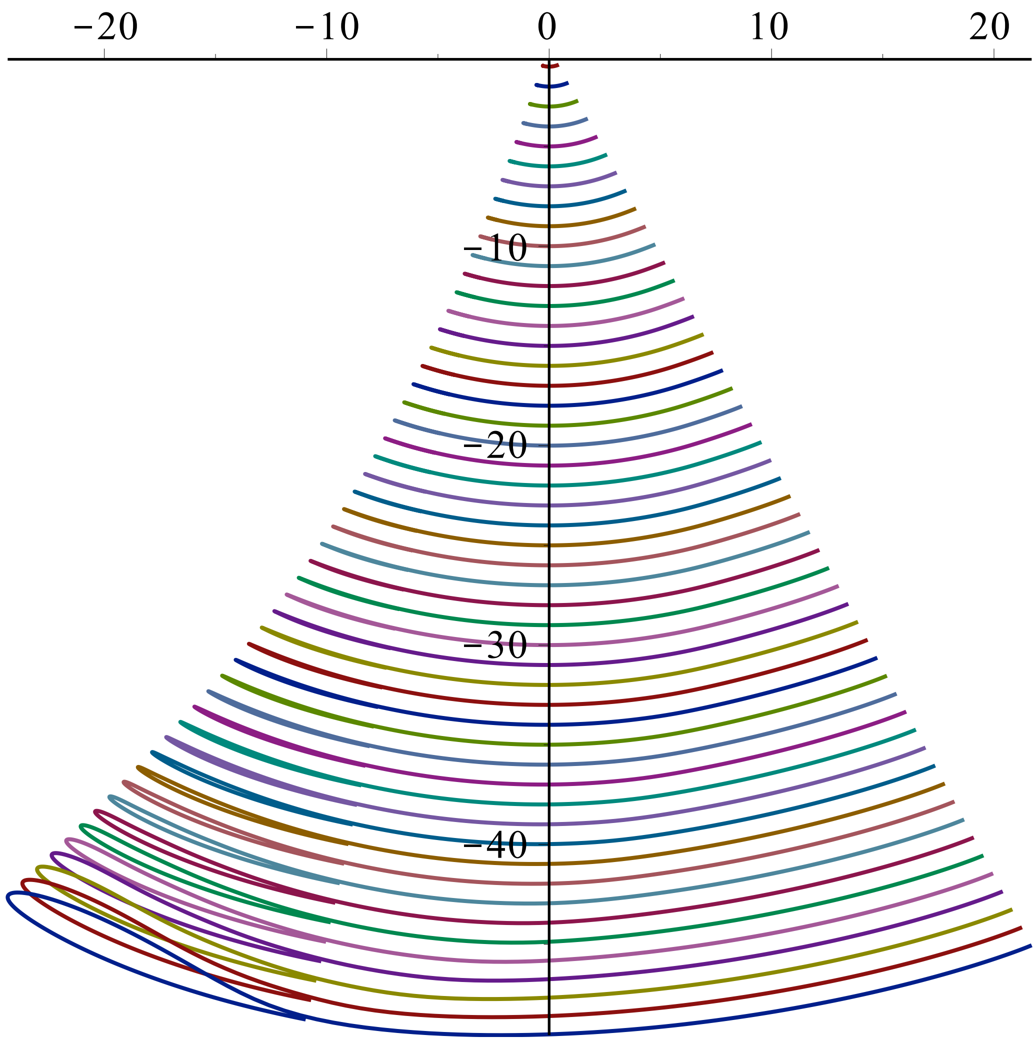
Now we obtain an animation of the solution - each curve in the animation represents the path of one of the weights.

Note that the numerical integration is performed here.

```
> all_vars := [seq([x[i](t), y[i](t)], i=1..nwgt)];
```

$$all\_vars := [[x_1(t), y_1(t)], [x_2(t), y_2(t)], [x_3(t), y_3(t)], [x_4(t), y_4(t)], [x_5(t), y_5(t)], [x_6(t), y_6(t)], [x_7(t), y_7(t)], [x_8(t), y_8(t)], [x_9(t), y_9(t)], [x_{10}(t), y_{10}(t)], [x_{11}(t), y_{11}(t)], [x_{12}(t), y_{12}(t)], [x_{13}(t), y_{13}(t)], [x_{14}(t), y_{14}(t)], [x_{15}(t), y_{15}(t)], [x_{16}(t), y_{16}(t)], [x_{17}(t), y_{17}(t)], [x_{18}(t), y_{18}(t)], [x_{19}(t), y_{19}(t)], [x_{20}(t), y_{20}(t)], [x_{21}(t), y_{21}(t)], [x_{22}(t), y_{22}(t)], [x_{23}(t), y_{23}(t)], [x_{24}(t), y_{24}(t)], [x_{25}(t), y_{25}(t)], [x_{26}(t), y_{26}(t)], [x_{27}(t), y_{27}(t)], [x_{28}(t), y_{28}(t)], [x_{29}(t), y_{29}(t)], [x_{30}(t), y_{30}(t)], [x_{31}(t), y_{31}(t)], [x_{32}(t), y_{32}(t)], [x_{33}(t), y_{33}(t)], [x_{34}(t), y_{34}(t)], [x_{35}(t), y_{35}(t)], [x_{36}(t), y_{36}(t)], [x_{37}(t), y_{37}(t)], [x_{38}(t), y_{38}(t)], [x_{39}(t), y_{39}(t)], [x_{40}(t), y_{40}(t)], [x_{41}(t), y_{41}(t)], [x_{42}(t), y_{42}(t)], [x_{43}(t), y_{43}(t)], [x_{44}(t), y_{44}(t)], [x_{45}(t), y_{45}(t)], [x_{46}(t), y_{46}(t)], [x_{47}(t), y_{47}(t)], [x_{48}(t), y_{48}(t)], [x_{49}(t), y_{49}(t)], [x_{50}(t), y_{50}(t)]] \quad \textbf{(1.1.1.5)}$$

```
> tt := time():
  plots[odeplot](dsn, all_vars, 0..8);
  `Integration time` = time()-tt;
```



Integration time = 3.672 **(1.1.1.6)**

Change the initial conditions ...

```
> theta0 := 0:
> ini := [seq(x[i](0)=i*evalf(sin(theta0)),i=1..nwgt),
>        seq(y[i](0)=-i*evalf(cos(theta0)),i=1..nwgt),
>        seq(D(x[i])(0)=5*(1-2*i/nwgt),i=1..nwgt),
>        seq(D(y[i])(0)=0,i=1..nwgt)];
```

$$ini := \left[x_1(0) = 0., x_2(0) = 0., x_3(0) = 0., x_4(0) = 0., x_5(0) = 0., x_6(0) = 0., x_7(0)\right. \quad \textbf{(1.1.1.7)}$$

$$= 0., x_8(0) = 0., x_9(0) = 0., x_{10}(0) = 0., x_{11}(0) = 0., x_{12}(0) = 0., x_{13}(0) = 0.,$$

$$x_{14}(0) = 0., x_{15}(0) = 0., x_{16}(0) = 0., x_{17}(0) = 0., x_{18}(0) = 0., x_{19}(0) = 0.,$$

$$x_{20}(0) = 0., x_{21}(0) = 0., x_{22}(0) = 0., x_{23}(0) = 0., x_{24}(0) = 0., x_{25}(0) = 0.,$$

$$x_{26}(0) = 0., x_{27}(0) = 0., x_{28}(0) = 0., x_{29}(0) = 0., x_{30}(0) = 0., x_{31}(0) = 0.,$$

$x_{32}(0) = 0., x_{33}(0) = 0., x_{34}(0) = 0., x_{35}(0) = 0., x_{36}(0) = 0., x_{37}(0) = 0.,$

$x_{38}(0) = 0., x_{39}(0) = 0., x_{40}(0) = 0., x_{41}(0) = 0., x_{42}(0) = 0., x_{43}(0) = 0.,$

$x_{44}(0) = 0., x_{45}(0) = 0., x_{46}(0) = 0., x_{47}(0) = 0., x_{48}(0) = 0., x_{49}(0) = 0.,$

$x_{50}(0) = 0., y_1(0) = -1., y_2(0) = -2., y_3(0) = -3., y_4(0) = -4., y_5(0)$

$= -5., y_6(0) = -6., y_7(0) = -7., y_8(0) = -8., y_9(0) = -9., y_{10}(0) = -10.,$

$y_{11}(0) = -11., y_{12}(0) = -12., y_{13}(0) = -13., y_{14}(0) = -14., y_{15}(0)$

$= -15., y_{16}(0) = -16., y_{17}(0) = -17., y_{18}(0) = -18., y_{19}(0) = -19.,$

$y_{20}(0) = -20., y_{21}(0) = -21., y_{22}(0) = -22., y_{23}(0) = -23., y_{24}(0)$

$= -24., y_{25}(0) = -25., y_{26}(0) = -26., y_{27}(0) = -27., y_{28}(0) = -28.,$

$y_{29}(0) = -29., y_{30}(0) = -30., y_{31}(0) = -31., y_{32}(0) = -32., y_{33}(0)$

$= -33., y_{34}(0) = -34., y_{35}(0) = -35., y_{36}(0) = -36., y_{37}(0) = -37.,$

$y_{38}(0) = -38., y_{39}(0) = -39., y_{40}(0) = -40., y_{41}(0) = -41., y_{42}(0)$

$= -42., y_{43}(0) = -43., y_{44}(0) = -44., y_{45}(0) = -45., y_{46}(0) = -46.,$

$y_{47}(0) = -47., y_{48}(0) = -48., y_{49}(0) = -49., y_{50}(0) = -50., \mathrm{D}(x_1)(0)$

$= \frac{24}{5}, \mathrm{D}(x_2)(0) = \frac{23}{5}, \mathrm{D}(x_3)(0) = \frac{22}{5}, \mathrm{D}(x_4)(0) = \frac{21}{5}, \mathrm{D}(x_5)(0) = 4,$

$\mathrm{D}(x_6)(0) = \frac{19}{5}, \mathrm{D}(x_7)(0) = \frac{18}{5}, \mathrm{D}(x_8)(0) = \frac{17}{5}, \mathrm{D}(x_9)(0) = \frac{16}{5},$

$\mathrm{D}(x_{10})(0) = 3, \mathrm{D}(x_{11})(0) = \frac{14}{5}, \mathrm{D}(x_{12})(0) = \frac{13}{5}, \mathrm{D}(x_{13})(0) = \frac{12}{5},$

$\mathrm{D}(x_{14})(0) = \frac{11}{5}, \mathrm{D}(x_{15})(0) = 2, \mathrm{D}(x_{16})(0) = \frac{9}{5}, \mathrm{D}(x_{17})(0) = \frac{8}{5},$

$$D(x_{18})(0) = \frac{7}{5}, D(x_{19})(0) = \frac{6}{5}, D(x_{20})(0) = 1, D(x_{21})(0) = \frac{4}{5},$$

$$D(x_{22})(0) = \frac{3}{5}, D(x_{23})(0) = \frac{2}{5}, D(x_{24})(0) = \frac{1}{5}, D(x_{25})(0) = 0,$$

$$D(x_{26})(0) = -\frac{1}{5}, D(x_{27})(0) = -\frac{2}{5}, D(x_{28})(0) = -\frac{3}{5}, D(x_{29})(0) = -\frac{4}{5},$$

$$D(x_{30})(0) = -1, D(x_{31})(0) = -\frac{6}{5}, D(x_{32})(0) = -\frac{7}{5}, D(x_{33})(0) = -\frac{8}{5},$$

$$D(x_{34})(0) = -\frac{9}{5}, D(x_{35})(0) = -2, D(x_{36})(0) = -\frac{11}{5}, D(x_{37})(0) =$$

$$-\frac{12}{5}, D(x_{38})(0) = -\frac{13}{5}, D(x_{39})(0) = -\frac{14}{5}, D(x_{40})(0) = -3, D(x_{41})(0)$$

$$= -\frac{16}{5}, D(x_{42})(0) = -\frac{17}{5}, D(x_{43})(0) = -\frac{18}{5}, D(x_{44})(0) = -\frac{19}{5},$$

$$D(x_{45})(0) = -4, D(x_{46})(0) = -\frac{21}{5}, D(x_{47})(0) = -\frac{22}{5}, D(x_{48})(0) =$$

$$-\frac{23}{5}, D(x_{49})(0) = -\frac{24}{5}, D(x_{50})(0) = -5, D(y_1)(0) = 0, D(y_2)(0) = 0,$$

