```
> ec1:=diff(x(t),t)=x(t)+4*y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) + 4\, y(t) \tag{1}$$

```
> ec2:=diff(y(t),t)=x(t)+y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) + y(t) \tag{2}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) + 4\, y(t),\ \frac{d}{dt}\, y(t) = x(t) + y(t) \tag{3}$$

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

$$infolevel_{dsolve} := 5 \tag{4}$$

```
> dsolve({sist},{x(t),y(t)})
-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 2*(diff(y(x) x))+3*y(x) y(x) singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
checking if the LODE has constant coefficients
<- constant coefficients successful
```

$$\left\{x(t) = c_1\, e^{3\,t} + c_2\, e^{-t},\ y(t) = \frac{c_1\, e^{3\,t}}{2} - \frac{c_2\, e^{-t}}{2}\right\} \tag{5}$$

```
> sist:=diff(x(t),t)=2*x(t)-y(t),diff(y(t),t)=x(t)+2*y(t)
```

$$sist := \frac{d}{dt}\, x(t) = 2\, x(t) - y(t),\ \frac{d}{dt}\, y(t) = x(t) + 2\, y(t) \tag{6}$$

```
> dsolve({sist},{x(t),y(t)})
-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 4*(diff(y(x) x))-5*y(x) y(x) singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
checking if the LODE has constant coefficients
<- constant coefficients successful
```

$$\left\{x(t) = e^{2\,t}\, (c_2 \cos(t) + c_1 \sin(t)),\ y(t) = -e^{2\,t}\, (\cos(t)\, c_1 - \sin(t)\, c_2)\right\} \tag{7}$$

```
> ec1:=diff(x(t),t)=x(t)-y(t)+z(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) - y(t) + z(t) \tag{8}$$

```
> ec2:=diff(y(t),t)=x(t)+y(t)-z(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) + y(t) - z(t) \tag{9}$$

```
> ec3:=diff(z(t),t)=-y(t)+2*z(t)
```

$$ec3 := \frac{d}{dt}\, z(t) = -y(t) + 2\, z(t) \qquad \mathbf{(10)}$$

```
> sist:=ec1,ec2,ec3
```

$$sist := \frac{d}{dt}\, x(t) = x(t) - y(t) + z(t),\ \frac{d}{dt}\, y(t) = x(t) + y(t) - z(t),\ \frac{d}{dt}\, z(t) = -y(t) + 2\, z(t) \qquad \mathbf{(11)}$$

```
> dsolve({sist},{x(t),y(t),z(t)})
-> Solving each unknown as a function of the next ones using the
order: [x(t), y(t), z(t)]
-> Calling odsolve with the ODE diff(diff(diff(y(x) x) x) x) =
4*(diff(diff(y(x) x) x))-5*(diff(y(x) x))+2*y(x) y(x) singsol =
none
Methods for third order ODEs:
--- Trying classification methods ---
trying a quadrature
checking if the LODE has constant coefficients
<- constant coefficients successful
```

$$\left\{x(t) = c_1\, e^{2\,t} + c_2\, e^{t} + c_3\, e^{t}\, t + c_3\, e^{t},\ y(t) = e^{t}\, (c_3\, t + c_2 - c_3),\ z(t) = c_1\, e^{2\,t} + c_2\, e^{t} + c_3\, e^{t}\, t\right\} \qquad \mathbf{(12)}$$

```
> ec1:=diff(x(t),t)=5*x(t)+3*y(t)+1
```

$$ec1 := \frac{d}{dt}\, x(t) = 5\, x(t) + 3\, y(t) + 1 \qquad \mathbf{(13)}$$

```
> ec2:=diff(y(t),t)=-6*x(t)-4*y(t)+exp(-t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -6\, x(t) - 4\, y(t) + e^{-t} \qquad \mathbf{(14)}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = 5\, x(t) + 3\, y(t) + 1,\ \frac{d}{dt}\, y(t) = -6\, x(t) - 4\, y(t) + e^{-t} \qquad \mathbf{(15)}$$

```
> dsolve({sist},{x(t),y(t)})
-> Solving each unknown as a function of the next ones using the
order: [y(t), x(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = diff(y(x)
x)+2*y(x)+4+3*exp(-x) y(x) singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
trying high order exact linear fully integrable
trying differential order: 2; linear nonhomogeneous with
symmetry [0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
   checking if the LODE has constant coefficients
   <- constant coefficients successful
```

```
   -> Determining now a particular solution to the non-homogeneous ODE
      building a particular solution using variation of parameters
<- solving first the homogeneous part of the ODE successful
```

$$\left\{x(t) = e^{2t} c_2 + e^{-t} c_1 - 2 - t e^{-t} - \frac{e^{-t}}{3}, y(t) = -e^{2t} c_2 - 2 e^{-t} c_1 + \frac{e^{-t}}{3} + 2 t e^{-t} + 3\right\} \quad (16)$$

```
> ec1:=diff(x(t),t)=x(t)+3*y(t)+cos(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) + 3\, y(t) + \cos(t) \quad (17)$$

```
> ec2:=diff(y(t),t)=x(t)-y(t)+2*t
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) - y(t) + 2\, t \quad (18)$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) + 3\, y(t) + \cos(t), \frac{d}{dt}\, y(t) = x(t) - y(t) + 2\, t \quad (19)$$

```
> dsolve({sist},{x(t),y(t)})
-> Solving each unknown as a function of the next ones using the order: [x(t), y(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 2+4*y(x)-2*x+cos(x) y(x) singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
trying high order exact linear fully integrable
trying differential order: 2; linear nonhomogeneous with symmetry [0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
   checking if the LODE has constant coefficients
   <- constant coefficients successful
   -> Determining now a particular solution to the non-homogeneous ODE
      building a particular solution using variation of parameters
<- solving first the homogeneous part of the ODE successful
```

$$\left\{x(t) = -e^{-2t} c_2 + 3 c_1 e^{2t} + \frac{\sin(t)}{5} - \frac{\cos(t)}{5} - \frac{3t}{2}, y(t) = e^{-2t} c_2 + c_1 e^{2t} - \frac{\cos(t)}{5} + \frac{t}{2} - \frac{1}{2}\right\} \quad (20)$$

```
> ec1:=diff(x(t),t)=x(t)-2*y(t)-2*z(t)+exp(-t)
```

(21)

$$ec1 := \frac{d}{dt}\, x(t) = x(t) - 2\, y(t) - 2\, z(t) + e^{-t} \quad \textbf{(21)}$$

```
> ec2:=diff(y(t),t)=-2*x(t)+y(t)+2*z(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -2\, x(t) + y(t) + 2\, z(t) \quad \textbf{(22)}$$

```
> ec3:=diff(z(t),t)=2*x(t)-y(t)-3*z(t)+exp(-t)
```

$$ec3 := \frac{d}{dt}\, z(t) = 2\, x(t) - y(t) - 3\, z(t) + e^{-t} \quad \textbf{(23)}$$

```
> sist:=ec1,ec2,ec3
```

$$sist := \frac{d}{dt}\, x(t) = x(t) - 2\, y(t) - 2\, z(t) + e^{-t}, \frac{d}{dt}\, y(t) = -2\, x(t) + y(t) + 2\, z(t), \frac{d}{dt}\, z(t) = 2\, x(t) - y(t) - 3\, z(t) + e^{-t} \quad \textbf{(24)}$$

```
> dsolve({sist},{x(t),y(t),z(t)})
-> Solving each unknown as a function of the next ones using the
order: [z(t), x(t), y(t)]
-> Calling odsolve with the ODE diff(y(x) x) = -y(x)-y(x)-(diff
(y(x) x))+exp(-x) y(x) singsol = none
Methods for first order ODEs:
--- Trying classification methods ---
trying a quadrature
trying 1st order linear
<- 1st order linear successful
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 3*y(x) y
(x) singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
checking if the LODE has constant coefficients
<- constant coefficients successful
```

$$\left\{x(t) = -c_2\, e^{\sqrt{3}\, t} - c_3\, e^{-\sqrt{3}\, t} + e^{-t}\, c_1 + t\, e^{-t}, y(t) = c_2\, e^{\sqrt{3}\, t} + c_3\, e^{-\sqrt{3}\, t}, z(t) = \frac{c_2 \sqrt{3}\, e^{\sqrt{3}\, t}}{2} - \frac{c_3 \sqrt{3}\, e^{-\sqrt{3}\, t}}{2} - \frac{3\, c_2\, e^{\sqrt{3}\, t}}{2} - \frac{3\, c_3\, e^{-\sqrt{3}\, t}}{2} + e^{-t}\, c_1 + t\, e^{-t}\right\} \quad \textbf{(25)}$$

```
> restart
> with (DEtools), with (plots)
```

[*AreSimilar, Closure, DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, Desingularize, FindODE, FunctionDecomposition, GCRD, Gosper, Heunsols, Homomorphisms, IVPsol, IsHyperexponential, LCLM, MeijerGsols, MultiplicativeDecomposition, ODEInvariants, PDEchangecoords, PolynomialNormalForm, RationalCanonicalForm, ReduceHyperexp,* **(26)** *RiemannPsols, Xchange, Xcommutator, Xgauge, Zeilberger, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diff_table, diffop2de, dperiodic_sols, dpolyform, dsubs, eigenring, endomorphism_charpoly, equinv, eta_k, eulersols, exactsol, expsols, exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic, genhomosol, gensys, hamilton_eqs, hypergeometricsols, hypergeomsols, hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor, invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, rational_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom*], [*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

```
> ec1:=diff(x(t),t)=x(t)+4*y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) + 4\, y(t) \tag{27}$$

```
> ec2:=diff(y(t),t) = x(t)+y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) + y(t) \tag{28}$$

```
> cond:=x(0)=1,y(0)=2
```

$$cond := x(0) = 1,\ y(0) = 2 \tag{29}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) + 4\, y(t),\ \frac{d}{dt}\, y(t) = x(t) + y(t) \tag{30}$$

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

$$infolevel_{dsolve} := 5 \tag{31}$$

```
> sol:=dsolve({sist,cond},{x(t),y(t)})
```