$$D(y_3)(0) = 0, D(y_4)(0) = 0, D(y_5)(0) = 0, D(y_6)(0) = 0, D(y_7)(0) = 0,$$

$$D(y_8)(0) = 0, D(y_9)(0) = 0, D(y_{10})(0) = 0, D(y_{11})(0) = 0, D(y_{12})(0)$$

$$= 0, D(y_{13})(0) = 0, D(y_{14})(0) = 0, D(y_{15})(0) = 0, D(y_{16})(0) = 0,$$

$$D(y_{17})(0) = 0, D(y_{18})(0) = 0, D(y_{19})(0) = 0, D(y_{20})(0) = 0, D(y_{21})(0)$$

$$= 0, D(y_{22})(0) = 0, D(y_{23})(0) = 0, D(y_{24})(0) = 0, D(y_{25})(0) = 0,$$

$$D(y_{26})(0) = 0, D(y_{27})(0) = 0, D(y_{28})(0) = 0, D(y_{29})(0) = 0, D(y_{30})(0)$$

$$= 0, D(y_{31})(0) = 0, D(y_{32})(0) = 0, D(y_{33})(0) = 0, D(y_{34})(0) = 0,$$

$$D(y_{35})(0) = 0, D(y_{36})(0) = 0, D(y_{37})(0) = 0, D(y_{38})(0) = 0, D(y_{39})(0)$$

$$= 0, D(y_{40})(0) = 0, D(y_{41})(0) = 0, D(y_{42})(0) = 0, D(y_{43})(0) = 0,$$

$$D(y_{44})(0) = 0, D(y_{45})(0) = 0, D(y_{46})(0) = 0, D(y_{47})(0) = 0, D(y_{48})(0)$$

$$= 0, D(y_{49})(0) = 0, D(y_{50})(0) = 0]$$

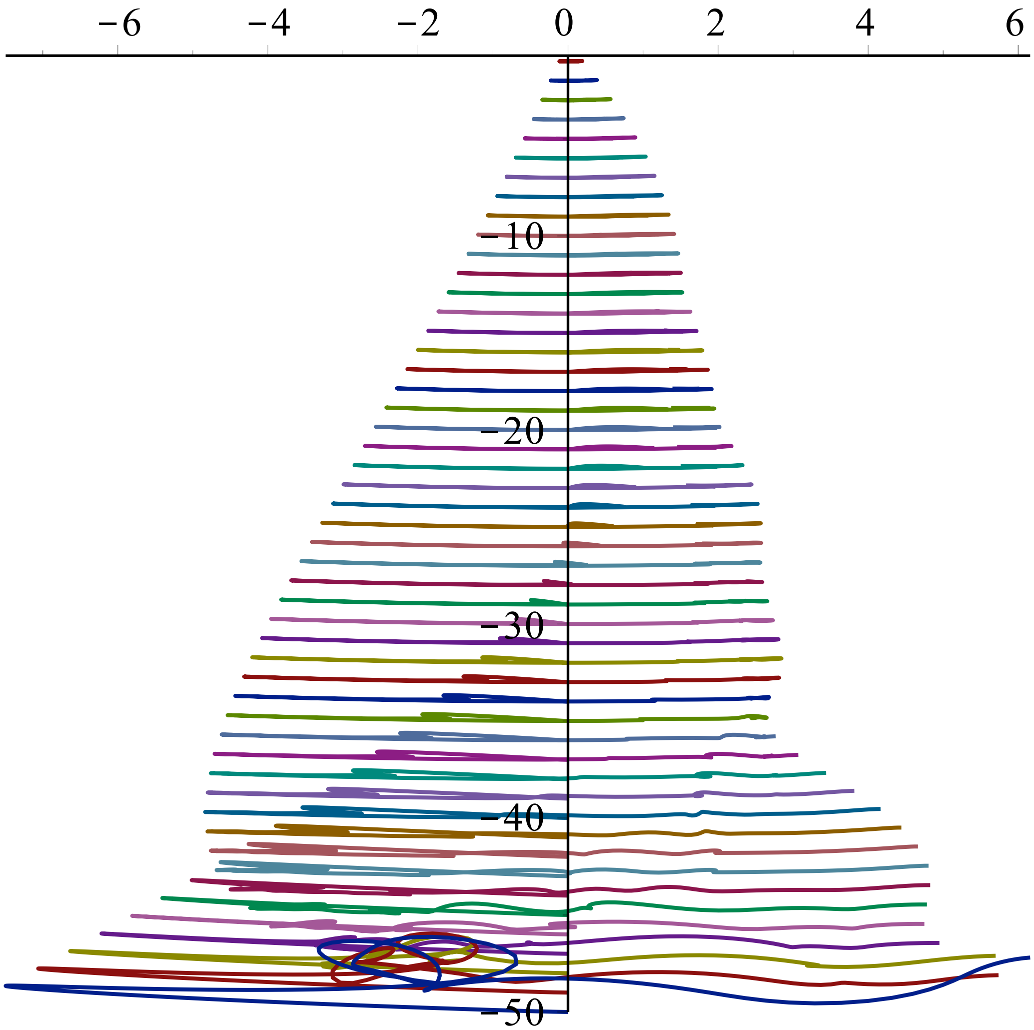
Compile again

```
> tt := time():
  dsn := dsolve({op(dsys), op(ini)}, numeric, implicit = true,
  method = rosenbrock_dae, optimize = true, compile = true,
  maxfun=0):
```

```
`Solution time:`,time()-tt;
```
Solution time:, 0.237 **(1.1.1.8)**

Integrate with these new initial conditions

```
> tt := time():
  plots[odeplot](dsn, all_vars, 0..8);
  `Integration time:`,time()-tt;
```



Integration time:, 9.490 **(1.1.1.9)**

Can a computer really be *more useful* than a good book for finding *exact* ODE and PDE solutions ?

Maple solves - without using lookup tables - 99 % of the 1500+ solvable equations of the famous book by E. Kamke, which summarized - in 1959 - most of the ODE solving algorithms available in the literature.

The performance of these differential equation solvers is under constant enhancement by scientists all around with improvements being added to the system at every release.

The kind of solutions one can compute are more rich than what can be found in a textbook.

Liouvillian versus special function solutions

```
> restart; _on;
```

y(x) will now be displayed as y

derivatives with respect to x of functions of one variable will now be displayed with ' **(1.2.1.1)**

A second order ODE

```
> ode := diff(y(x), x, x) = (beta*x+2*x^2-alpha-1)*(diff(y(x), x))/x+((2*alpha+4)*x+beta*(alpha+1))*y(x)/x;
```

$$ode := y'' = \frac{(\beta x + 2x^2 - \alpha - 1)\, y'}{x} + \frac{((2\alpha + 4)\, x + \beta\, (\alpha + 1))\, y}{x} \qquad \textbf{(1.2.1.2)}$$

Its solution can be expressed in terms of special functions - in this case the biconfluent Heun function (HB)

```
> dsolve(ode, [Heun]);
```

$$y = \_C1\, e^{x\,(x+\beta)} + \_C2\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, x^{-\alpha} \qquad \textbf{(1.2.1.3)}$$

But this ODE also admits Liouvillian solutions, computable with Kovacic's algorithm

```
> dsolve(ode, [Kovacic]);
```

$$y = \_C1\, e^{x\,(x+\beta)} + \_C2\, e^{x\,(x+\beta)} \left(\int x^{-\alpha-1}\, e^{-x\,(x+\beta)}\, dx\right) \qquad \textbf{(1.2.1.4)}$$

The integral cannot be computed by existing integration algorithms ... BUT it should be equal to the solution in terms of HB for some values of the constants _C1 and _C2:

```
> eval(rhs((1.2.1.4)), [_C1=0,_C2=1]) = rhs((1.2.1.3));
```

$$e^{x\,(x+\beta)} \left(\int x^{-\alpha-1}\, e^{-x\,(x+\beta)}\, dx\right) = \_C1\, e^{x\,(x+\beta)} + \_C2\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, x^{-\alpha} \qquad \textbf{(1.2.1.5)}$$

```
> combine(expand(exp(-x*(beta+x)) * (1.2.1.5)));
```

$$\int x^{-\alpha-1}\, e^{-\beta x - x^2}\, dx = \_C2\, x^{-\alpha}\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, e^{-\beta x - x^2} + \_C1 \qquad \textbf{(1.2.1.6)}$$

So _C1 is just an integration constant; for _C2:

```
> diff((1.2.1.6), x);
```

$$x^{-\alpha-1}\, e^{-\beta x - x^2} = -\frac{\_C2\, x^{-\alpha}\, \alpha\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, e^{-\beta x - x^2}}{x} + \_C2\, x^{-\alpha}\, HB'(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, e^{-\beta x - x^2} + \_C2\, x^{-\alpha}\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x)\, (-\beta - 2x)\, e^{-\beta x - x^2} \qquad \textbf{(1.2.1.7)}$$

```
> simplify(expand((lhs-rhs)((1.2.1.7))), size);
```

$$\frac{1}{x^{\alpha}\, x\, e^{\beta x}\, e^{x^2}} \left(2\left(x^2 + \frac{1}{2}\beta x + \frac{1}{2}\alpha\right) \_C2\, HB(-\alpha, \beta, -2 - \alpha, \beta\,(\alpha + 1), x) \qquad \textbf{(1.2.1.8)}\right.$$

$$-\_C2\,HB'\left(-\alpha, \beta, -2-\alpha, \beta\,(\alpha+1), x\right) x+1\Big)$$

```
> series(numer((1.2.1.8)), x, 2);
```

$$\_C2\,\alpha+1+\left(-\frac{\_C2\,\alpha\,\beta}{\alpha-1}+\frac{\_C2\,\beta}{\alpha-1}+\_C2\,\beta\right)x+\mathrm{O}\left(x^2\right) \tag{1.2.1.9}$$

```
> coeffs(convert((1.2.1.9), polynom), x) =~ [0,0];
```

$$\left[\_C2\,\alpha+1=0,\ -\frac{\_C2\,\alpha\,\beta}{\alpha-1}+\frac{\_C2\,\beta}{\alpha-1}+\_C2\,\beta=0\right] \tag{1.2.1.10}$$

```
> solve((1.2.1.10), _C2);
```

$$\left\{\_C2=-\frac{1}{\alpha}\right\} \tag{1.2.1.11}$$

So, a new integration formula appears

```
> subs((1.2.1.11), (1.2.1.6));
```

$$\int x^{-\alpha-1}\,\mathrm{e}^{-\beta x-x^2}\,\mathrm{d}x=-\frac{x^{-\alpha}\,HB\left(-\alpha, \beta, -2-\alpha, \beta\,(\alpha+1), x\right)\mathrm{e}^{-\beta x-x^2}}{\alpha} \tag{1.2.1.12}$$

$$+\_C1$$

Note for instance that in the literature it is said that integral forms for Heun functions are not know.

Singular solutions depending on the values of the parameters

Compute the point symmetries of an ordinary differential equation (ODE), that is, solve the determining PDE system for the infinitesimals of the symmetry generator. Consider example 11 from Kamke's book:

Consider this ODE example from Kamke's book, equation 11 of the nonlinear sector

```
> ode[11] := diff(y(x),x,x)+a*x^r*y(x)^n = 0;
```

$$ode_{11} := y''+a\,x^r\,y^n=0 \tag{1.2.2.1}$$

In the computer we can directly study its symmetries. The PDE system satisfied by these symmetries, that is, infinitesimals $[\xi, \eta]$ of the symmetry generator, is given by

```
> declare( (xi,eta)(x,y) );
```

$\xi(x, y)$ *will now be displayed as* ξ

$\eta(x, y)$ *will now be displayed as* η (1.2.2.2)

```
> sys := [gensys( ode[11], [xi,eta](x,y))]:
> for _eq in sys do _eq=0 end do;
```

$$\xi_{y,y}=0$$

$$\eta_{y,y}-2\,\xi_{x,y}=0$$

$$3\,x^r\,y^n\,\xi_y\,a-\xi_{x,x}+2\,\eta_{x,y}=0$$

(1.2.2.3)

$$2\,\xi_x\,x^r\,y^n\,a - x^r\,y^n\,\eta_y\,a + \frac{\eta\,a\,x^r\,y^n\,n}{y} + \frac{\xi\,a\,x^r\,r\,y^n}{x} + \eta_{x,x} = 0 \tag{1.2.2.3}$$

This is a second order linear PDE system, with two unknowns $\{\eta(x, y), \xi(x, y)\}$ and four equations. Its *general solution* is given by

```
> sol := pdsolve(sys);
```

$$sol := \left\{\eta = -\frac{\_C1\,y\,(r+2)}{n-1}, \xi = \_C1\,x\right\} \tag{1.2.2.4}$$

Solutions to PDE systems can be tested by using pdetest (the command tests whether the solution satisfies each PDE in the system).

```
> pdetest(sol, sys);
```

$$[0, 0, 0, 0] \tag{1.2.2.5}$$

Since symmetries are defined up to a multiplicative constant, we can drop $\_C1$. Hence, for arbitrary $\{a, n, r\}$, the ODE has only one point symmetry, and with it we can perform only a reduction in order by one:

```
> reduce_order(ode[11], [xi = x, eta = -y*(r+2)/(n-1)]):
  simplify(%, size);
```

$$y = \left(\_a\,\mathrm{e}^{-\frac{\left(\int \_b(\_a)\,\mathrm{d}\_a + \_C1\right)(r+2)}{n-1}}\right) \&\text{where} \left[\left\{\_b_{\_a} = \frac{1}{(n-1)^2}\left(\_b(\_a)^2\left(a\,\_b(\_a)\,(n-1)^2\,\_a^n + \_a\,(r+2)\,(r+1+n)\,\_b(\_a) - (n-1)\,(n+2\,r+3)\right)\right)\right\}, \left\{\_a = y\,x^{\frac{r+2}{n-1}}, \_b(\_a) = \frac{x^{\frac{-r-2}{n-1}}\,(n-1)}{x\,(n-1)\,y' + y\,(r+2)}\right\}, \left\{x = \mathrm{e}^{\int \_b(\_a)\,\mathrm{d}\_a + \_C1}, y = \_a\,\mathrm{e}^{-\frac{\left(\int \_b(\_a)\,\mathrm{d}\_a + \_C1\right)(r+2)}{n-1}}\right\}\right] \tag{1.2.2.6}$$

Now, an interesting question is: Are there other solutions related to *particular values of the parameters n and r*?

To answer that question one must solve a more difficult, now nonlinear, PDE system for ξ and η, taking n and r are *parameters*. Also, in this particular problem, from the form of the ODE $y'' + a\,x^r\,y^n = 0$, the case $n = 1$ is of no interest since the ODE would become linear.

We are interested in a solution for different values of n and r but also for $n \neq 1$. The first step is then to add this inequation to the PDE system.

```
> sys1 := [op(sys), n<>1];
```

(1.2.2.7)

$$sys1 := \left[\xi_{y,y}, \eta_{y,y} - 2\,\xi_{x,y}, 3\,x^r\,y^n\,\xi_y\,a - \xi_{x,x} + 2\,\eta_{x,y}, 2\,\xi_x\,x^r\,y^n\,a - x^r\,y^n\,\eta_y\,a + \frac{\eta\,a\,x^r\,y^n\,n}{y} + \frac{\xi\,a\,x^r\,r\,y^n}{x} + \eta_{x,x}, n \neq 1\right] \qquad \textbf{(1.2.2.7)}$$

Next we indicate to **pdsolve** that n and r are parameters of the problem, thus arriving at the desired solutions.

```
> sol1 := pdsolve(sys1, parameters = {n, r});
```

$$sol1 := \{n = 2, r = -5, \eta = y\,(\_C2\,x + 3\,\_C1), \xi = x\,(\_C2\,x + \_C1)\}, \left\{n = 2, r = -\frac{20}{7}, \eta = -\frac{2\,(-6\,\_C2\,x^2 - 98\,x^{8/7}\,\_C2\,a\,y - 147\,\_C1\,a\,x\,y)}{343\,x\,a}, \xi = \_C1\,x + \_C2\,x^{8/7}\right\}, \left\{n = 2, r = -\frac{15}{7}, \eta = -\frac{-49\,\_C1\,a\,x\,y - 147\,x^{6/7}\,\_C2\,a\,y + 12\,\_C2\,x}{343\,x\,a}, \xi = \_C1\,x + \_C2\,x^{6/7}\right\}, \{n = 2, r = r, \eta = -\_C1\,y\,(r+2), \xi = \_C1\,x\}, \left\{n = -r - 3, r = r, \eta = \frac{(4\,\_C2\,x + 2\,\_C1 + (\_C2\,x + \_C1)\,r)\,y}{r+4}, \xi = x\,(\_C2\,x + \_C1)\right\}, \left\{n = n, r = r, \eta = -\frac{\_C1\,y\,(r+2)}{n-1}, \xi = \_C1\,x\right\} \qquad \textbf{(1.2.2.8)}$$

```
> map(pdetest, [sol1], sys1);
```

$$[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]] \qquad \textbf{(1.2.2.9)}$$

So there exist particular values of n and r for which the system has additional solutions, and the presence of two arbitrary constants ensures that the ode[11] is solvable to the end, not just a reduction of order. For instance:

```
> subs(n = 2, r = -5, ode[11]);
```

$$y'' + \frac{a\,y^2}{x^5} = 0 \qquad \textbf{(1.2.2.10)}$$

```
> dsolve((1.2.2.10));
```

$$y = -\frac{6\,\wp\left(-\frac{1}{x} + \_C1; 0, \_C2\right)\,x}{a} \qquad \textbf{(1.2.2.11)}$$

```
> InputForm((1.2.2.11));
y(x) = -6*WeierstrassP(-1/x+_C1, 0, _C2)*x/a
```

Aren't these computer algebra environments more like a *black-box approach* to the problem ?

There are help-pages with examples for every solving algorithm used, and while solving dsolve can provide user information on what is doing.

userinfos - what happens behind the scene

```
> infolevel[dsolve] := 5;
```

$$infolevel_{dsolve} := 5 \quad (1.3.1.1)$$

Example 9 from the set of linear ODEs of Kamke's book, in terms of a parameter n

```
> ode[9] := diff(y(x),x,x) = (11/4 - x^(2*n) - 2*x^n)*n^2/
  (x^n+1)^2*x^(2*n-2)*y(x) + (n-1)/x*diff(y(x),x);
```

$$ode_9 := y'' = \frac{\left(\frac{11}{4} - x^{2n} - 2x^n\right) n^2 x^{2n-2} y}{\left(x^n + 1\right)^2} + \frac{(n-1)\,y'}{x} \quad (1.3.1.2)$$

```
> dsolve(ode[9]);
Methods for second order ODEs:
--- Trying classification methods ---
trying a symmetry of the form [xi=0, eta=F(x)]
   testing BRANCH 1 ->
   testing BRANCH 2 ->
   testing BRANCH 3 ->
   testing BRANCH 4 ->
   testing BRANCH 5 ->
   testing BRANCH 6 ->
checking if the LODE is missing 'y'
-> Trying an equivalence, under non-integer power transformations,
   to LODEs admitting Liouvillian solutions.
   -> Trying a Liouvillian solution using Kovacic's algorithm
   <- No Liouvillian solutions exists
-> Trying a solution in terms of special functions:
   -> Bessel
   -> elliptic
   -> Legendre
   -> Whittaker
      -> hyper3: Equivalence to 1F1 under 'a power @ Moebius'
      <- hyper3 successful: received ODE is equivalent to the 1F1
ODE
   <- Whittaker successful
<- special function solution successful
```

$$y = \_C1\, M_{0,2}\left(2\,\mathrm{I}\left(x^n + 1\right)\right) + \_C2\, W_{0,2}\left(2\,\mathrm{I}\left(x^n + 1\right)\right) \quad (1.3.1.3)$$

```
> InputForm((1.3.1.3));
y(x) = _C1*WhittakerM(0, 2, (2*I)*(x^n+1))+_C2*WhittakerW(0, 2, (2*I)*
(x^n+1))
```

But Whittaker functions belong the the 1F1 hypergeometric class, so this equation can be solved in terms of hypergeometric functions as well

```
> dsolve(ode[9], [hypergeometric]);
Classification methods on request
Methods to be used are: [hypergeometric]
----------------------------
* Tackling ODE using method: hypergeometric
--- Trying classification methods ---
trying a solution in terms of hypergeometric functions
   -> heuristic approach
   -> hyper3: Equivalence to 2F1, 1F1 or 0F1 under 'a power @
Moebius'
   <- hyper3 successful: received ODE is equivalent to the 1F1 ODE
```

```
   pFq equation: _a*y(x)+(x-_c)*diff(y(x),x)-x*diff(y(x),x,x) = 0
Transformation resolving the equivalence: {x = 2*I*(t^n+1), y(x) =
1/(t^n+1)^(5/2)/exp(-I*(t^n+1))*u(t)}
Values of hypergeometric parameters: [_a = 5/2], [_c = 5]
<- hypergeometric solution successful
```

$$y = \_C1\,(x^n+1)^{5/2}\,e^{-I(x^n+1)}\,{}_1F_1\left(\frac{5}{2}; 5; 2I(x^n+1)\right) + \_C2\,(x^n+1)^{5/2}\,e^{-I(x^n+1)}\,U\left(\frac{5}{2}, 5, 2I(x^n+1)\right) \quad \textbf{(1.3.1.4)}$$

By the way, 1F1 functions where the first parameter is a half integer, typically can be expressed in terms of Bessel functions:

```
> combine(convert((1.3.1.4), Bessel));
```

$$y = 8\,\_C1\,\sqrt{x^n+1}\,J_2(x^n+1)\,e^{-I(x^n+1)+I+I x^n} - \frac{\_C2\sqrt{\frac{x^n+1}{\pi}}\,K_2(I(x^n+1))}{4} \quad \textbf{(1.3.1.5)}$$

```
> odetest((1.3.1.5), ode[9]);
```

$$0 \quad \textbf{(1.3.1.6)}$$

```
> infolevel[dsolve] := 1;
```

$$infolevel_{dsolve} := 1 \quad \textbf{(1.3.1.7)}$$

Can we *really* study *the* "differential equations" behind a problem using Computer Algebra as we would do by hand?

The two packages behind dsolve and pdsolve, DEtools and PDEtools, are actually computational environments for ***studying*** differential equations step by step.

"Given an ODE we can formulate the finding of its symmetries; ***BUT****, given* ***the symmetries****, could we formulate the finding of the given invariant ODE family ?"*

"Given an ODE we can often find its symmetries and from there its exact solution, ***BUT****, given* ***the solution****, could we formulate the finding of the symmetries of the unknown underlying ODE problem ?"*

Equations invariant under symmetry groups (equinv) and symmetries behind equation solutions

1) A pair of Lie symmetry infinitesimals

```
> [_xi = 0, _eta = exp(y)/ln(x)];
```

$$\left[\_\xi = 0, \_\eta = \frac{e^y}{\ln(x)}\right] \quad \textbf{(1.4.1.1)}$$

The most general first order ODE invariant under the above is:

```
> equinv((1.4.1.1), y(x));
```

$$y' = \frac{1}{\ln(x)\, x} + e^{y}\, \_F1(x) \tag{1.4.1.2}$$

This ODE is actually solved using Lie symmetry methods

```
> dsolve((1.4.1.2));
```

$$y = -\ln\left(-\frac{\int \_F1(x)\ln(x)\,dx + \_C1}{\ln(x)}\right) \tag{1.4.1.3}$$

The infinitesimals can be reobtained from this solution

```
> buildsym((1.4.1.3), y(x));
```

$$\left[\_\xi = 0,\ \_\eta = \frac{e^{y}}{\ln(x)}\right] \tag{1.4.1.4}$$

2) Three pairs of infinitesimals not containing a 2D subalgebra

```
> symmetries := [[xi = 1, eta = 1], [xi = x, eta = y], [xi =
  x^2, eta = y^2]];
```

$$symmetries := [[\xi = 1, \eta = 1], [\xi = x, \eta = y], [\xi = x^2, \eta = y^2]] \tag{1.4.1.5}$$

The most general second order ODE *simultaneously invariant under these three point symmetries*

```
> equinv(symmetries, y(x), 2);
```

$$y'' = -\frac{y'^{3/2}\, \_C1 + 2\, y'^2 + 2\, y'}{x - y} \tag{1.4.1.6}$$

The third order ODE family admitting these symmetries

```
> equinv(symmetries, y(x), 3);
```

$$y''' = \frac{1}{(x-y)^2}\left(y'^2\, \_F1\left(-\frac{2\, y'^2 + x\, y'' - y\, y'' + 2\, y'}{y'^{3/2}}\right) - 6\, y'^3 - 6\, y'\, y''\, x + 6\, y'\, y''\, y - 24\, y'^2 - 6\, x\, y'' + 6\, y\, y'' - 6\, y'\right) \tag{1.4.1.7}$$