-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]

$$sol := \left\{x(t) = \frac{5\,e^{3\,t}}{2} - \frac{3\,e^{-t}}{2},\ y(t) = \frac{5\,e^{3\,t}}{4} + \frac{3\,e^{-t}}{4}\right\} \tag{32}$$

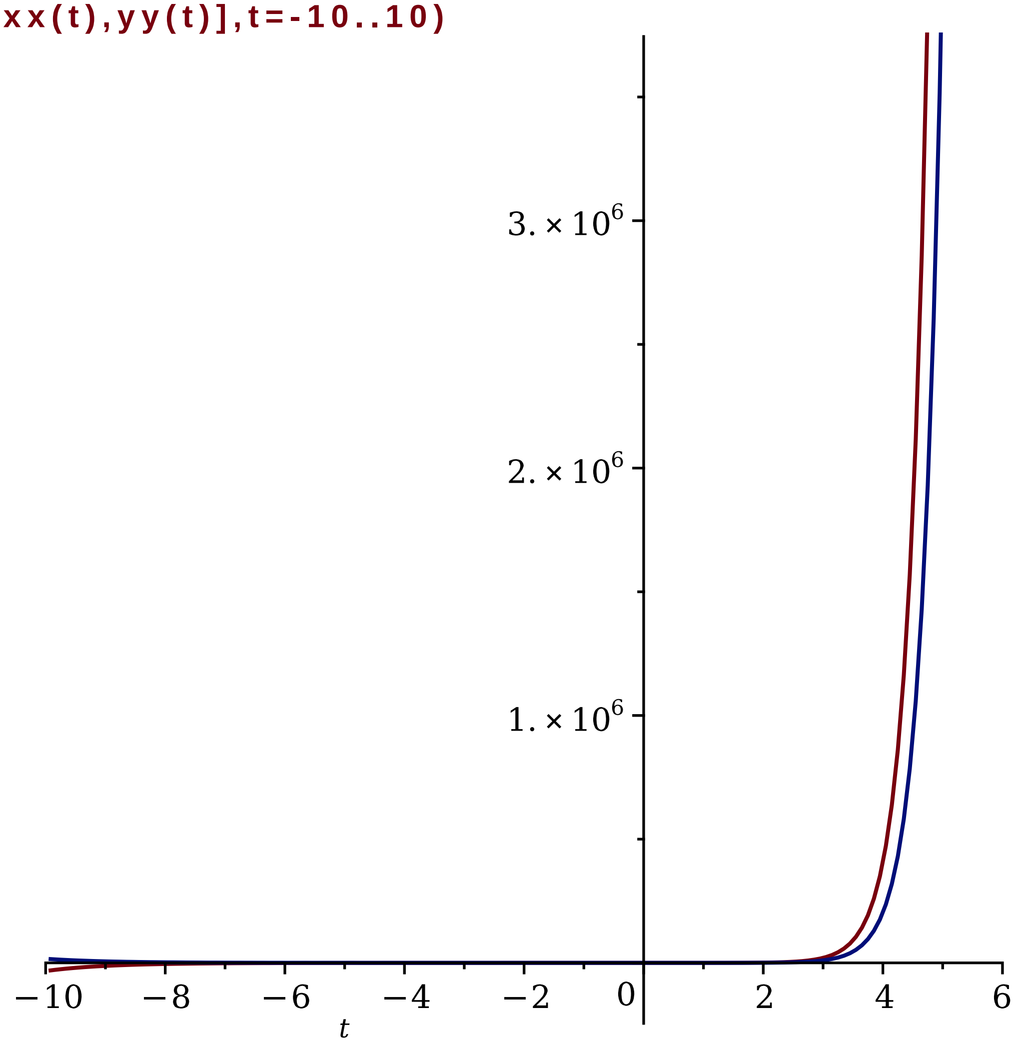
```
> xx:=unapply(rhs(sol[1]),t)
```

$$xx := t \mapsto \frac{5 \cdot e^{3 \cdot t}}{2} - \frac{3 \cdot e^{-t}}{2} \tag{33}$$

```
> yy:=unapply(rhs(sol[2]),t)
```

$$yy := t \mapsto \frac{5 \cdot e^{3 \cdot t}}{4} + \frac{3 \cdot e^{-t}}{4} \tag{34}$$

```
> plot([xx(t),yy(t)],t=-10..10)
```




```
> ec1:=diff(x(t),t)=x(t)-y(t)+t-1
```

(35)

$$ec1 := \frac{d}{dt}\, x(t) = x(t) - y(t) + t - 1 \qquad (35)$$

```
> ec2:=diff(y(t),t)=-2*x(t)+4*y(t)+cos(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -2\, x(t) + 4\, y(t) + \cos(t) \qquad (36)$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) - y(t) + t - 1,\ \frac{d}{dt}\, y(t) = -2\, x(t) + 4\, y(t) + \cos(t) \qquad (37)$$

```
> cond:=x(0)=0,y(0)=1
```

$$cond := x(0) = 0,\ y(0) = 1 \qquad (38)$$

```
> sol:=dsolve({sist,cond},{x(t),y(t)})
```

```
-> Solving each unknown as a function of the next ones using the
order: [y(t), x(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 5*(diff(y
(x) x))+5-2*y(x)-4*x-cos(x) y(x) explicit singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
trying high order exact linear fully integrable
trying differential order: 2; linear nonhomogeneous with
symmetry [0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
   checking if the LODE has constant coefficients
   <- constant coefficients successful
   -> Determining now a particular solution to the non-
homogeneous ODE
      building a particular solution using variation of
parameters
<- solving first the homogeneous part of the ODE successful
```

$$sol := \left\{ x(t) = e^{\frac{(5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} - \frac{5\sqrt{17}}{13}\right) + e^{-\frac{(-5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} + \frac{5\sqrt{17}}{13}\right) - \frac{\cos(t)}{26} + \frac{5\sin(t)}{26} - 2\,t - \frac{5}{2},\ y(t) = -\frac{e^{\frac{(5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} - \frac{5\sqrt{17}}{13}\right)\sqrt{17}}{2} + \frac{e^{-\frac{(-5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} + \frac{5\sqrt{17}}{13}\right)\sqrt{17}}{2} - \frac{3\, e^{\frac{(5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} - \frac{5\sqrt{17}}{13}\right)}{2} - \frac{3\, e^{-\frac{(-5+\sqrt{17})\,t}{2}} \left(\frac{33}{26} + \frac{5\sqrt{17}}{13}\right)}{2} + \frac{2\sin(t)}{13} - \frac{3\cos(t)}{13} - t - \frac{3}{2} \right\} \qquad (39)$$

```
> xx:=DEplot([sist],[x,y],t=-10..10,[[cond]],scene=[t,x(t)])
```

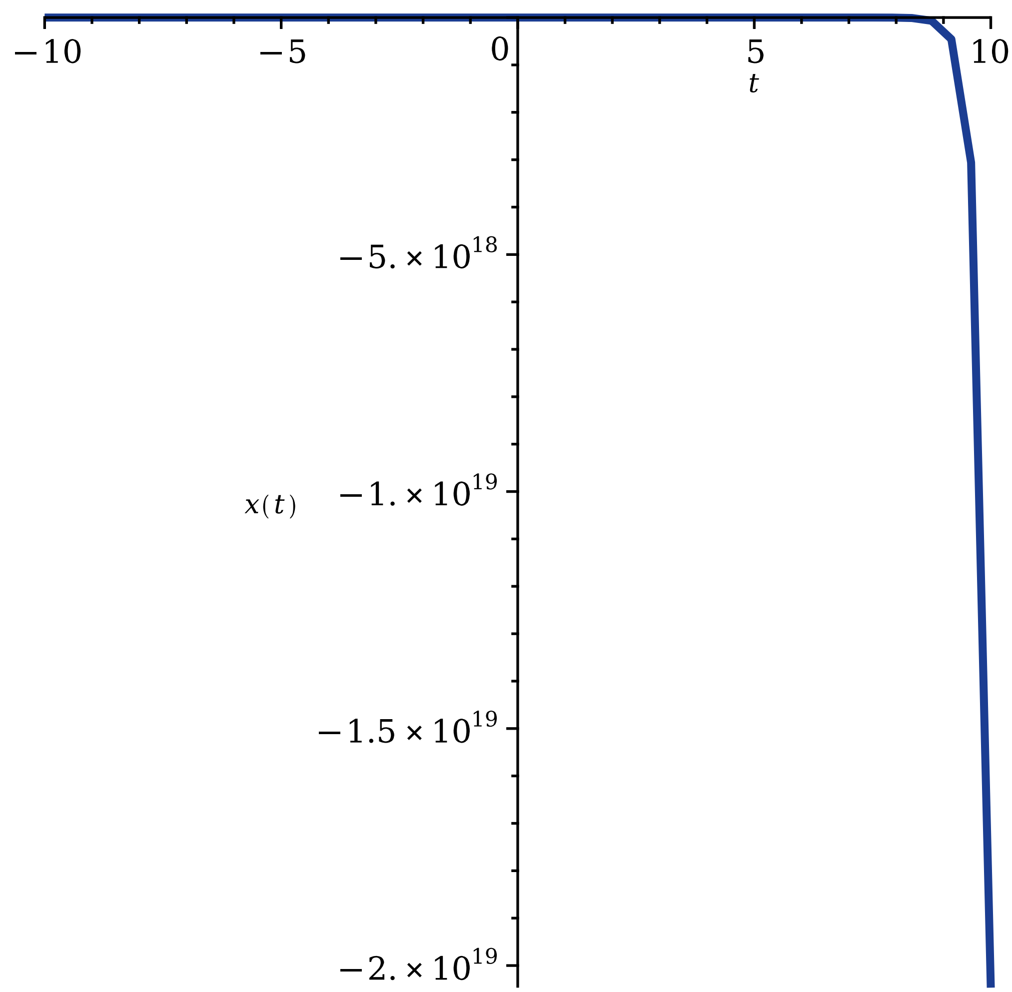
dsolve/numeric: entering dsolve/numeric
DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x))
namely (with Y' represented by YP)

$$[YP_1 = Y_1 - Y_2 + t - 1, YP_2 = -2\,Y_1 + 4\,Y_2 + \cos(t)]$$

DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]-Y[2]+X-1; YP[2] := -2*Y[1]+4*Y[2]+cos(X); 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[0., 1.]
dsolve/numeric/SC/firststep: Checking ODE for hardware floating point computation
dsolve/numeric/SC/firststep: Initial point OK for hardware floating point computation
dsolve/numeric/evalat_ext: loading hardware evalat