The fact that these ODE families are invariant under the three pairs of infinitesimals above can be tested using a related symtest command

```
> map(symtest, symmetries, (1.4.1.7));
```

$$[0, 0, 0] \tag{1.4.1.8}$$

3) Two pairs of dynamical symmetries

```
> dynamical_symmmetries := [[xi = _y1, eta = 0], [xi = 0, eta =
  1/_y1]];
```

$$dynamical\_symmmetries := \left[[\xi = y', \eta = 0], \left[\xi = 0, \eta = \frac{1}{y'}\right]\right] \tag{1.4.1.9}$$

The most general second order ODE simultaneously invariant under these two dynamical symmetries

```
> equinv(dynamical_symmmetries, y(x), 2);
```

(1.4.1.10)

$$y'' = \frac{y'^2}{\_F1(y')\, y'^2 + 2\, y'\, x - 2\, y} \quad \textbf{(1.4.1.10)}$$

The 4th order equation

```
> equinv(dynamical_symmmetries, y(x), 4);
```

$$y'''' = \frac{1}{y^4\, y''} \left(-8\, y'\, y''^5\, x + \_F1\left(y', \frac{-2\, y'\, y''\, x + y'^2 + 2\, y\, y''}{y''\, y'^2}, \frac{-2\, y'\, x\, y''^3 + y'''\, y^3 + 4\, y\, y''^3}{y''^3} \right) y''^5 + 3\, y^4\, y'''^2 + 16\, y''^5\, y \right) \quad \textbf{(1.4.1.11)}$$

```
> map(symtest, dynamical_symmmetries, (1.4.1.11));
```

$$[0, 0] \quad \textbf{(1.4.1.12)}$$

"Given an ODE we can formulate the finding of its integrating factors, ***BUT****, given* ***these integrating factors****, could we formulate the finding of the associated reducible ODE family ?"*

Equations reducible by given integrating factors (redode)

We want to set up an algorithm such that, given a second order linear ODE,

```
> ode__G := diff(y(x),x,x) = G[1](x)*diff(y(x),x) + G[2](x)*y
  (x) + G[3](x);
```

$$ode_G := y'' = G_1(x)\, y' + G_2(x)\, y + G_3(x) \quad \textbf{(1.4.2.1)}$$

```
> declare(G(x));
```

$$G(x) \textit{ will now be displayed as } G \quad \textbf{(1.4.2.2)}$$

where there are no restrictions on $G_1(x)$, $G_2(x)$, $G_3(x)$, the algorithm determines if the ODE admits an integrating factor $\mu = F(x)$ that reduces the equation to a 1st order ODE that admits the same integrating factor, and if so it also determines this integrating factor $F(x)$.

To start with, we obtain the general first order ODE admitting $\mu = F(x)$ as integrating factor:

```
> declare(F(x));
```

$$F(x) \textit{ will now be displayed as } F \quad \textbf{(1.4.2.3)}$$

```
> ode__1 := redode(F(x), y(x));
```

$$ode_1 := y' = -\frac{F'\, y + \_F1(x)}{F} \quad \textbf{(1.4.2.4)}$$

where $\_F1(x)$ is an arbitrary function. To obtain the second order ODE mentioned above, we pass ode_1 as the last argument, the expected reduced ODE to obtain the form of the second order ODE:

```
> ode__2 := redode(F(x), y(x), ode_1);
```

(1.4.2.5)

$$ode_2 := y' = -\frac{F' y + \_F1(x)}{F} \quad (1.4.2.5)$$

Taking this general ODE pattern as our starting point, we set up the required solving scheme by comparing coefficients in ode_2 and ode_G and running a differential elimination process

```
> collect(ode__G - ode__2, [diff(y(x),x), y(x)]);
```

$$y'' - y' = G_1 y' + \left(G_2 + \frac{F'}{F}\right) y + G_3 + \frac{\_F1(x)}{F} \quad (1.4.2.6)$$

```
> [dcoeffs(rhs((1.4.2.6)), y(x))] =~ [0, 0, 0];
```

$$\left[G_1 = 0, G_2 + \frac{F'}{F} = 0, G_3 + \frac{\_F1(x)}{F} = 0\right] \quad (1.4.2.7)$$

The differential elimination process triangularizes the system in any desired way. In this case we want F as a function of G plus conditions on the G

```
> ranking := [F, [G[1], G[2], G[3]]];
```

$$ranking := [F, [G_1, G_2, G_3]] \quad (1.4.2.8)$$

```
> casesplit((1.4.2.7), ranking);
```

$$\left[F = -\frac{\_F1(x)}{G_3}, G_1 = 0, G_3' = \frac{\_F1(x)\, G_2\, G_3 + \_F1'\, G_3}{\_F1(x)}\right] \&\text{where } [\_F1(x) \neq 0, G_3 \neq 0], [F' = -G_2 F, G_1 = 0, G_3 = 0, \_F1(x) = 0] \&\text{where } [F \neq 0] \quad (1.4.2.9)$$

Is there something fundamentally relevant regarding ODEs and PDEs that we can *only do* with a computer?

Differential elimination permits computing a) singular cases, b) triangularizging DE systems and c) reducing PDE systems taking into account integrability conditions.

Equivalent to computing a differential Groebner basis, the amount of intermediate computations it involves makes it just not practical by hand.

1) The singular cases of an ODE

```
> ode__1 := (diff(y(x), x, x))^2+2*(diff(y(x), x, x))*y(x)^3*
  (diff(y(x), x))-4*y(x)^2*(diff(y(x), x))^3 = 0;
```

$$ode_1 := y''^2 + 2 y'' y^3 y' - 4 y^2 y'^3 = 0 \quad (1.5.1)$$

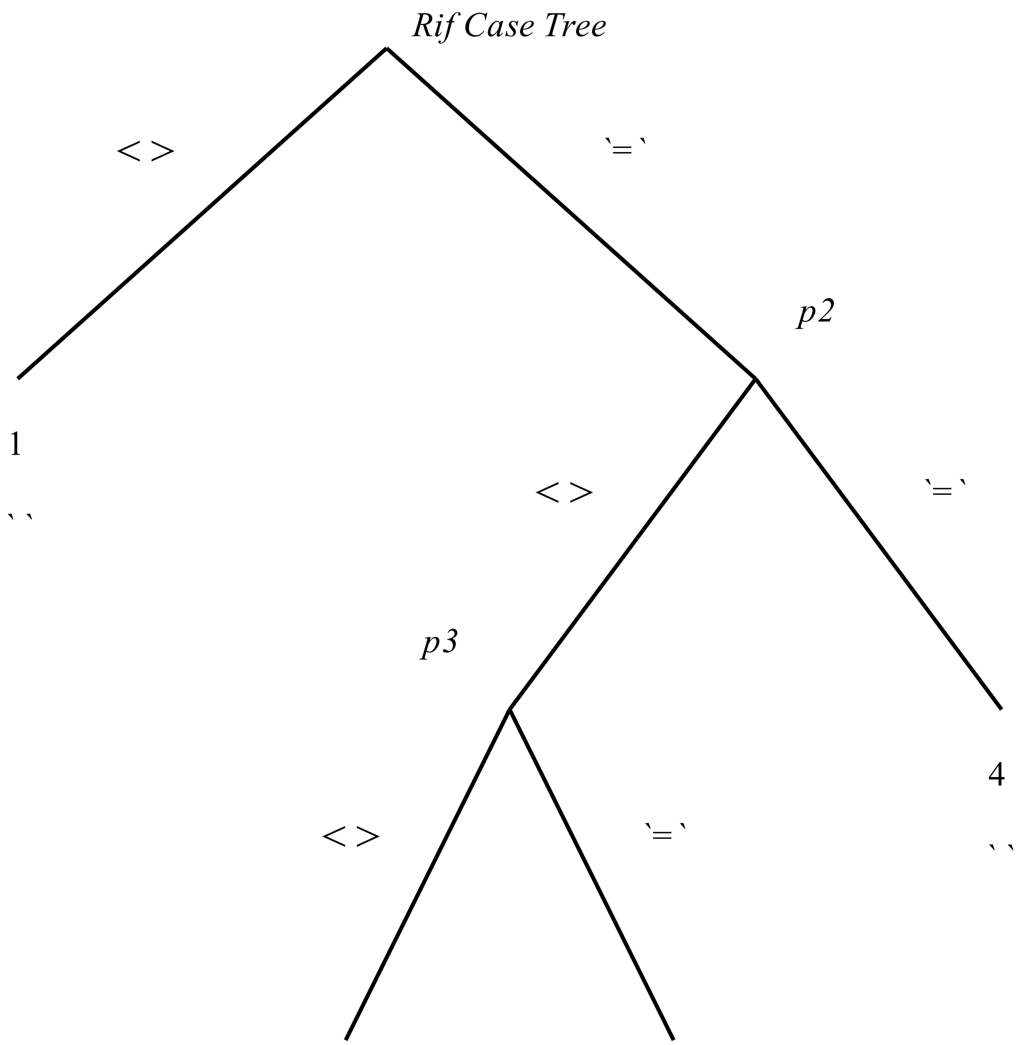
```
> casesplit(ode__1, 'caseplot');
```

========= *Pivots Legend* =========

$$p1 = y^3 y' + y''$$

$$p2 = y$$

$$p3 = y'$$



$$\left[y'^2 = -2\, y''\, y^3\, y' + 4\, y^2\, y'^3\right] \&\text{where } \left[y^3\, y' + y'' \neq 0\right], \left[y' = -\frac{y^4}{4}\right] \&\text{where } \left[y \neq 0\right], \left[y' = 0\right] \&\text{where } \left[y \neq 0\right], \left[y = 0\right] \&\text{where } [\ ] \quad \textbf{(1.5.2)}$$

2) Simplifying PDE systems with respect to their integrability conditions. Depart from an ODE

```
> ode__2 := diff(y(x),x,x) - 3*y(x)*diff(y(x),x) - 3*a*y(x)^2 -
  4*a^2*y(x) - b = 0;
```

$$ode_2 := y'' - 3\, y\, y' - 3\, a\, y^2 - 4\, a^2\, y - b = 0 \quad \textbf{(1.5.3)}$$

Build the PDE system satisfied by its symmetry infinitesimals ξ and η

```
> sys_xi_eta := [gensys(ode__2, [xi(x,y), eta(x,y)])]:
  for _eq in % do _eq = 0; od;
```

$$\xi_{y,y} = 0$$

$$-6\, \xi_y\, y + \eta_{y,y} - 2\, \xi_{x,y} = 0$$

$$-12\, \xi_y\, a^2\, y - 9\, \xi_y\, a\, y^2 - 3\, \xi_x\, y - 3\, \xi_y\, b - 3\, \eta + 2\, \eta_{x,y} - \xi_{x,x} = 0$$

$$-8\, \xi_x\, a^2\, y - 6\, \xi_x\, a\, y^2 + 4\, \eta_y\, a^2\, y + 3\, \eta_y\, a\, y^2 - 4\, \eta\, a^2 - 6\, \eta\, a\, y - 2\, \xi_x\, b - 3\, \eta_x\, y \quad \textbf{(1.5.4)}$$

$$+ \eta_y b + \eta_{x,x} = 0$$

This system of 4 PDEs simplify in a significant way if you take into accound integrability conditions

```
> casesplit(sys_xi_eta);
```

$$\left[\eta = 0, \xi_x = 0, \xi_y = 0\right] \&\text{where } [\ ] \tag{1.5.5}$$

From where it's solution follows trivially

```
> pdsolve((1.5.5));
```

$$\{\eta = 0, \xi = \_C1\} \tag{1.5.6}$$

Special Functions

"Special functions, their inter-relation and representations become alive within a computer"

Conversions between mathematical functions

The type of conversion we are considering

Example:

```
> restart;
> hypergeom([a],[1/2],1/2*z^2);
```

$${}_1F_1\left(a; \frac{1}{2}; \frac{z^2}{2}\right) \tag{2.1.1}$$

```
> convert(%, CylinderU);
```

$$\frac{e^{\frac{z^2}{4}} \Gamma\left(a + \frac{1}{2}\right) \left(U\left(2\,a - \frac{1}{2}, z\right) + U\left(2\,a - \frac{1}{2}, -z\right)\right) 2^a}{2\sqrt{\pi}} \tag{2.1.2}$$

```
> E := Ei(a, z);
```

$$E := \mathrm{Ei}_a(z) \tag{2.1.3}$$

```
> E = convert(E, GAMMA);
```

$$\mathrm{Ei}_a(z) = z^{a-1}\,\Gamma(1 - a, z) \tag{2.1.4}$$

```
> E = convert(E, KummerU);
```

$$\mathrm{Ei}_a(z) = \frac{U(1, 2 - a, z)}{e^z} \tag{2.1.5}$$

```
> E = convert(E, hypergeom);
```

(2.1.6)

$$\mathrm{Ei}_a(z) = \frac{{}_1F_1(1-a;2-a;-z)}{a-1} + z^{a-1}\,\Gamma(1-a) \qquad \textbf{(2.1.6)}$$

```
> E = convert(E, hypergeom) assuming a = 1;
```

$$\mathrm{Ei}_a(z) = -\frac{z^{a-1}\left(z\,{}_2F_2(1,1;2,1+a;-z) - a\left(-\Psi(a)+\ln(z)\right)\right)}{\Gamma(1+a)\,(-1)^a} \qquad \textbf{(2.1.7)}$$

```
> E = convert(E, hypergeom) assuming a::posint;
```

$$\mathrm{Ei}_a(z) = \left(-\frac{z\,{}_2F_2(1,1;2,1+a;-z)}{\Gamma(1+a)\,(-1)^a} + \frac{\sum_{\_k1=0}^{-2+a} -\frac{(-1)^{-\_k1}\,z^{-\_k1}}{(1-a+\_k1)\,\Gamma(\_k1+1)}}{z^{a-1}} + \frac{a\left(-\Psi(a)+\ln(z)\right)}{\Gamma(1+a)\,(-1)^a}\right) z^{a-1} \qquad \textbf{(2.1.8)}$$

Contiguity relations are implemented as "rule conversions"

The rules currently implemented:

"raise a",	"lower a",	"normalize a",
"raise b",	"lower b",	"normalize b",
"raise c",	"lower c",	"normalize c",
"mix a and b",	"1F1 to 0F1",	"0F1 to 1F1"
"quadratic 1",	"quadratic 2",	"quadratic 3"
"quadratic 4",	"quadratic 5",	"quadratic 6"
"2a2b",	"raise 1/2",	"lower 1/2"

Example

```
> hypergeom([a],[b],z);
```

$${}_1F_1(a;b;z) \qquad \textbf{(2.1.9)}$$

```
> % = convert(%, hypergeom, "mix a and b");
```

$${}_1F_1(a;b;z) = \mathrm{e}^z\,{}_1F_1(b-a;b;-z) \qquad \textbf{(2.1.10)}$$

```
> %% = convert(%%, hypergeom, "raise b");
```

$${}_1F_1(a;b;z) = \frac{(b+z)\,{}_1F_1(a;1+b;z) + \frac{z\,(a-1-b)\,{}_1F_1(a;b+2;z)}{1+b}}{b} \qquad \textbf{(2.1.11)}$$

```
> hypergeom([a, b],[-1/2+a+b],z);
```

$$ {}_2F_1\left(a, b; -\frac{1}{2}+a+b; z\right) \tag{2.1.12}$$

```
> % = convert(%,hypergeom,"quadratic 1");
```

$$ {}_2F_1\left(a, b; -\frac{1}{2}+a+b; z\right) = \frac{{}_2F_1\left(-1+2a, -1+2b; -\frac{1}{2}+a+b; \frac{1}{2}-\frac{\sqrt{1-z}}{2}\right)}{\sqrt{1-z}} \tag{2.1.13}$$

```
> %% = convert(%%,hypergeom,"quadratic 2");
```

$$ {}_2F_1\left(a, b; -\frac{1}{2}+a+b; z\right) = \frac{{}_2F_1\left(-1+2a, \frac{1}{2}+a-b; -\frac{1}{2}+a+b; \frac{-1+\sqrt{1-z}}{1+\sqrt{1-z}}\right) 2^{2a}}{2\left(1+\sqrt{1-z}\right)^{-1+2a}\sqrt{1-z}} \tag{2.1.14}$$

Class conversions

- The network also implements **conversions to the function classes**:

\`0F1\`	\`1F1\`	\`2F1\`	Airy
arctrig	arctrigh	Bessel_related	Chebyshev
Cylinder	Ei_related	elementary	Elliptic_related
erf_related	GAMMA_related	Hankel	Heun
Kelvin	Kummer	Legendre	trig
trigh	Whittaker		

Example

Conversions to the *elementary* class

```
> hypergeom([1,2],[3/2],z) + KummerU(-2,-2,-9*z^(1/6));
```

$$ {}_2F_1\left(1, 2; \frac{3}{2}; z\right) + U\left(-2, -2, -9 z^{1/6}\right) \tag{2.1.15}$$

```
> convert(%, elementary);
```

$$\left(2-18\,z^{1/6}+81\,z^{1/3}\right)\mathrm{e}^{9\,z^{1/6}}\,\mathrm{e}^{-9\,z^{1/6}}+\frac{\sqrt{1-z}\,\arcsin\left(\sqrt{z}\right)}{2\,(z-1)^{2}\sqrt{z}}-\frac{1}{2\,(z-1)} \qquad \textbf{(2.1.16)}$$

Conversions between subclasses of the 2F1 hypegeometric class

```
> LegendreP(1/2, z);
```

$$P_{\frac{1}{2}}(z) \qquad \textbf{(2.1.17)}$$

```
> convert(%, Elliptic_related );
```

$$\frac{2\sqrt{2}\,\sqrt{z+1}\,\boldsymbol{E}\left(\sqrt{\frac{z-1}{z+1}}\right)}{\pi}-\frac{2\sqrt{2}\,\boldsymbol{K}\left(\sqrt{\frac{z-1}{z+1}}\right)}{\pi\sqrt{z+1}} \qquad \textbf{(2.1.18)}$$

The plotcompare facility

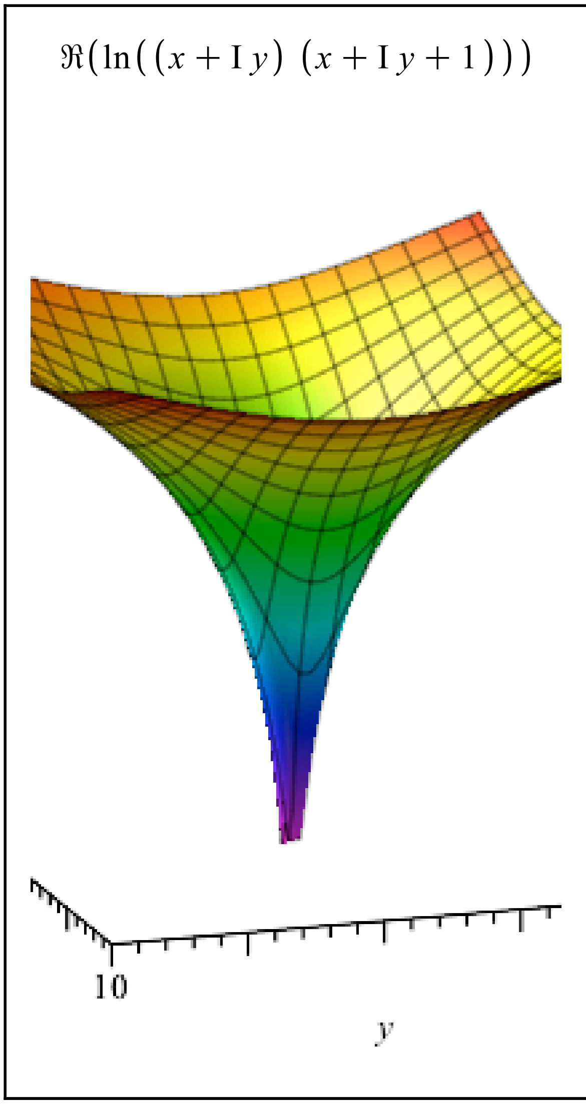
For educational purposes a related facility for "visually comparing" mathematical expressions over the complex plane was prepared

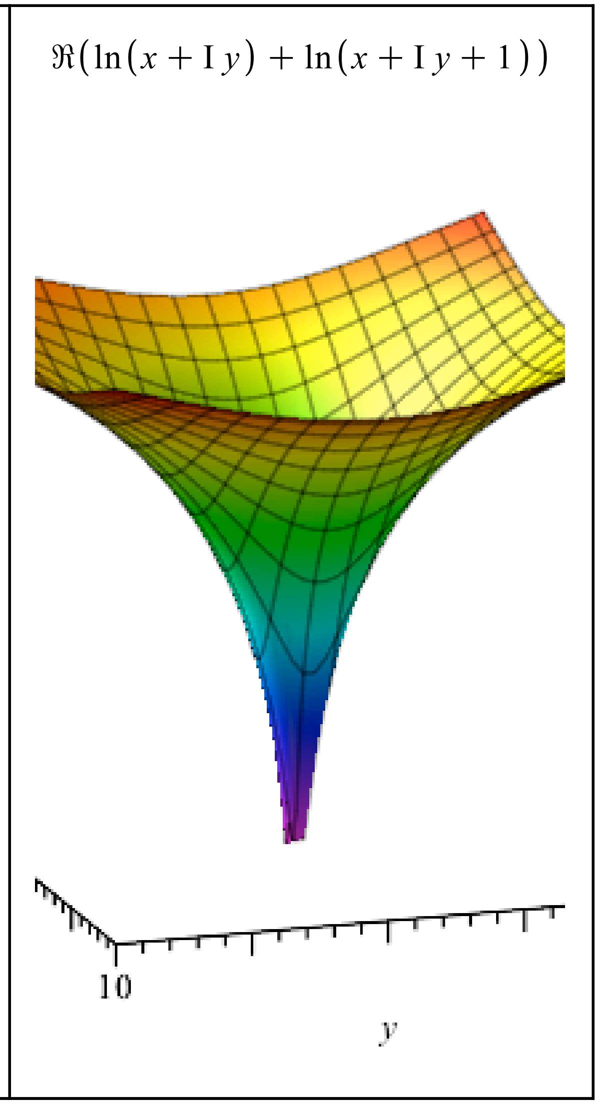
Example

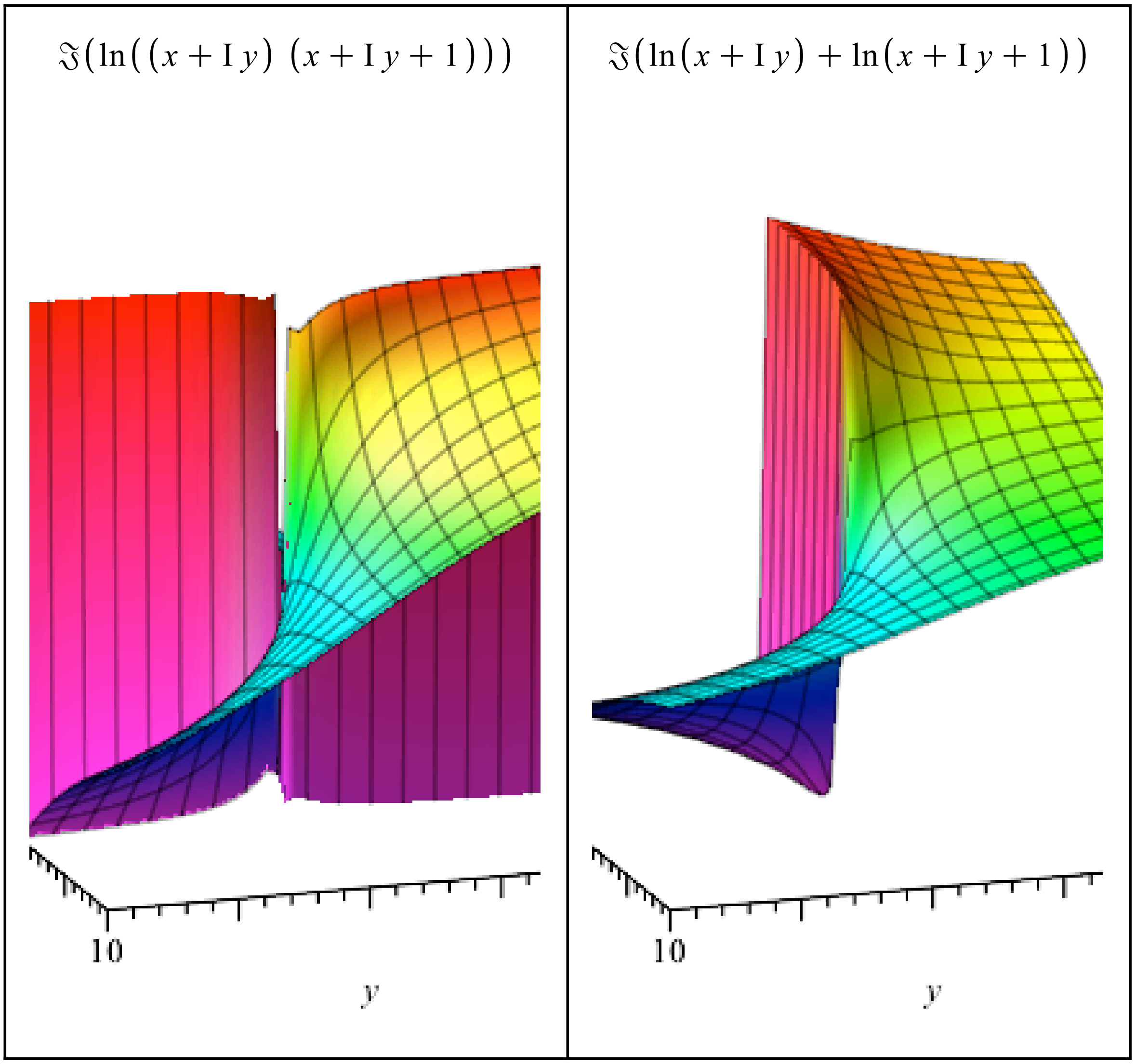
```
> eq := ln(z*(1+z)) = ln(z)+ln(1+z);
```

$$eq := \ln(z\,(z+1)) = \ln(z)+\ln(z+1) \qquad \textbf{(2.1.19)}$$

```
> plots[plotcompare](eq, scale_range = 10, 5);
```