```
> yy:=DEplot([sist],[x,y],t=-10..10,[[cond]],linecolor=red,scene=
  [t,y(t)])
```

dsolve/numeric: entering dsolve/numeric

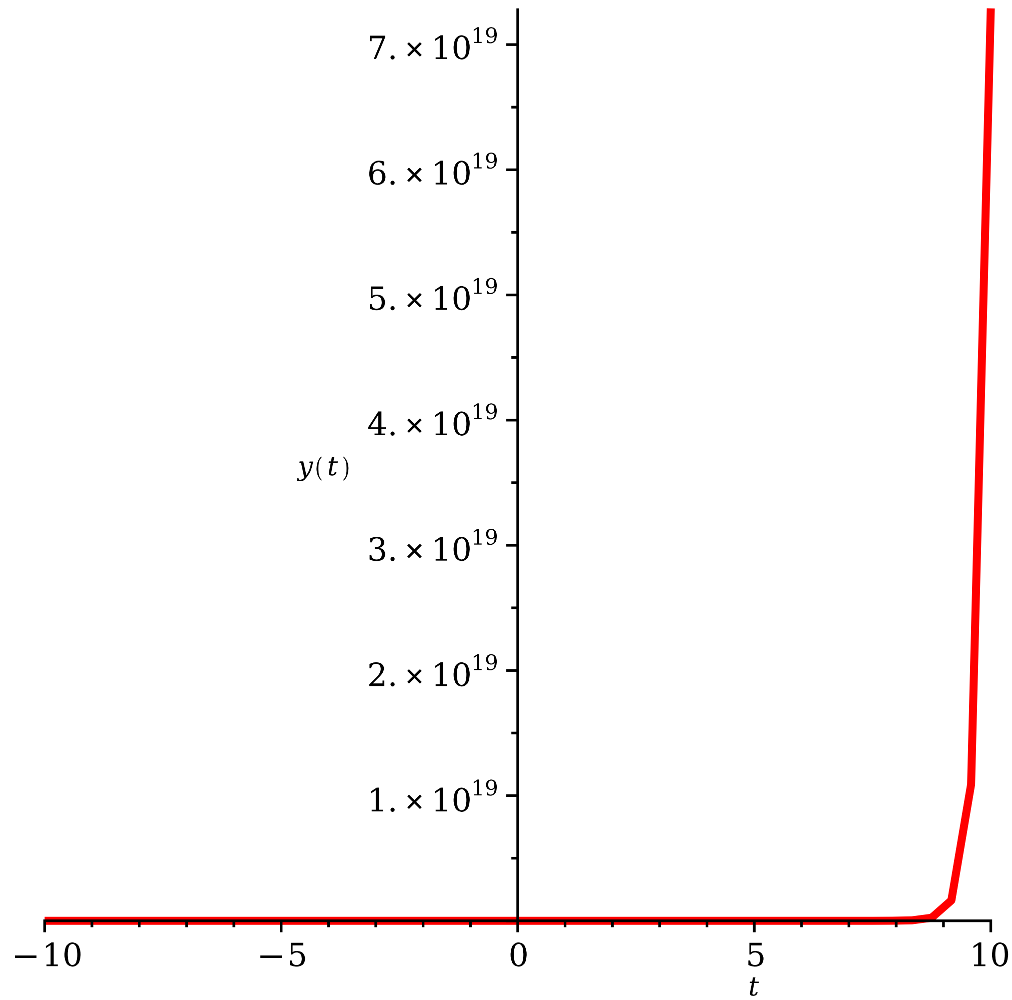
DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x))
namely (with Y' represented by YP)

$$[YP_1 = Y_1 - Y_2 + t - 1, YP_2 = -2\,Y_1 + 4\,Y_2 + \cos(t)]$$

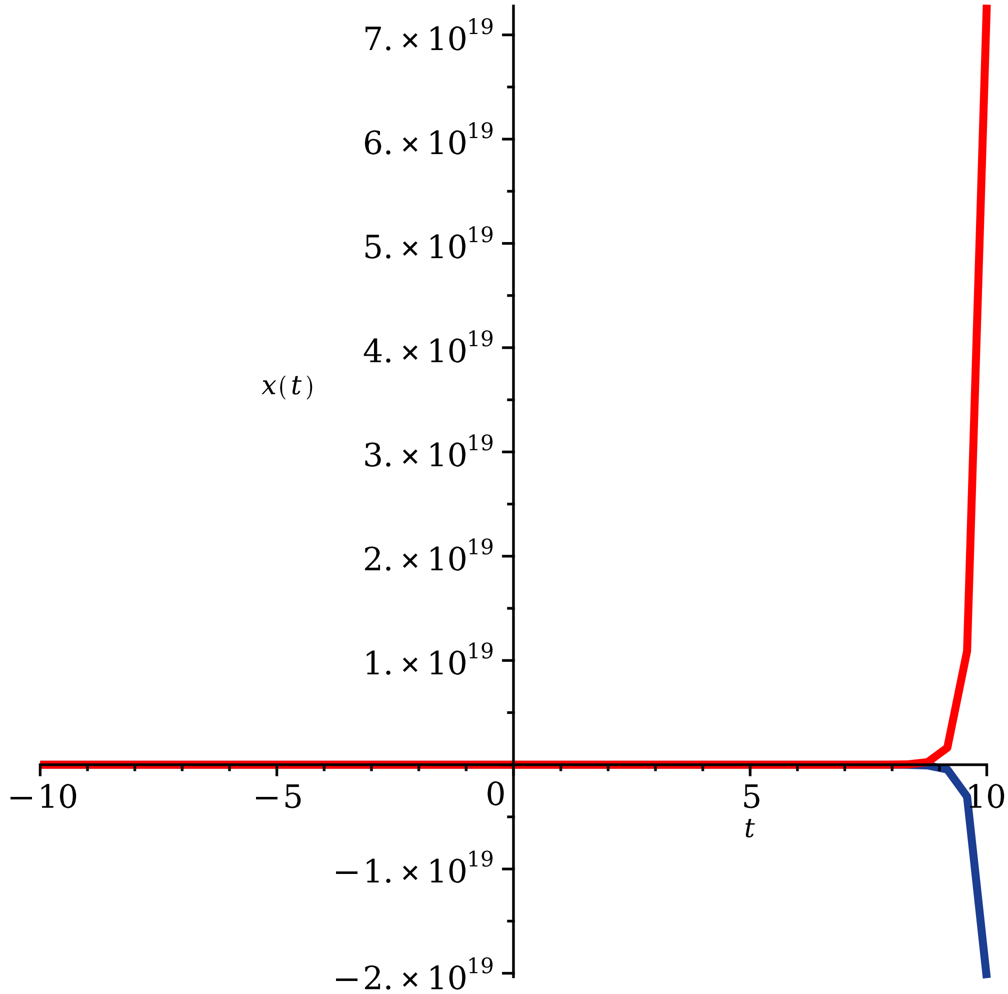
DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]-Y[2]+X-1; YP[2] := -2*Y[1]+4*Y[2]+cos(X); 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[0., 1.]
dsolve/numeric/SC/firststep: Checking ODE for hardware floating point computation
dsolve/numeric/SC/firststep: Initial point OK for hardware floating point computation



> display([xx,yy])



> ec1:=diff(x(t),t)=x(t)+2*y(t)+exp(-t)

$$ec1 := \frac{d}{dt}\, x(t) = x(t) + 2\, y(t) + e^{-t} \quad \textbf{(40)}$$

> ec2:=diff(y(t),t)=-2*x(t)+y(t)+1

$$ec2 := \frac{d}{dt}\, y(t) = -2\, x(t) + y(t) + 1 \quad \textbf{(41)}$$

> cond:=x(0)=0,y(0)=1

$$cond := x(0) = 0, y(0) = 1 \quad \textbf{(42)}$$

> sist:=ec1,ec2

$$sist := \frac{d}{dt}\, x(t) = x(t) + 2\, y(t) + e^{-t}, \frac{d}{dt}\, y(t) = -2\, x(t) + y(t) + 1 \quad \textbf{(43)}$$

> sol:=dsolve({sist,cond},{x(t),y(t)})

-> Solving each unknown as a function of the next ones using the order: [x(t), y(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = 2*(diff(y(x) x))-5*y(x)-1-2*exp(-x) y(x) explicit singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature

trying high order exact linear fully integrable
trying differential order: 2; linear nonhomogeneous with symmetry [0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
   checking if the LODE has constant coefficients
   <- constant coefficients successful
   -> Determining now a particular solution to the non-homogeneous ODE
      building a particular solution using variation of parameters
<- solving first the homogeneous part of the ODE successful

$$sol := \left\{x(t) = -\frac{3\,\mathrm{e}^{t}\cos(2\,t)}{20} + \frac{29\,\mathrm{e}^{t}\sin(2\,t)}{20} - \frac{\mathrm{e}^{-t}}{4} + \frac{2}{5}, y(t) = \frac{3\,\mathrm{e}^{t}\sin(2\,t)}{20} + \frac{29\,\mathrm{e}^{t}\cos(2\,t)}{20} - \frac{1}{5} - \frac{\mathrm{e}^{-t}}{4}\right\} \quad \textbf{(44)}$$

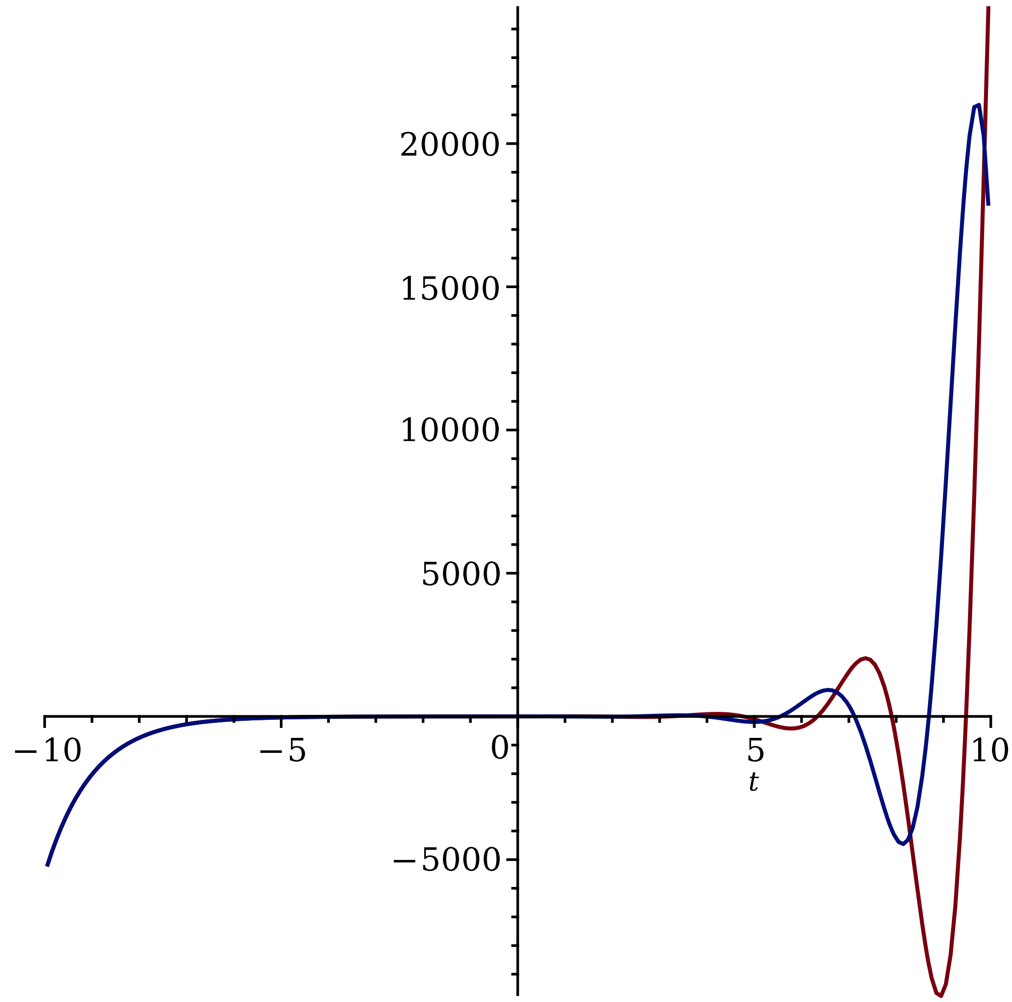
```
> xx:=unapply(rhs(sol[1]),t)
```

$$xx := t \mapsto -\frac{3\cdot\mathrm{e}^{t}\cdot\cos(2\cdot t)}{20} + \frac{29\cdot\mathrm{e}^{t}\cdot\sin(2\cdot t)}{20} - \frac{\mathrm{e}^{-t}}{4} + \frac{2}{5} \quad \textbf{(45)}$$