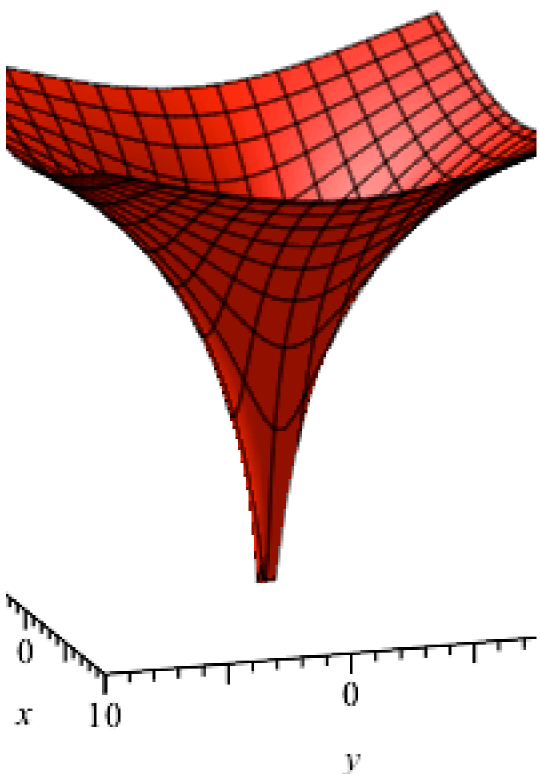

ℜ(ln((x + I y) (x + I y + 1)))
10
y


ℜ(ln(x + I y) + ln(x + I y + 1))
10
y

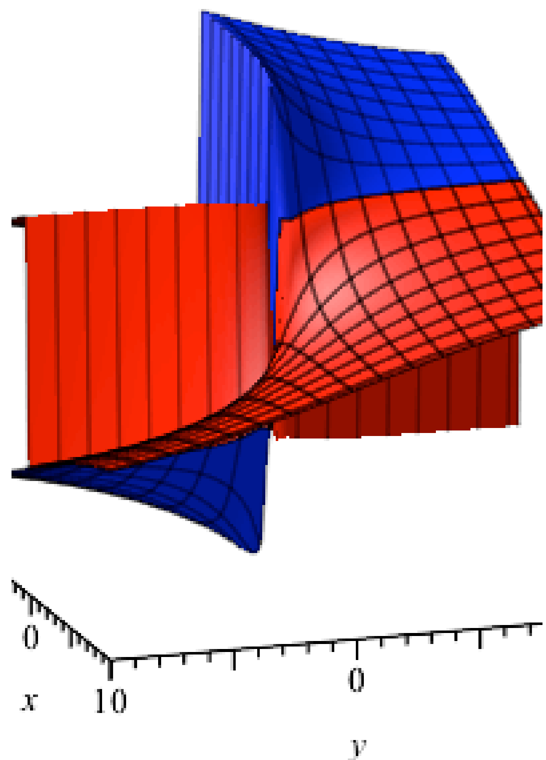



```
> plots[plotcompare](eq, scale_range = 10, 5, same_box);
```

$\Re(\ln((x+\mathrm{I}\,y)\,(x+\mathrm{I}\,y+1)))$ **and** $\Re(\ln(x+\mathrm{I}\,y)+\ln(x+\mathrm{I}\,y+1))$



$\Im(\ln((x+\mathrm{I}\,y)\,(x+\mathrm{I}\,y+1)))$ **and** $\Im(\ln(x+\mathrm{I}\,y)+\ln(x+\mathrm{I}\,y+1))$



The FunctionAdvisor project

"The requirement concerning math functions, is not just computational: typically one also needs information on identities, simplifications or integral forms valid for particular cases, series; mathematical properties in general. We usually look for that information in handbooks like Abramowitz & Stegun.*"*

The FunctionAdvisor provides information by processing on the fly a large number of mathematical information using a (growing) number of mathematical algorithms.

"I don't know where to start"

```
> FunctionAdvisor();
The usage is as follows:
```

```
    > FunctionAdvisor( topic, function, ... );
where 'topic' indicates the subject on which advice is required,
'function' is the name of a Maple function, and '...' represents possible
additional input depending on the 'topic' chosen. To list the possible
topics:
    > FunctionAdvisor( topics );
A short form usage,
    > FunctionAdvisor( function );
with just the name of the function is also available and displays a
summary of information about the function.
```

Following the information above,

```
> FunctionAdvisor( topics );
The topics on which information is available are:
```

$$[DE, analytic\_extension, asymptotic\_expansion, branch\_cuts, branch\_points, calling\_sequence, class\_members, classify\_function, definition, describe, differentiation\_rule, function\_classes, identities, integral\_form, known\_functions, relate, required\_assumptions, series, singularities, special\_values, specialize, sum\_form, symmetries, synonyms] \tag{2.2.1}$$

Mathematical information is then available for these functions

```
> FunctionAdvisor( known_functions );
The functions on which information is available via
    > FunctionAdvisor( function_name );
are:
```

[AiryAi, AiryBi, AngerJ, BellB, BesselI, BesselJ, BesselK, BesselY, B, ChebyshevT, ChebyshevU, Chi, Ci, CoulombF, CylinderD, CylinderU, CylinderV, Dirac, Ei, EllipticCE, EllipticCK, EllipticCPi, EllipticE, EllipticF, EllipticK, EllipticModulus, EllipticNome, EllipticPi, FresnelC, FresnelS, Fresnelf, Fresnelg, Γ, GaussAGM, GegenbauerC, HankelH1, HankelH2, Heaviside, HermiteH, HeunB, HeunBPrime, HeunC, HeunCPrime, HeunD, HeunDPrime, HeunG, HeunGPrime, HeunT, HeunTPrime, Hypergeom, $\Im$, InverseJacobiAM, InverseJacobiCD, InverseJacobiCN, InverseJacobiCS, InverseJacobiDC, InverseJacobiDN, InverseJacobiDS, InverseJacobiNC, InverseJacobiND, InverseJacobiNS, InverseJacobiSC, InverseJacobiSD, InverseJacobiSN, JacobiAM, JacobiCD, JacobiCN, JacobiCS, JacobiDC, JacobiDN, JacobiDS, JacobiNC, JacobiND, JacobiNS, JacobiP, JacobiSC, JacobiSD, JacobiSN, JacobiTheta1, JacobiTheta2, JacobiTheta3, JacobiTheta4, JacobiZeta, KelvinBei, KelvinBer, KelvinHei, KelvinHer, KelvinKei, KelvinKer, KummerM, KummerU, LaguerreL, LambertW, LegendreP, LegendreQ, LerchPhi, Li, LommelS1, LommelS2, MathieuA, MathieuB, MathieuC, MathieuCE, **(2.2.2)**

MathieuCEPrime, MathieuCPrime, MathieuExponent, MathieuFloquet, MathieuFloquetPrime, MathieuS, MathieuSE, MathieuSEPrime, MathieuSPrime, MeijerG, Ψ, ℜ, Shi, Si, SphericalY, Ssi, Stirling1, Stirling2, StruveH, StruveL, WeberE, WeierstrassP, WeierstrassPPrime, WeierstrassSigma, WeierstrassZeta, WhittakerM, WhittakerW, Wrightomega, ζ, abs, arccos, arccosh, arccot, arccoth, arccsc, arccsch, arcsec, arcsech, arcsin, arcsinh, arctan, arctanh, argument, bernoulli, binomial, cos, cosh, cot, coth, csc, csch, csgn, dawson, dilog, doublefactorial, erf, erfc, erfi, euler, exp, factorial, harmonic, hypergeom, ln, lnGAMMA, log, pochhammer, polylog, sec, sech, signum, sin, sinh, tan, tanh, *unwindK*]

```
> nops(%);
```

$$179 \tag{2.2.3}$$

Lets see how some of these topics work

The ***form*** of any of these 128 Maple functions can be obtained via

```
> BesselJ;
```

$$\mathrm{BesselJ} \tag{2.2.4}$$

```
> FunctionAdvisor( syntax, % );
```

$$J_a(z) \tag{2.2.5}$$

```
> FunctionAdvisor(DE, BesselJ);
```

$$\left[f(z) = J_a(z), \left[\frac{d^2}{dz^2} f(z) = -\frac{\frac{d}{dz} f(z)}{z} + \frac{(a^2 - z^2) f(z)}{z^2}\right]\right] \tag{2.2.6}$$

```
> FunctionAdvisor(identities, BesselJ);
```

$$\Bigg[J_a(-z) = (-z)^a z^{-a} J_a(z), J_{a-1}(z) J_{-a}(z) + J_{1-a}(z) J_a(z) = \frac{2 \sin(\pi a)}{\pi z}, \tag{2.2.7}$$

$$J_a\left(\sqrt{z^2}\right) = z^{-a} (z^2)^{\frac{a}{2}} J_a(z), \left[J_a\left(b (c z^q)^p\right) = \frac{\left(b (c z^q)^p\right)^a J_a\left(b c^p z^{p q}\right)}{(b c^p z^{p q})^a},\right.$$

$$\left.\mathrm{And}(2\, p{::}integer)\right], J_a(z) = \frac{2 (a-1) J_{a-1}(z)}{z} - J_{a-2}(z), J_a(z)$$

$$= \frac{2\,(a+1)\,J_{a+1}(z)}{z} - J_{a+2}(z) \Bigg]$$

```
> FunctionAdvisor(display, Ei);
The symmetries for Ei are unknown to the FunctionAdvisor
Ei belongs to the subclass "Ei_related" of the class "1F1" and so, in
principle, it can be related to various of the 20 functions of those
classes - see FunctionAdvisor( "Ei_related" ); and FunctionAdvisor( "1F1"
);
```

$$describe = (\mathrm{Ei} = exponential\ integral)$$

$$classify\_function = (Ei\_related,\ 1F1)$$

$$definition = \left(\left[\mathrm{Ei}(z) = PV \int_{-\infty}^{z} \frac{\mathrm{e}^{\_k1}}{\_k1}\, \mathrm{d}\_k1,\ \mathrm{And}(z{::}real) \right], \left[\mathrm{Ei}_a(z) = \int_{1}^{\infty} \frac{1}{\mathrm{e}^{\_k1\, z}\ \_k1^{a}} \right.\right.$$

$$\left.\left. \mathrm{d}\_k1,\ \mathrm{And}(0 < \Re(z)) \right] \right)$$

$$analytic\_extension = \left(\mathrm{Ei}(z) = \gamma - \frac{\ln\left(\frac{1}{z}\right)}{2} + \frac{\ln(z)}{2} + z\ {}_2F_2(1, 1; 2, 2; z) \right)$$

$$singularities = \left([\mathrm{Ei}(z), z = \infty + \infty\,\mathrm{I}], [\mathrm{Ei}_a(z), a = \infty + \infty\,\mathrm{I}, z = \infty + \infty\,\mathrm{I}] \right)$$

$$branch\_points = \left([\mathrm{Ei}(z), z \in [0, \infty + \infty\,\mathrm{I}]], [\mathrm{Ei}_a(z), z \in \{0, \infty + \infty\,\mathrm{I}\}] \right)$$

$$branch\_cuts = \left([\mathrm{Ei}(z), z < 0], [\mathrm{Ei}_a(z), z < 0] \right)$$