```
> yy:=unapply(rhs(sol[2]),t)
```

$$yy := t \mapsto \frac{3\cdot\mathrm{e}^{t}\cdot\sin(2\cdot t)}{20} + \frac{29\cdot\mathrm{e}^{t}\cdot\cos(2\cdot t)}{20} - \frac{1}{5} - \frac{\mathrm{e}^{-t}}{4} \quad \textbf{(46)}$$

```
> plot([xx(t),yy(t)],t=-10..10)
```




```
> ec1:=diff(x(t),t)=-x(t)+3*y(t)+3*z(t)+27*t^2
```

$$ec1 := \frac{d}{dt}\, x(t) = -x(t) + 3\,y(t) + 3\,z(t) + 27\,t^2 \tag{47}$$

```
> ec2:=diff(y(t),t)=2*x(t)-2*y(t)-5*z(t)+3*t
```

$$ec2 := \frac{d}{dt}\, y(t) = 2\,x(t) - 2\,y(t) - 5\,z(t) + 3\,t \tag{48}$$

```
> ec3:=diff(z(t),t)=-2*x(t)+3*y(t)+6*z(t)+3
```

$$ec3 := \frac{d}{dt}\, z(t) = -2\,x(t) + 3\,y(t) + 6\,z(t) + 3 \tag{49}$$

```
> cond:=x(0)=50,y(0)=-30,z(0)=26
```

$$cond := x(0) = 50, y(0) = -30, z(0) = 26 \tag{50}$$

```
> sist:=ec1,ec2,ec3
```

$$sist := \frac{d}{dt}\, x(t) = -x(t) + 3\,y(t) + 3\,z(t) + 27\,t^2, \frac{d}{dt}\, y(t) = 2\,x(t) - 2\,y(t) - 5\,z(t) + 3\,t, \frac{d}{dt}\, z(t) = -2\,x(t) + 3\,y(t) + 6\,z(t) + 3 \tag{51}$$

```
> sol:=dsolve({sist,cond},{x(t),y(t),z(t)})
-> Solving each unknown as a function of the next ones using the
```

```
order: [z(t), y(t), x(t)]
-> Calling odsolve with the ODE diff(y(x) x) = 45*x^2-(5/3)*
(diff(x(x) x))+(1/3)*x(x)+3*y(x)+3*x y(x) explicit singsol = none
Methods for first order ODEs:
--- Trying classification methods ---
trying a quadrature
trying 1st order linear
<- 1st order linear successful
-> Calling odsolve with the ODE diff(diff(y(x) x) x) = -27*x^2+y
(x)+63*x+9 y(x) explicit singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
trying high order exact linear fully integrable
trying differential order: 2; linear nonhomogeneous with
symmetry [0,1]
trying a double symmetry of the form [xi=0, eta=F(x)]
-> Try solving first the homogeneous part of the ODE
   checking if the LODE has constant coefficients
   <- constant coefficients successful
   -> Determining now a particular solution to the non-
homogeneous ODE
      trying a rational particular solution
      <- rational particular solution successful
<- solving first the homogeneous part of the ODE successful
```

$$sol := \{x(t) = 3\,\mathrm{e}^{t} + 2\,\mathrm{e}^{-t} + 27\,t^2 - 63\,t + 45, y(t) = \mathrm{e}^{3\,t} + 2\,\mathrm{e}^{t} - 18\,t^2 + 24\,t - 32 - \mathrm{e}^{-t}, z(t) = -\mathrm{e}^{3\,t} - 27\,t + 26 + 18\,t^2 + \mathrm{e}^{-t}\} \quad \textbf{(52)}$$

```
> xx:=unapply(rhs(sol[1]),t)
```

$$xx := t \mapsto 3\cdot\mathrm{e}^{t} + 2\cdot\mathrm{e}^{-t} + 27\cdot t^2 - 63\cdot t + 45 \quad \textbf{(53)}$$

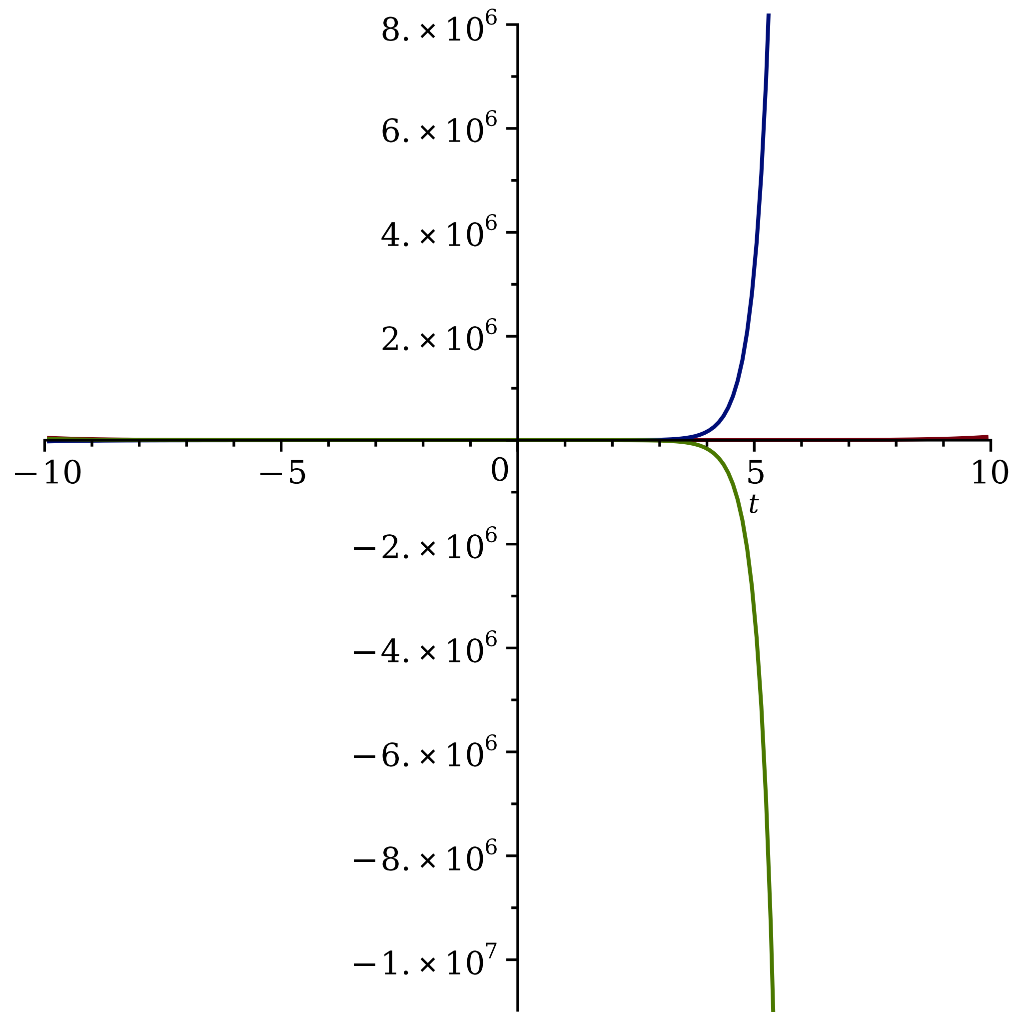
```
> yy:=unapply(rhs(sol[2]),t)
```

$$yy := t \mapsto \mathrm{e}^{3\cdot t} + 2\cdot\mathrm{e}^{t} - 18\cdot t^2 + 24\cdot t - 32 - \mathrm{e}^{-t} \quad \textbf{(54)}$$