$$special\_values = \Big[\mathrm{Ei}(\infty\,\mathrm{I}) = \mathrm{I}\,\pi, \mathrm{Ei}(-\infty\,\mathrm{I}) = -\mathrm{I}\,\pi, \mathrm{Ei}(0) = -\infty, \mathrm{Ei}(\infty) = \infty, \mathrm{Ei}(-\infty)$$

$$= 0, \mathrm{Ei}_0(z) = \frac{\mathrm{e}^{-z}}{z}, \mathrm{Ei}_{-1}(z) = \frac{\mathrm{e}^{-z}\,(1+z)}{z^2}, \mathrm{Ei}_0(0) = \infty + \infty\,\mathrm{I}, \mathrm{Ei}_1(0) = \infty +$$

$$\mathrm{undefined}\ \mathrm{I}, \left[\mathrm{Ei}_a(0) = \frac{1}{a-1}, \mathrm{And}(1 < \Re(a)) \right], \left[\mathrm{Ei}(z) = -U(1, 1, -z)\,\mathrm{e}^{z}, \mathrm{And}(z \right.$$

$$\left. < 0) \right], \left[\mathrm{Ei}(z) = \mathrm{Li}(\mathrm{e}^{z}), -\pi \le \Im(z) \le \pi \right], \left[\mathrm{Ei}_{\frac{1}{2}}(z) = \frac{\left(1 - \mathrm{erf}\left(\sqrt{z}\right)\right)\sqrt{\pi}}{\sqrt{z}}, \right.$$

$$\text{And}(z \neq 0)\Big]\Big], \left[\text{Ei}_{-\frac{1}{2}}(z) = -\frac{\sqrt{\pi}\,\text{erf}\left(\sqrt{z}\right)}{2\,z^{3/2}} + \frac{\sqrt{\pi}}{2\,z^{3/2}} + \frac{1}{z\,e^{z}}, \text{And}(z \neq 0)\right],$$

$$\Bigg[\text{Ei}_{a}(z) = -\text{Li}\left(\frac{1}{e^{z}}\right) - \frac{\ln\left(-\frac{1}{z}\right)}{2} + \frac{\ln(-z)}{2} - \ln(z), a = 1 \textbf{ And } -\pi \leq \Im(z) \textbf{ And}$$

$$\Im(z) \leq \pi\Bigg], \left[\text{Ei}_{a}(0) = \infty + \infty\,\text{I}, \text{And}(\Re(a) < 1)\right]\Bigg]$$

$$\textit{identities} = \Bigg[\text{Ei}\left(\sqrt{z^2}\right) = \text{Ei}(z) + \left(\frac{\sqrt{z^2}}{z} - 1\right)\text{Shi}(z) + \frac{\ln\left(\frac{1}{z}\right)}{2} + \frac{\ln(-\text{I}\,z)}{2}$$

$$+ \frac{\ln(\text{I}\,z)}{2} - \frac{\ln(z)}{2}, \text{Ei}(z) = \gamma - \frac{\ln\left(\frac{1}{z}\right)}{2} + \frac{\ln(z)}{2} + z\,{}_2F_2(1, 1; 2, 2; z), \text{Ei}_1(z)$$

$$= -\text{Ei}(-z) + \frac{\ln(-z)}{2} - \frac{\ln\left(-\frac{1}{z}\right)}{2} - \ln(z), \left[\text{Ei}_{-a}(z) = a!\,e^{-z}\left(\sum_{k=0}^{a}\frac{z^{k-a-1}}{k!}\right),\right.$$

$$\left.\text{And}(a{::}\textit{nonnegint})\right], \text{Ei}_{a}(z) = \frac{z\,\text{Ei}_{-2+a}(z) + (-2 + a - z)\,\text{Ei}_{-1+a}(z)}{-1 + a}, \text{Ei}_{a}(z)$$

$$= \frac{(-a + z)\,\text{Ei}_{1+a}(z) + (1 + a)\,\text{Ei}_{2+a}(z)}{z}\Bigg]$$

$$\textit{sum\_form} = \Bigg(\Bigg[\Bigg[\text{Ei}(z) = \gamma - \frac{\ln\left(\frac{1}{z}\right)}{2} + \frac{\ln(z)}{2} + \sum_{\_k1=1}^{\infty}\frac{z^{\_k1}}{\_k1\ \_k1!},$$

$$\textit{with no restrictions on } (z)\Bigg], \left[\text{Ei}_{a}(z) = \sum_{\_k1=0}^{\infty}\frac{(-z)^{\_k1}}{\Gamma(\_k1 + 1)\,(-\_k1 - 1 + a)}\right.$$

$$\left. + z^{a-1}\,\Gamma(1 - a), \text{And}(a{::}\textbf{Not}\ \textit{posint})\right], \Bigg[\text{Ei}_{a}(z)$$

$$= \sum_{\_k1=0}^{\infty}\frac{(-1)^{a+2\,\_k1}\,z^{a-1+\_k1}\,\left(-\Psi(\_k1 + 1) + \ln(z)\right)}{\Gamma(\_k1 + 1)\,\Gamma(a)\,e^{z}}$$

$$+ \sum_{\_k1=0}^{-2+a}\frac{(-1)^{\_k1}\,z^{\_k1}\,\Gamma(-\_k1 - 1 + a)}{\Gamma(a)\,e^{z}}, a{::}\textit{posint} \textbf{ And } z \neq 0\Bigg], \Big[\text{Ei}_{a}(z)$$

$$= \sum_{\_k1=0}^{-a} \frac{(-a)!\, z^{a-1+\_k1}}{e^z\, \_k1!}, a::negint \textbf{ And } z \neq 0 \Big]\Big]\Big)$$

$$integral\_form = \left(\left[\left[\mathrm{Ei}(z) = PV\int_{-\infty}^{z} \frac{e^{\_k1}}{\_k1}\, d\_k1, \mathrm{And}(z::real)\right]\right], \left[\mathrm{Ei}(z) = \int_0^z \frac{e^{\_k1}-1}{\_k1}\, d\_k1 + \ln(z) + \gamma, \mathrm{And}(0 < \Re(z))\right], \left[\mathrm{Ei}_a(z) = \int_1^\infty \frac{1}{e^{\_k1\, z}\, \_k1^a}\, d\_k1, \mathrm{And}(0 < \Re(z))\right]\right)$$

$$differentiation\_rule = \left(\frac{\partial}{\partial a} \mathrm{Ei}_{f(a)}(z) = -\left(\frac{d}{da} f(a)\right) z^{f(a)}\, G_{2,3}^{3,0}\left(z \,\middle|\, \begin{matrix} 0, 0 \\ -1, -1, -f(a) \end{matrix}\right), \frac{\partial}{\partial z} \mathrm{Ei}_a(f(z)) = -\left(\frac{d}{dz} f(z)\right) \mathrm{Ei}_{a-1}(f(z))\right)$$

$$DE = \left(\left[\left[f(z) = \mathrm{Ei}(z), \left[\frac{d^2}{dz^2} f(z) = \frac{\left(\frac{d}{dz} f(z)\right)(z-1)}{z}\right]\right]\right], \left[f(z) = \mathrm{Ei}_a(z), \left[\frac{d^2}{dz^2} f(z) = \frac{(a-z-2)\left(\frac{d}{dz} f(z)\right)}{z} + \frac{(a-1) f(z)}{z}\right]\right]\right)$$

$$series = \left(series(\mathrm{Ei}(z), z, 4) = \gamma + \ln(z) + z + \frac{1}{4} z^2 + \frac{1}{18} z^3 + O(z^4), series(\mathrm{Ei}_a(z), a, 4) = \frac{e^{-z}}{z} - G_{1,2}^{2,0}\left(z \,\middle|\, \begin{matrix} 0 \\ -1, -1 \end{matrix}\right) a + G_{2,3}^{3,0}\left(z \,\middle|\, \begin{matrix} 0, 0 \\ -1, -1, -1 \end{matrix}\right) a^2 - G_{3,4}^{4,0}\left(z \,\middle|\, \begin{matrix} 0, 0, 0 \\ -1, -1, -1, -1 \end{matrix}\right) a^3 + O(a^4)\right)$$

$$asymptotic\_expansion = \left(asympt(\mathrm{Ei}(z), z, 4) = \left(\frac{1}{z} + \frac{1}{z^2} + \frac{2}{z^3} + O\left(\frac{1}{z^4}\right)\right) e^z, asympt(\mathrm{Ei}_a(z), z, 4) = \frac{\frac{1}{z} - \frac{a!}{(a-1)!\, z^2} + \frac{(a+1)!}{(a-1)!\, z^3} + O\left(\frac{1}{z^4}\right)}{e^z}\right) \quad \textbf{(2.2.8)}$$

Functions can be related or specialized in different manners,

```
> FunctionAdvisor( relate, sin, WhittakerM);
```

$$\sin(z) = -\frac{I}{2} M_{0, \frac{1}{2}}(2\, I\, z) \quad \textbf{(2.2.9)}$$

```
> FunctionAdvisor( relate, KummerU, BesselK );
```

$$K_a(z) = \frac{\sqrt{\pi}\,(2z)^a\, U\left(a+\frac{1}{2}, 2a+1, 2z\right)}{e^z} \qquad \textbf{(2.2.10)}$$

```
> FunctionAdvisor(specialize, Ei);
```

$$\left[\mathrm{Ei}_a(z) = \frac{2\,\Gamma(2-a)\,(\mathrm{I}\,z)^{-1+\frac{a}{2}}\, F_{-\frac{a}{2}}\left(-\frac{\mathrm{I}}{2}\,a, \frac{\mathrm{I}}{2}\,z\right)}{(a-1)\,e^{\frac{\mathrm{I}\,\pi\,a}{4}+\frac{z}{2}}} + z^{a-1}\,\Gamma(1-a), \right. \qquad \textbf{(2.2.11)}$$

$$\left. \textit{with no restrictions on } (a, z)\right], \left[\mathrm{Ei}_a(z) = z^{a-1}\,\Gamma(1-a, z), \mathrm{And}(z \neq 0)\right], \left[\mathrm{Ei}_a(z)\right.$$

$$\left. = \frac{HB\left(2-2a, 0, 2a, 0, \sqrt{-z}\right)}{a-1} + z^{a-1}\,\Gamma(1-a), \textit{with no restrictions on } (a, z)\right]$$

$$, \left[\mathrm{Ei}_a(z) = \frac{(-z+1)\,HC\left(1, 1-a, 1, -\frac{a}{2}, \frac{1}{2}+\frac{a}{2}, z\right)}{a-1} + z^{a-1}\,\Gamma(1-a), \right.$$

$$\left. \textit{with no restrictions on } (a, z)\right], \left[\mathrm{Ei}_a(z) = \frac{M(1-a, 2-a, -z)}{a-1} + z^{a-1}\,\Gamma(1\right.$$

$$\left. -a), \textit{with no restrictions on } (a, z)\right], \left[\mathrm{Ei}(z) = -U(1, 1, -z)\,e^z, \mathrm{And}(z < 0)\right],$$

$$\left[\mathrm{Ei}_a(z) = \frac{U(1, 2-a, z)}{e^z}, \mathrm{And}(z \neq 0)\right], \left[\mathrm{Ei}_a(z) = z^{a-1}\left(\Gamma(1-a)\right.\right.$$

$$\left.\left. - \frac{L_{a-1}^{(1-a)}(-z)}{z^{a-1}\,(1-a)\begin{cases} 1 & a-1=0 \\ \frac{\sin(\pi\,(a-1))}{\pi\,(a-1)} & \textit{otherwise} \end{cases}}\right), \mathrm{And}(a::\textbf{Not}\ \textit{posint})\right],$$

$$\left[\mathrm{Ei}(z) = \mathrm{Li}\left(e^z\right), -\pi \leq \Im(z) \leq \pi\right], \left[\mathrm{Ei}_a(z) = -\mathrm{Li}\left(\frac{1}{e^z}\right) - \frac{\ln\left(-\frac{1}{z}\right)}{2} + \frac{\ln(-z)}{2}\right.$$

$- \ln(z), a = 1 \textbf{ And } -\pi \le \Im(z) \textbf{ And } \Im(z) \le \pi \Big], \Big[\mathrm{Ei}(z) = -G_{1,2}^{2,0}\left(-z \,\middle|\, \begin{matrix}1\\0, 0\end{matrix}\right),$

$\textit{with no restrictions on } (z)\Big], \Big[\mathrm{Ei}_a(z) = G_{1,2}^{2,0}\left(z \,\middle|\, \begin{matrix}a\\0, a-1\end{matrix}\right), \textit{with no restrictions on } (a,$