```
> zz:=unapply(rhs(sol[3]),t)
```

$$zz := t \mapsto -\mathrm{e}^{3\cdot t} - 27\cdot t + 26 + 18\cdot t^2 + \mathrm{e}^{-t} \quad \textbf{(55)}$$

```
> plot([xx(t),yy(t),zz(t)],t=-10..10)
```




```
> restart
> with (DEtools), with (plots)
```

[*AreSimilar, Closure, DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, Desingularize, FindODE, FunctionDecomposition, GCRD, Gosper, Heunsols, Homomorphisms, IVPsol, IsHyperexponential, LCLM, MeijerGsols, MultiplicativeDecomposition, ODEInvariants, PDEchangecoords, PolynomialNormalForm, RationalCanonicalForm, ReduceHyperexp, RiemannPsols, Xchange, Xcommutator, Xgauge, Zeilberger, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diff_table, diffop2de, dperiodic_sols, dpolyform, dsubs, eigenring, endomorphism_charpoly, equinv, eta_k, eulersols, exactsol, expsols, exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic, genhomosol, gensys, hamilton_eqs, hypergeometricsols, hypergeomsols, hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor,* **(56)**

invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, rational_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom], [*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

```
> ec1:=diff(x(t),t)=x(t)+y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) + y(t) \tag{57}$$

```
> ec2:=diff(y(t),t)=-2*x(t)+4*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -2\,x(t) + 4\,y(t) \tag{58}$$

```
> cond1:=x(0)=3,y(0)=0
```

$$cond1 := x(0) = 3,\ y(0) = 0 \tag{59}$$

```
> cond2:=x(0)=0,y(0)=3
```

$$cond2 := x(0) = 0,\ y(0) = 3 \tag{60}$$

```
> cond3:=x(0)=-3,y(0)=0
```

$$cond3 := x(0) = -3,\ y(0) = 0 \tag{61}$$

```
> cond4:=x(0)=0,y(0)=-3
```

$$cond4 := x(0) = 0,\ y(0) = -3 \tag{62}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) + y(t),\ \frac{d}{dt}\, y(t) = -2\,x(t) + 4\,y(t) \tag{63}$$

```
> infolevel[dsolve]:=3
```

(64)

$$infolevel_{dsolve} := 3 \tag{64}$$

```
> sol1:=dsolve({sist, cond1}, {x(t), y(t)})
```

-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]

$$sol1 := \{x(t) = 6\,e^{2\,t} - 3\,e^{3\,t},\, y(t) = 6\,e^{2\,t} - 6\,e^{3\,t}\} \tag{65}$$

```
> sol2:=dsolve({sist,cond2},{x(t),y(t)})
```

-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]

$$sol2 := \{x(t) = -3\,e^{2\,t} + 3\,e^{3\,t},\, y(t) = -3\,e^{2\,t} + 6\,e^{3\,t}\} \tag{66}$$

```
> sol3:=dsolve({sist,cond3},{x(t),y(t)})
```

-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]

$$sol3 := \{x(t) = -6\,e^{2\,t} + 3\,e^{3\,t},\, y(t) = -6\,e^{2\,t} + 6\,e^{3\,t}\} \tag{67}$$

```
> sol4:=dsolve({sist,cond4},{x(t),y(t)})
```

-> Solving each unknown as a function of the next ones using the order: [y(t), x(t)]

$$sol4 := \{x(t) = 3\,e^{2\,t} - 3\,e^{3\,t},\, y(t) = 3\,e^{2\,t} - 6\,e^{3\,t}\} \tag{68}$$

```
> limit(sol1[1],t=infinity)
```

$$\lim_{t \to \infty} x(t) = -\infty \tag{69}$$

```
> limit(sol1[2],t=infinity)
```

$$\lim_{t \to \infty} y(t) = -\infty \tag{70}$$

```
> limit(sol2[1],t=infinity)
```

$$\lim_{t \to \infty} x(t) = \infty \tag{71}$$

```
> limit(sol2[2],t=infinity)
```

$$\lim_{t \to \infty} y(t) = \infty \tag{72}$$

```
> limit(sol3[1],t=infinity)
```

$$\lim_{t \to \infty} x(t) = \infty \tag{73}$$

```
> limit(sol3[2],t=infinity)
```

$$\lim_{t \to \infty} y(t) = \infty \tag{74}$$

```
> limit(sol4[1],t=infinity)
```

$$\lim_{t \to \infty} x(t) = -\infty \tag{75}$$

```
> limit(sol4[2],t=infinity)
```

$$\lim_{t \to \infty} y(t) = -\infty \tag{76}$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10,[[cond1],
  [cond2],[cond3],[cond4]])
```

dsolve/numeric: entering dsolve/numeric

DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x)) namely (with Y' represented by YP)

$$[YP_1 = Y_1 + Y_2,\, YP_2 = -2\,Y_1 + 4\,Y_2]$$

DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]+Y[2]; YP[2] := -2*Y[1]+4*Y[2]; 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[3., 0.]
dsolve/numeric: entering dsolve/numeric
DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x))
namely (with Y' represented by YP)

$$[YP_1 = Y_1 + Y_2, YP_2 = -2\,Y_1 + 4\,Y_2]$$

DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]+Y[2]; YP[2] := -2*Y[1]+4*Y[2]; 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[0., 3.]
dsolve/numeric: entering dsolve/numeric
DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x))
namely (with Y' represented by YP)

$$[YP_1 = Y_1 + Y_2, YP_2 = -2\,Y_1 + 4\,Y_2]$$

DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

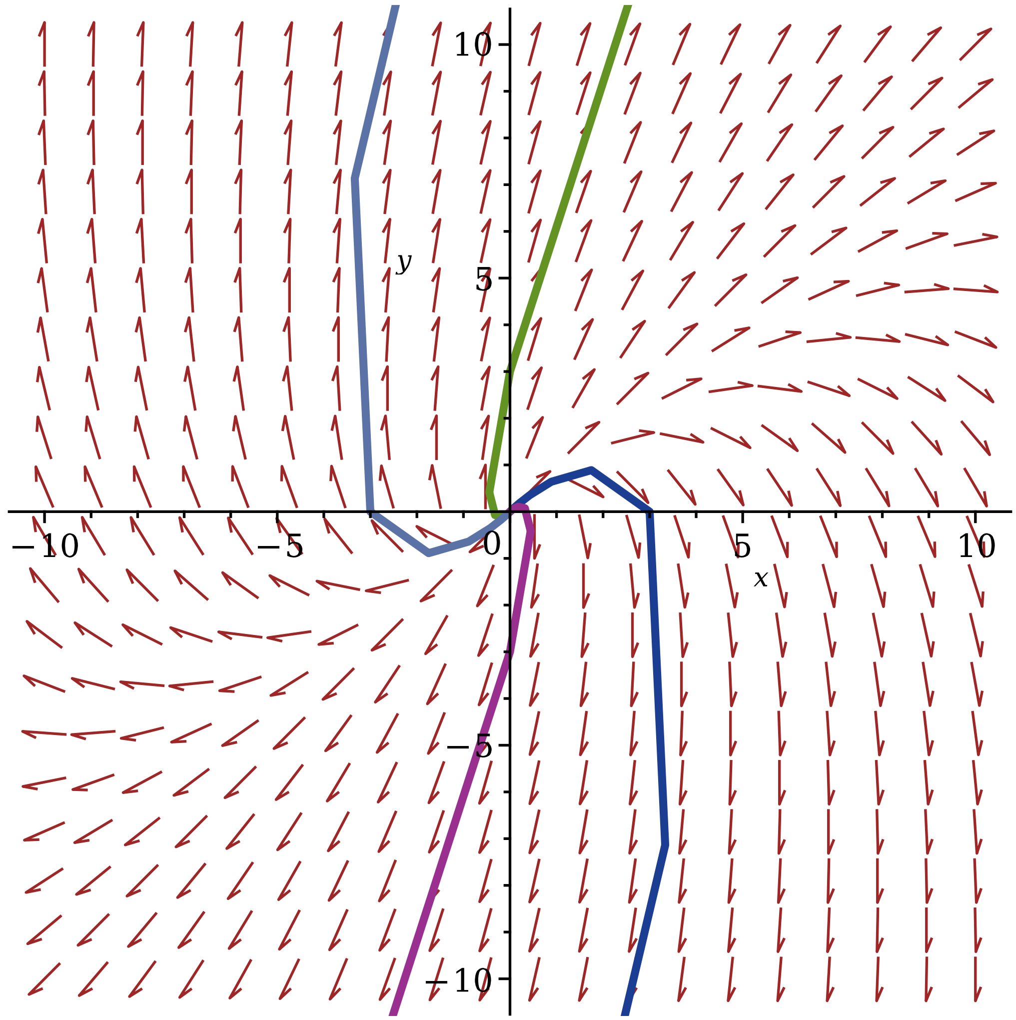
dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]+Y[2]; YP[2] := -2*Y[1]+4*Y[2]; 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[-3., 0.]
dsolve/numeric: entering dsolve/numeric
DEtools/convertsys: converted to first-order system Y'(x) = f(x,Y(x))
namely (with Y' represented by YP)

$$[YP_1 = Y_1 + Y_2, YP_2 = -2\,Y_1 + 4\,Y_2]$$

DEtools/convertsys: correspondence between Y[i] names and original functions:

$$[Y_1 = x(t), Y_2 = y(t)]$$

dsolve/numeric: the procedure F(x,Y,YP) for computing Y'(x)=f(x,Y(x)) is: proc (N, X, Y, YP) option [Y[1] = x(t), Y[2] = y(t)]; YP[1] := Y[1]+Y[2]; YP[2] := -2*Y[1]+4*Y[2]; 0 end proc
dsolve/numeric: initial conditions: x0=0., y0=[0., -3.]




```
> restart
> with (DEtools), with (plots)
```

[AreSimilar, Closure, DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, Desingularize, FindODE, FunctionDecomposition, GCRD, Gosper, Heunsols, Homomorphisms, IVPsol, IsHyperexponential, LCLM, MeijerGsols, MultiplicativeDecomposition, ODEInvariants, PDEchangecoords, PolynomialNormalForm, RationalCanonicalForm, ReduceHyperexp, RiemannPsols, Xchange, Xcommutator, Xgauge, Zeilberger, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diff_table, diffop2de, dperiodic_sols, dpolyform, dsubs, eigenring, endomorphism_charpoly, equinv, eta_k, eulersols, exactsol, expsols, exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic, genhomosol, gensys, hamilton_eqs, hypergeometricsols, hypergeomsols, hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor, **(77)**

invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, rational_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom], [*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

```
> ec1:=diff(x(t),t)=y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = y(t) \qquad (78)$$

```
> ec2:=diff(y(t),t)=-x(t)-2*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -x(t) - 2\,y(t) \qquad (79)$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = y(t),\ \frac{d}{dt}\, y(t) = -x(t) - 2\,y(t) \qquad (80)$$

```
> sol:=dsolve({sist},{x(t),y(t)})
```

$$sol := \left\{x(t) = \mathrm{e}^{-t}\left(c_2\,t + c_1\right),\ y(t) = -\mathrm{e}^{-t}\left(c_2\,t + c_1 - c_2\right)\right\} \qquad (81)$$

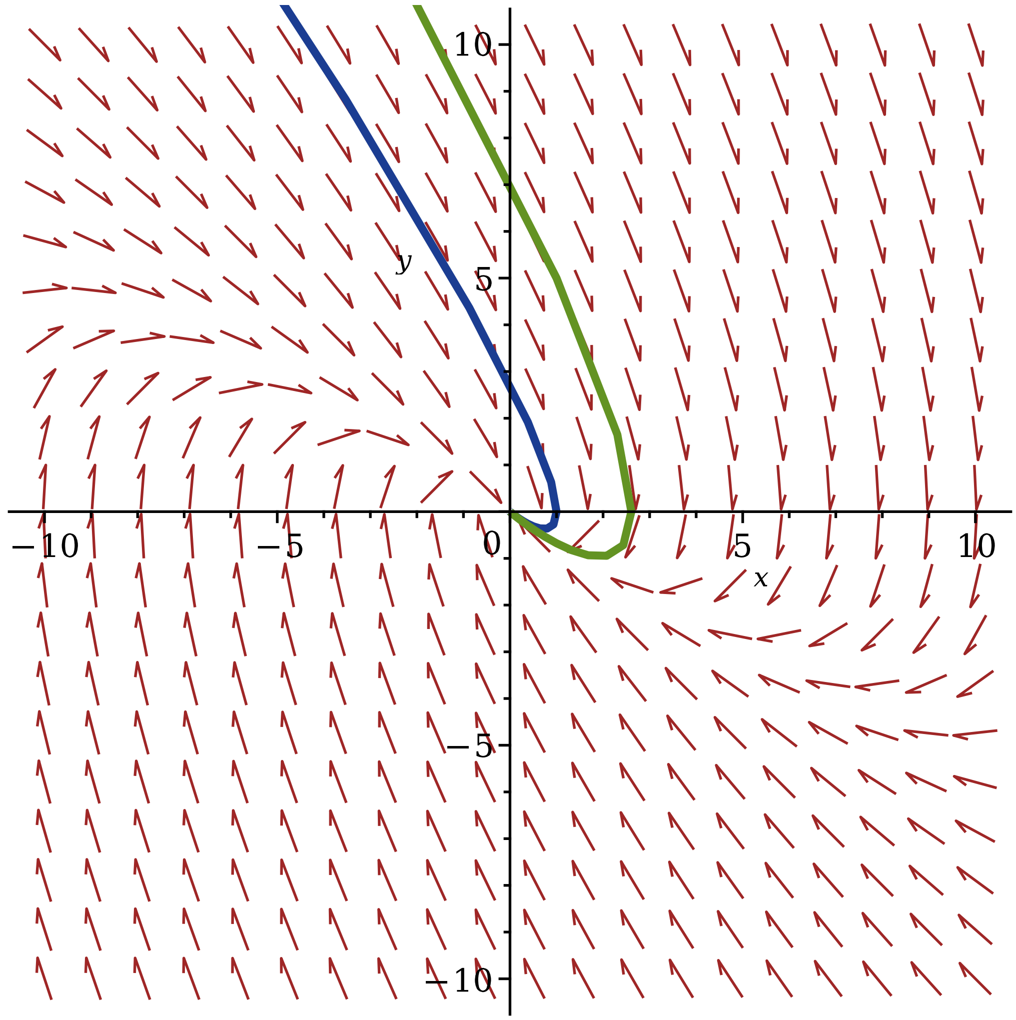
```
> limit(sol[1],t=infinity)
```

$$\lim_{t\to\infty} x(t) = 0 \qquad (82)$$

```
> limit(sol[2],t=infinity)
```

$$\lim_{t\to\infty} y(t) = 0 \qquad (83)$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10,[[x(0)=1,
y(0)=0], [x(0)=1,y(0)=5]])
```




```
> restart
> with (DEtools), with(plots)
```

[*AreSimilar*, *Closure*, *DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *DFactorLCLM*, *DFactorsols*, *Dchangevar*, *Desingularize*, *FindODE*, *FunctionDecomposition*, *GCRD*, *Gosper*, *Heunsols*, *Homomorphisms*, *IVPsol*, *IsHyperexponential*, *LCLM*, *MeijerGsols*, *MultiplicativeDecomposition*, *ODEInvariants*, *PDEchangecoords*, *PolynomialNormalForm*, *RationalCanonicalForm*, *ReduceHyperexp*, *RiemannPsols*, *Xchange*, *Xcommutator*, *Xgauge*, *Zeilberger*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *caseplot*, *casesplit*, *checkrank*, *chinisol*, *clairautsol*, *constcoeffsols*, *convertAlg*, *convertsys*, *dalembertsol*, *dcoeffs*, *de2diffop*, *dfieldplot*, *diff_table*, *diffop2de*, *dperiodic_sols*, *dpolyform*, *dsubs*, *eigenring*, *endomorphism_charpoly*, *equinv*, *eta_k*, *eulersols*, *exactsol*, *expsols*, *exterior_power*, *firint*, *firtest*, *formal_sol*, *gen_exp*, *generate_ic*, *genhomosol*, *gensys*, *hamilton_eqs*, *hypergeometricsols*, *hypergeomsols*, *hyperode*, *indicialeq*, *infgen*, *initialdata*, *integrate_sols*, *intfactor*, **(84)**

invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, ode_int_y, ode_y1, odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, rational_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom], [*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

```
> ec1:=diff(x(t),t)=2*x(t)+y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = 2\, x(t) + y(t) \tag{85}$$