$z)\Big], \left[\mathrm{Ei}_a(z) = z^{a-1}\left(\Gamma(1-a) - \frac{M_{\frac{a}{2}, \frac{1}{2}-\frac{a}{2}}(-z)\,(-z)^{-1+\frac{a}{2}}}{z^{a-1}\,(1-a)\,\mathrm{e}^{\frac{z}{2}}}\right),\right.$

$\left.\textit{with no restrictions on } (a, z)\right], \left[\mathrm{Ei}(z) = -\frac{W_{-\frac{1}{2}, 0}(-z)\,\mathrm{e}^{\frac{z}{2}}}{\sqrt{-z}}, \mathrm{And}(z < 0)\right], \left[\mathrm{Ei}_a(z)\right.$

$\left.= \frac{W_{-\frac{a}{2}, \frac{1}{2}-\frac{a}{2}}(z)\, z^{-1+\frac{a}{2}}}{\mathrm{e}^{\frac{z}{2}}}, \mathrm{And}(z \ne 0)\right], \left[\mathrm{Ei}_a(z)\right.$

$\left.= \frac{-I\left(-2\cos\left(\frac{I}{2}(I\pi + z)\right)\mathrm{e}^{-\frac{z}{2}} + I\right)}{z}, \mathrm{And}(a = 0)\right], \left[\mathrm{Ei}_a(z)\right.$

$\left.= \frac{-I\left(-2\cosh\left(\frac{I\pi}{2} + \frac{z}{2}\right)\mathrm{e}^{-\frac{z}{2}} + I\right)}{z}, \mathrm{And}(a = 0)\right], \left[\mathrm{Ei}_a(z)\right.$

$\left.= \frac{2\,I\,\mathrm{e}^{-\frac{z}{2}} + \csc\left(\frac{I}{2}z\right)}{\csc\left(\frac{I}{2}z\right)z}, \mathrm{And}(a = 0)\right], \left[\mathrm{Ei}_a(z) = \frac{\mathrm{csch}\left(-\frac{z}{2}\right) + 2\,\mathrm{e}^{-\frac{z}{2}}}{\mathrm{csch}\left(-\frac{z}{2}\right)z}, \mathrm{And}(a\right.$

$\left.= 0)\right], \left[\mathrm{Ei}_a(z) = \frac{-2\,\mathrm{dawson}\left(\sqrt{-z}\right)\mathrm{e}^{-z}\sqrt{z} + \sqrt{-z}\,\sqrt{\pi}}{\sqrt{z}\,\sqrt{-z}}, \mathrm{And}\left(a = \frac{1}{2}\right)\right],$

$\left[\mathrm{Ei}_a(z) = -\frac{\left(-1 + \mathrm{erf}\left(\sqrt{z}\right)\right)\sqrt{\pi}}{\sqrt{z}}, \mathrm{And}\left(a = \frac{1}{2}\right)\right], \left[\mathrm{Ei}_a(z) =\right.$

$$- \frac{\left(\mathrm{erfi}\left(\sqrt{-z}\right)\sqrt{z} - \sqrt{-z}\right)\sqrt{\pi}}{\sqrt{z}\sqrt{-z}}, \mathrm{And}\left(a = \frac{1}{2}\right)\Bigg]\Bigg], \Bigg[\mathrm{Ei}(z) = \gamma - \frac{\ln\left(\frac{1}{z}\right)}{2}$$

$$+ \frac{\ln(z)}{2} + z\, {}_2F_2(1, 1; 2, 2; z), \textit{with no restrictions on } (z)\Bigg], \Bigg[\mathrm{Ei}_a(z)$$

$$= \frac{{}_1F_1(1 - a; 2 - a; -z)}{a - 1} + z^{a-1}\,\Gamma(1 - a), \mathrm{And}(a::\textbf{Not}\; posint)\Bigg], \Bigg[\mathrm{Ei}_a(z) = \Bigg($$

$$- \frac{z\, {}_2F_2(1, 1; 2, 1 + a; -z)}{\Gamma(1 + a)\,(-1)^a} + \frac{\sum_{\_k1 = 0}^{-2 + a} - \frac{(-1)^{\_k1}\, z^{\_k1}}{(1 - a + \_k1)\,\Gamma(\_k1 + 1)}}{z^{a-1}}$$

$$+ \frac{a\left(-\Psi(a) + \ln(z)\right)}{\Gamma(1 + a)\,(-1)^a}\Bigg) z^{a-1}, \mathrm{And}(a::posint)\Bigg], \Bigg[\mathrm{Ei}_a(z)$$

$$= \frac{2\,\mathrm{I}\,\mathrm{e}^{-\frac{z}{2}} + \sec\left(\frac{\mathrm{I}}{2}\,(\mathrm{I}\,\pi + z)\right)}{\sec\left(\frac{\mathrm{I}}{2}\,(\mathrm{I}\,\pi + z)\right) z}, \mathrm{And}(a = 0)\Bigg], \Bigg[\mathrm{Ei}_a(z)$$

$$= \frac{-\mathrm{I}\left(\mathrm{I}\,\mathrm{sech}\left(\frac{\mathrm{I}\,\pi}{2} + \frac{z}{2}\right) - 2\,\mathrm{e}^{-\frac{z}{2}}\right)}{z\,\mathrm{sech}\left(\frac{\mathrm{I}\,\pi}{2} + \frac{z}{2}\right)}, \mathrm{And}(a = 0)\Bigg], \Bigg[\mathrm{Ei}_a(z)$$

$$= \frac{-\mathrm{I}\left(-2\sin\left(\frac{\mathrm{I}}{2}\, z\right)\mathrm{e}^{-\frac{z}{2}} + \mathrm{I}\right)}{z}, \mathrm{And}(a = 0)\Bigg], \Bigg[\mathrm{Ei}_a(z)$$

$$= \frac{2\sinh\left(-\frac{z}{2}\right)\mathrm{e}^{-\frac{z}{2}} + 1}{z}, \mathrm{And}(a = 0)\Bigg]$$

Differential Polynomial Form for non-polynomial expressions

Goal: *"Given a system of equations (differential or not) containing **non-polynomial** objects, compute an equivalent **differential polynomial** system such that any solution of the **non-***

***polynomial** system is also a solution of this **differential polynomial** one".*

These routines allow **to tackle non-polynomial systems using methods for polynomial ones.** **dpolyform** is used by Maple's ODE and PDE solvers when uncoupling non-polynomial systems of differential equations

Examples

```
> restart; with(PDEtools):
```

For ease of reading we declare these functions to be displayed in a compact way and derivatives to be displayed indexed

```
> declare(g(x,y), _F1(x,y), _F2(x,y), _F3(x,y));
```

$g(x, y)$ *will now be displayed as* g

$\_F1(x, y)$ *will now be displayed as* $\_F1$

$\_F2(x, y)$ *will now be displayed as* $\_F2$

$\_F3(x, y)$ *will now be displayed as* $\_F3$ **(2.3.1.1)**

Consider the following **non-polynomial** expression

```
> e1 := g(x,y) = tan(2*x+y^(1/2));
```

$$e1 := g = \tan\left(2x + \sqrt{y}\right) \quad \textbf{(2.3.1.2)}$$

Since tan and powers admit a DPF, the composition of them also does:

```
> DPF := dpolyform(e1);
```

$$DPF := \left[g - \_F1 = 0, 2\,\_F2\,\_F1_y - 1 - \_F1^2 = 0, -2\,\_F1^2 + \_F1_x - 2 = 0, \_F2^2 - y = 0, \_F2_x = 0\right], \left[\_F2 \neq 0, \_F1_x \neq 0, \_F2\,\_F1_y \neq 0\right], \left[\_F2 = \sqrt{y}, \_F1 = \tan\left(2x + \sqrt{y}\right)\right] \quad \textbf{(2.3.1.3)}$$

In the above there is a sequence of 3 lists. The first list contains the "Equations" of the problem; the second list contains the "Inequations" and the third list contains the "back-substitution equations", telling which non-polynomial object is represented by each auxiliary function _Fn.

The fact that the representation is polynomial allows one to run a differential elimination process and completely remove the auxiliary functions **_F1**, **_F2**, ...

```
> pde_sys := dpolyform(e1, no_Fn);
```

$$pde\_sys := \left[g_x = 2g^2 + 2, g_y^2 = \frac{g^4}{4y} + \frac{g^2}{2y} + \frac{1}{4y}\right] \&\text{where} \left[g^2 + 1 \neq 0, g_y \neq 0\right] \quad \textbf{(2.3.1.4)}$$

A tool is provided for testing these manipulations ...

```
> e1;
```

$$g = \tan\left(2\,x + \sqrt{y}\right) \tag{2.3.1.5}$$

```
> pdetest(%, pde_sys);
```

$$[0, 0] \tag{2.3.1.6}$$

Identities for special functions, or relations between them and simpler Liouvillian functions

Example

```
> 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 ' **(2.3.1.7)**

```
> a3 := y(x) = hypergeom([1],[2],-2*I*x);
```

$$a3 := y = {}_1F_1(1;\,2;\,-2\,\mathrm{I}\,x) \tag{2.3.1.8}$$

A polynomial (in this case linear) ODE satisfied by **y(x)** is given by:

```
> e3 := dpolyform(a3, no_Fn);
```

$$e3 := \left[y'' = \frac{(-2\,\mathrm{I}\,x - 2)\,y'}{x} - \frac{2\,\mathrm{I}\,y}{x}\right] \&\text{where } [y \neq 0] \tag{2.3.1.9}$$

The ODE above also admits a solution in terms of Liouvillian functions, which can be obtained by using Kovacic's algorithm. See DEtools[kovacicsols].

```
> a3_bis := dsolve( op([1,1],e3), [Kovacic] );
```

$$a3\_bis := y = \frac{\_C1}{x} + \frac{\_C2\,\mathrm{e}^{-2\,\mathrm{I}\,x}}{x} \tag{2.3.1.10}$$

This means that the hypergeometric function appearing in **a3** is equal to the right-hand side of **a3_bis** for some particular values of **_C1** and **_C2**. The actual values of **_C1** and **_C2** can be obtained by expanding in series. In Brief, the system of equations satisfied by these constants is

```
> e4 := a3 - a3_bis;
```

$$e4 := 0 = {}_1F_1(1;\,2;\,-2\,\mathrm{I}\,x) - \frac{\_C1}{x} - \frac{\_C2\,\mathrm{e}^{-2\,\mathrm{I}\,x}}{x} \tag{2.3.1.11}$$

```
> series(rhs(e4),x,2);
```

$$(-\_C2 - \_C1)\,x^{-1} + 2\,\mathrm{I}\,\_C2 + 1 + \mathrm{O}(x) \tag{2.3.1.12}$$

```
> sys := map(eq -> eq=0, {coeffs(convert(%, polynom), x)});
```

$$sys := \{-\_C2 - \_C1 = 0,\, 2\,\mathrm{I}\,\_C2 + 1 = 0\} \tag{2.3.1.13}$$

resulting in:

```
> ans_C := solve(sys,{_C1,_C2});
```

$$ans\_C := \left\{\_C1 = -\frac{\mathrm{I}}{2},\, \_C2 = \frac{\mathrm{I}}{2}\right\} \tag{2.3.1.14}$$

At these values of _C1 and _C2,

```
> eval(e4, ans_C);
```

$$0 = {}_1F_1(1;2;-2\,I\,x) + \frac{I}{2\,x} - \frac{I\,e^{-2\,I\,x}}{2\,x} \qquad \textbf{(2.3.1.15)}$$

from where the hypergeometric function can be isolated, resulting in the desired identity.

```
> isolate(%,hypergeom([1],[2],-2*I*x));
```

$${}_1F_1(1;2;-2\,I\,x) = -\frac{I}{2\,x} + \frac{I\,e^{-2\,I\,x}}{2\,x} \qquad \textbf{(2.3.1.16)}$$

Conclusion

"Research and education are two things highly inter-related"

- Constructive learning processes are mostly based on the building of logic structures by testing conjectures and analyzing the results. The proportion between success (the conjecture solves the problem) and frustration plays an important role as an emotional (+/-) accelerating factor for the whole "learning & discovery" process.

- The simultaneous analysis of a greater number of results turns apparent the underlying logic structures more rapidly, and can strengthen the intuition unexpectedly.

- Genuine learning processes only happen when the individual who is learning participates actively.

- Inspiration is a function of intuition, excitement and fun, transformed into results through heavy exploration. Symbolic computation can be used with these purposes, perhaps as the most important educational and research tool available at present.