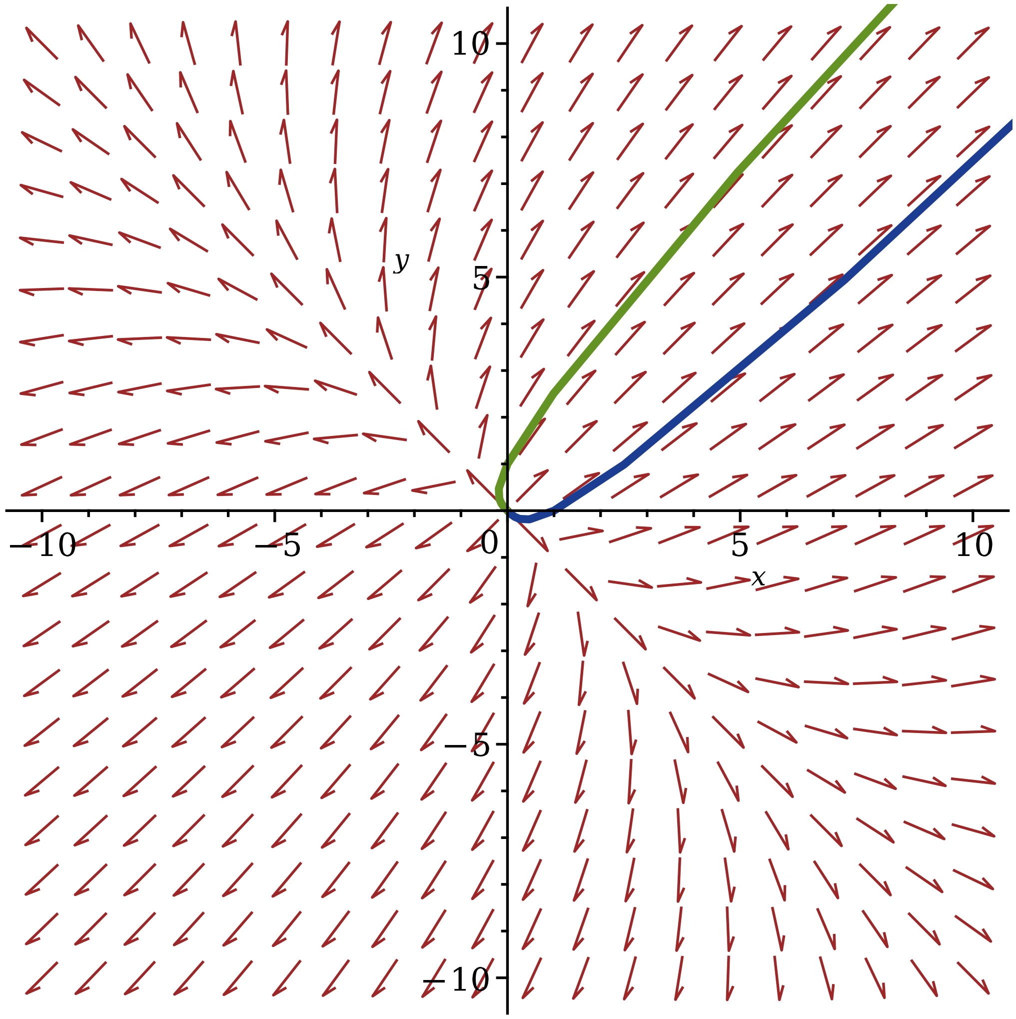
```
> ec2:=diff(y(t),t)=x(t)+2*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) + 2\, y(t) \tag{86}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = 2\, x(t) + y(t),\ \frac{d}{dt}\, y(t) = x(t) + 2\, y(t) \tag{87}$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=
  1,y(0)=0], [x(0)=0,y(0)=1]])
```




```
> #NU
> ec1:=diff(x(t),t)=-x(t)-y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = -x(t) - y(t) \tag{88}$$

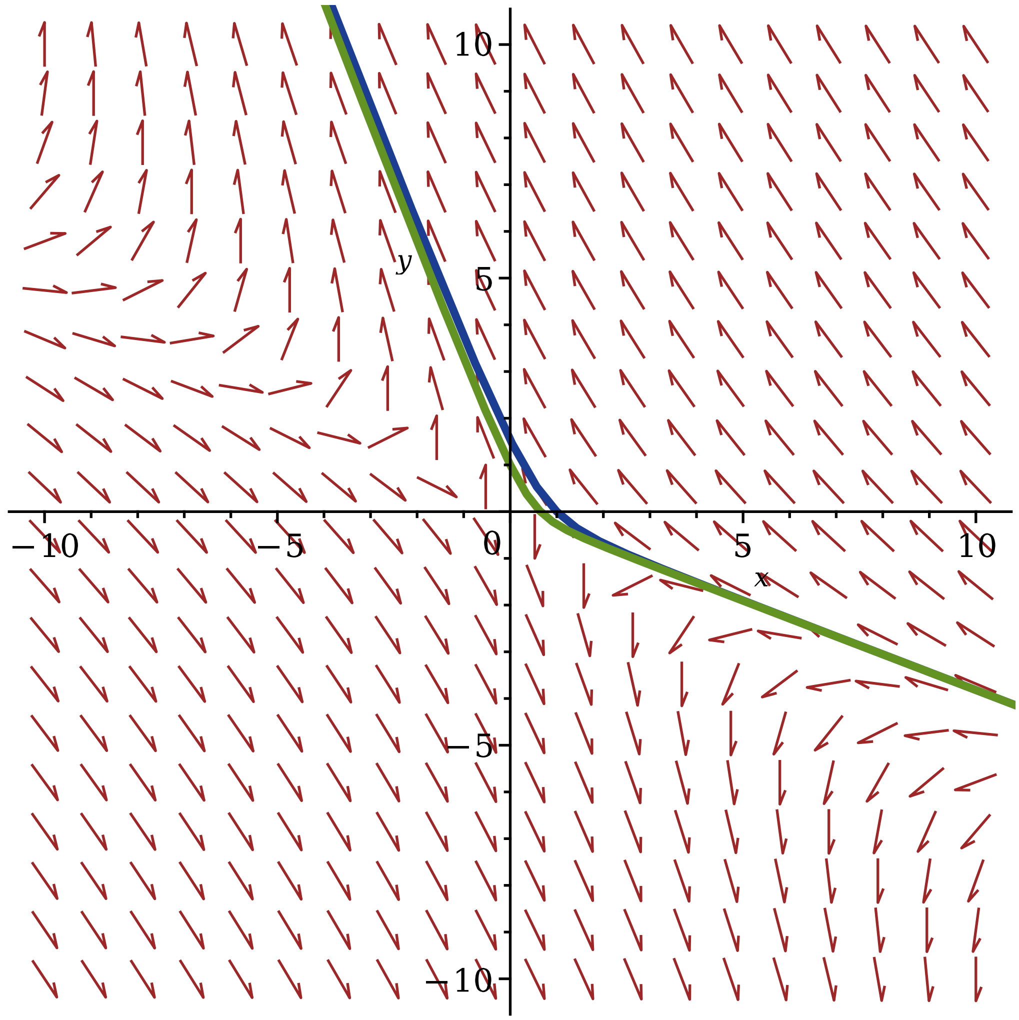
```
> ec2:=diff(y(t),t)=x(t)+2*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = x(t) + 2\, y(t) \tag{89}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = -x(t) - y(t),\ \frac{d}{dt}\, y(t) = x(t) + 2\, y(t) \tag{90}$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=
  1,y(0)=0], [x(0)=0,y(0)=1]])
```



> #NU

> ec1:=diff(x(t),t)=y(t)

$$ec1 := \frac{d}{dt}\, x(t) = y(t) \quad (91)$$

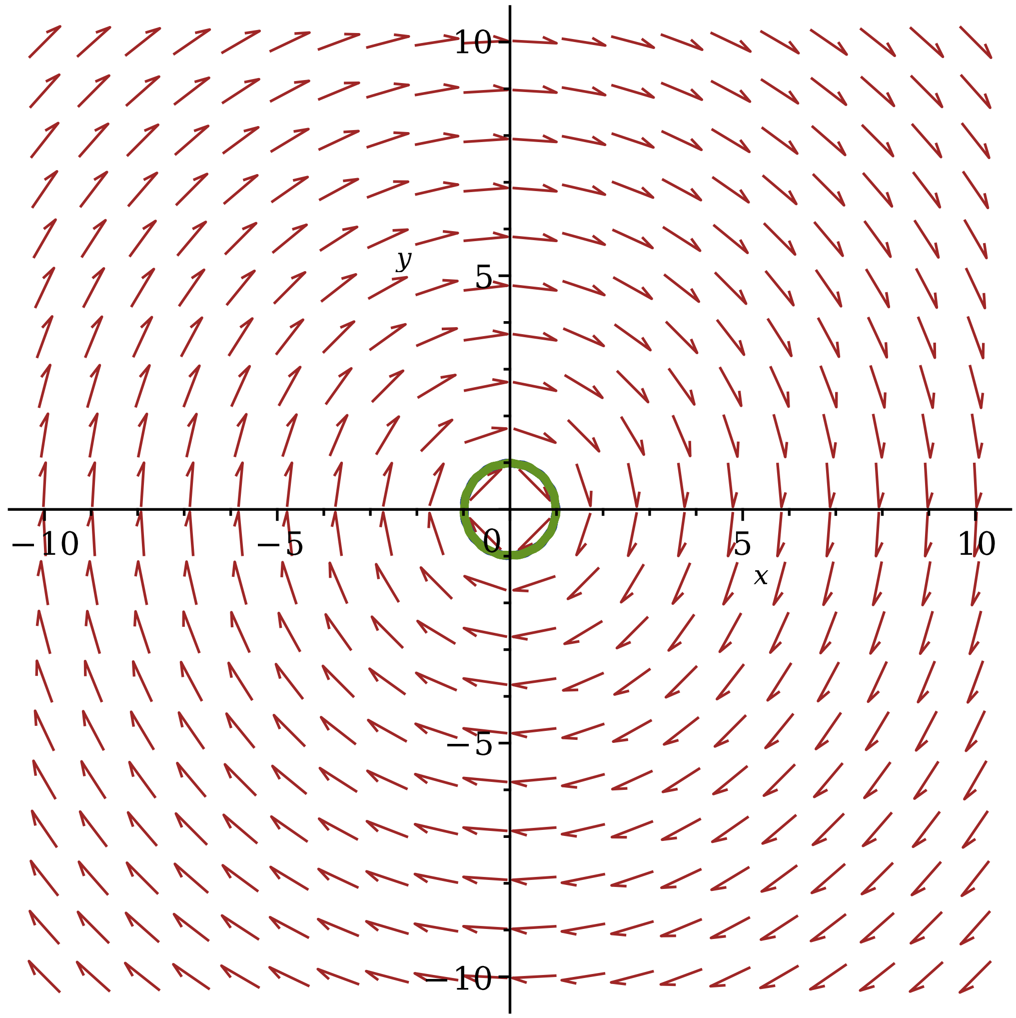
> ec2:=diff(y(t),t)=-x(t)

$$ec2 := \frac{d}{dt}\, y(t) = -x(t) \quad (92)$$

> sist:=ec1,ec2

$$sist := \frac{d}{dt}\, x(t) = y(t),\ \frac{d}{dt}\, y(t) = -x(t) \quad (93)$$

> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=1,y(0)=0], [x(0)=0,y(0)=1]])




```
> #NU
```

```
> ec1:=diff(x(t),t)=-2*x(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = -2\, x(t) \tag{94}$$

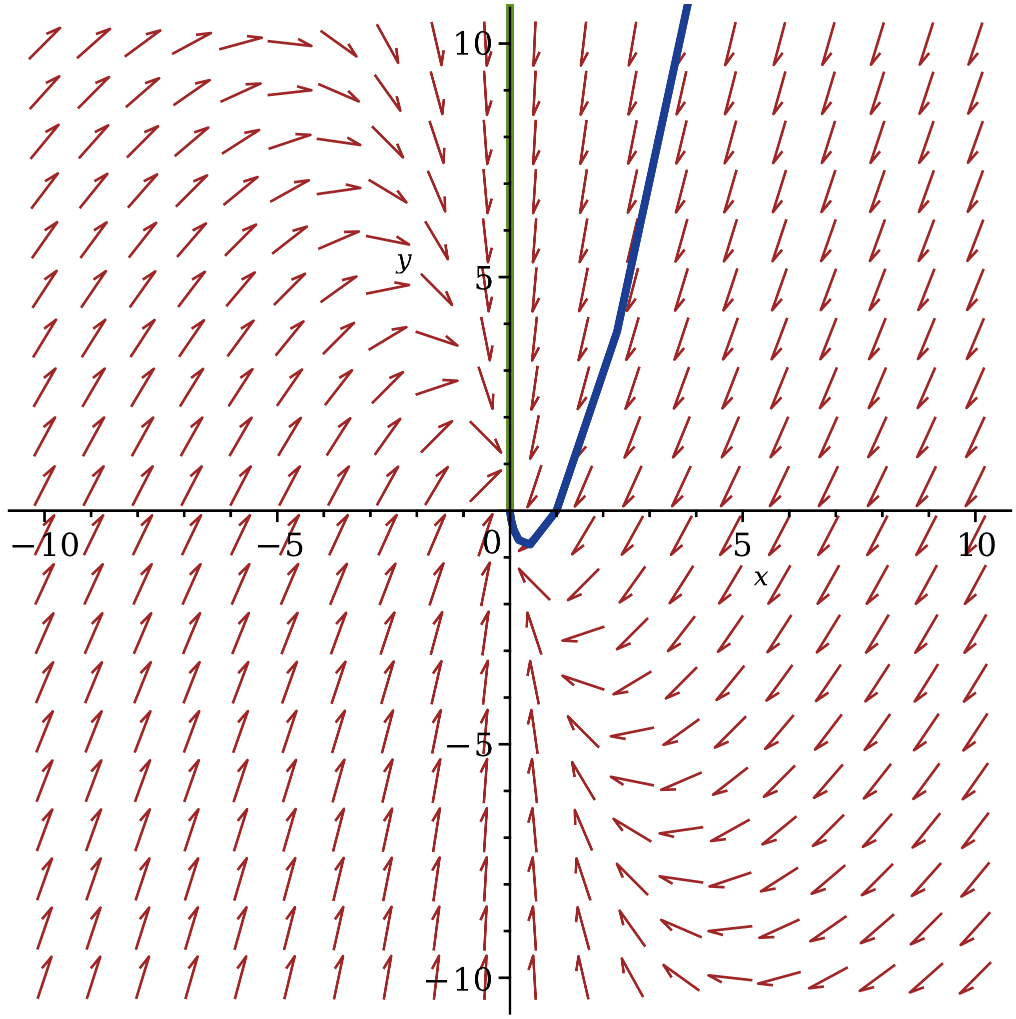
```
> ec2:=diff(y(t),t)=-4*x(t)-2*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = -4\, x(t) - 2\, y(t) \tag{95}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = -2\, x(t),\ \frac{d}{dt}\, y(t) = -4\, x(t) - 2\, y(t) \tag{96}$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=
  1,y(0)=0], [x(0)=0,y(0)=1]])
```




```
> #DA
> ec1:=diff(x(t),t)=x(t)-4*y(t)
```

$$ec1 := \frac{d}{dt}\, x(t) = x(t) - 4\, y(t) \tag{97}$$

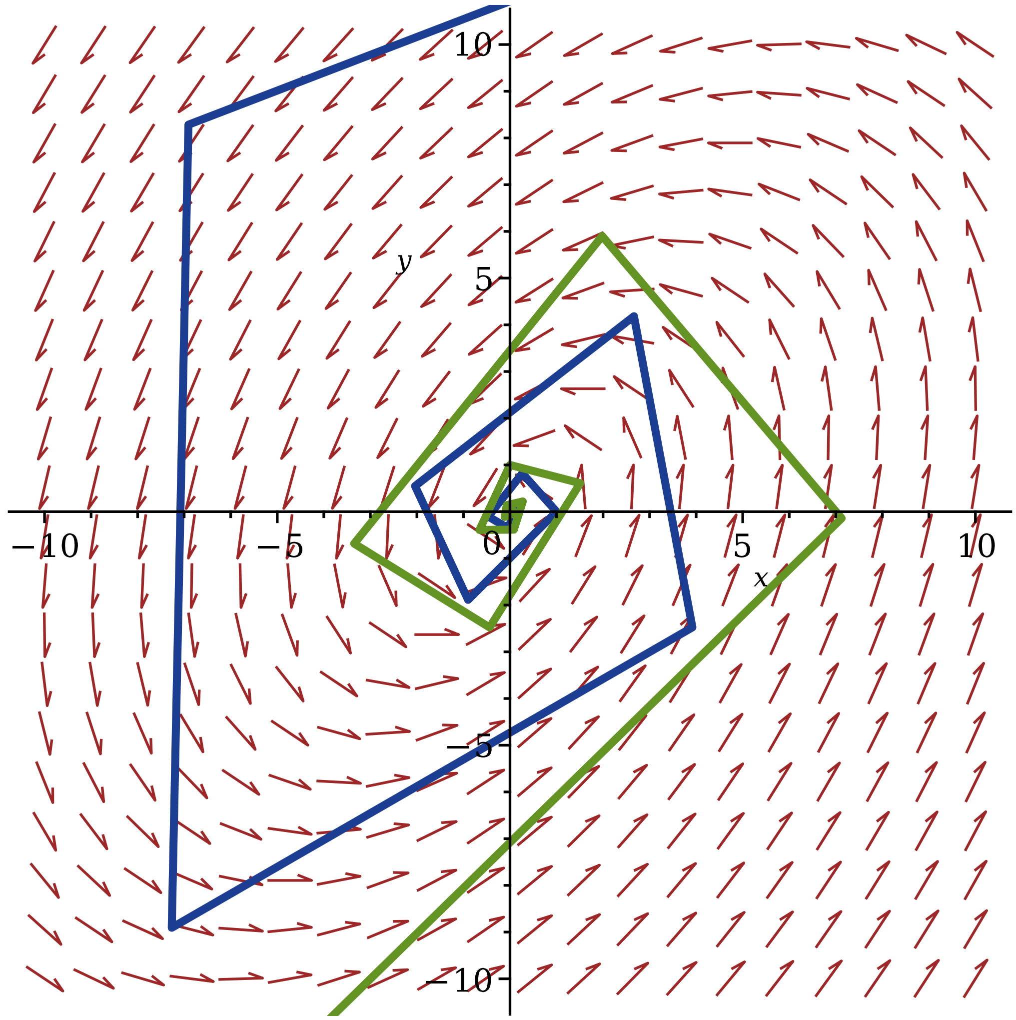
```
> ec2:=diff(y(t),t)=5*x(t)-3*y(t)
```

$$ec2 := \frac{d}{dt}\, y(t) = 5\, x(t) - 3\, y(t) \tag{98}$$

```
> sist:=ec1,ec2
```

$$sist := \frac{d}{dt}\, x(t) = x(t) - 4\, y(t),\ \frac{d}{dt}\, y(t) = 5\, x(t) - 3\, y(t) \tag{99}$$

```
> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=
  1,y(0)=0], [x(0)=0,y(0)=1]])
```



> #DA

> ec1:=diff(x(t),t)=3*x(t)-y(t)

$$ec1 := \frac{d}{dt}\, x(t) = 3\, x(t) - y(t) \qquad (100)$$

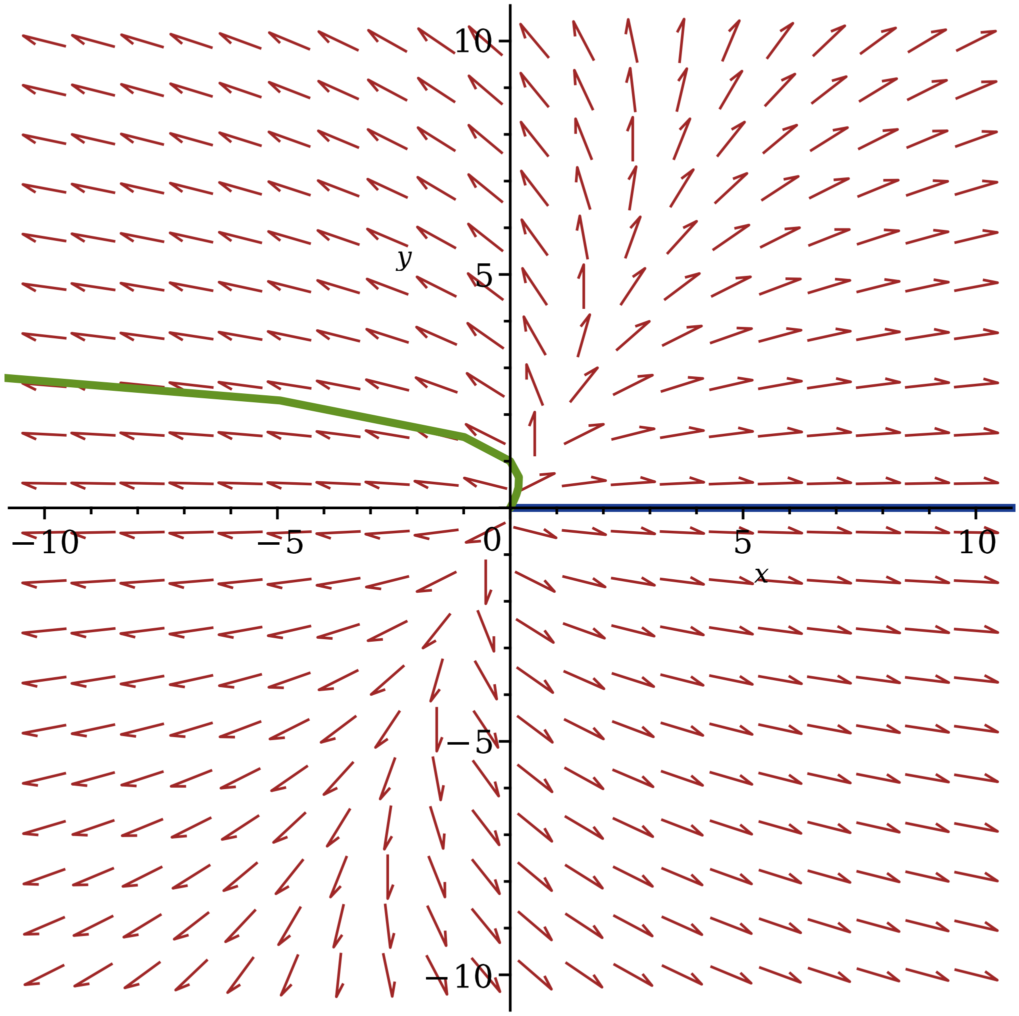
> ec2:=diff(y(t),t)=y(t)

$$ec2 := \frac{d}{dt}\, y(t) = y(t) \qquad (101)$$

> sist:=ec1,ec2

$$sist := \frac{d}{dt}\, x(t) = 3\, x(t) - y(t),\ \frac{d}{dt}\, y(t) = y(t) \qquad (102)$$

> DEplot([sist],[x(t),y(t)],t=-10..10,x=-10..10,y=-10..10, [[x(0)=1,y(0)=0], [x(0)=0,y(0)=1]])



> #NU