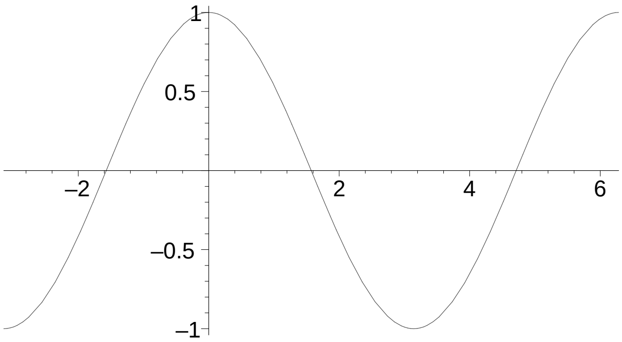
TP 1 : Fonctions usuelles, équations différentielles

```
> restart:
```

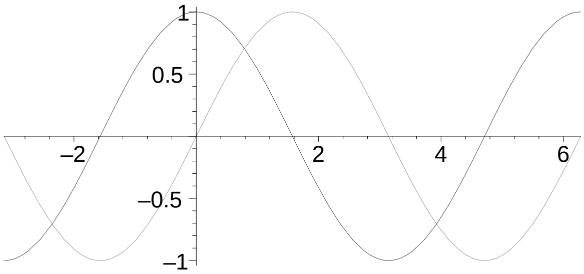
1 Représentation de fonctions

1.1 Représenter

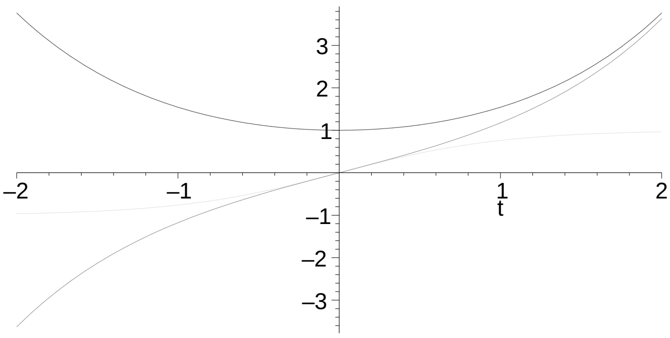
```
> plot(cos,-Pi..2*Pi);
```



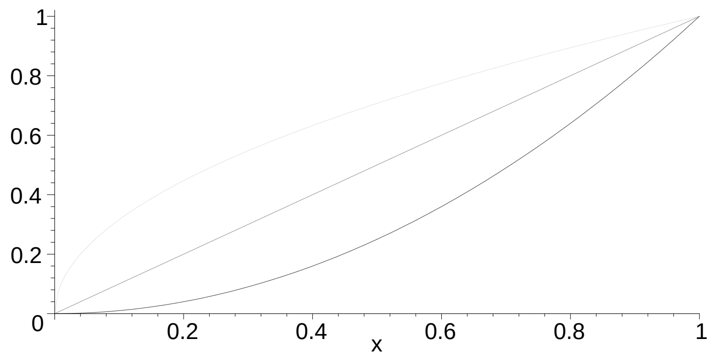

```
> plot({cos,sin},-Pi..2*Pi);
```



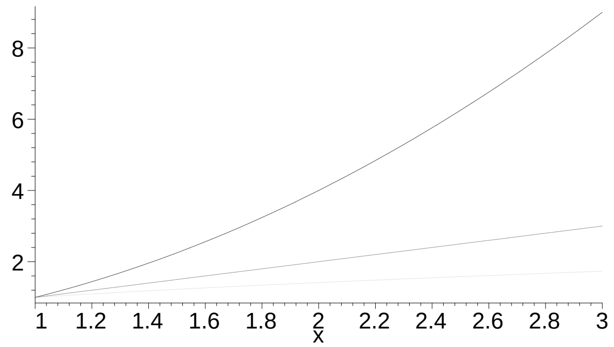

```
> plot({cosh(t),sinh(t),tanh(t)},t=-2..2);
```



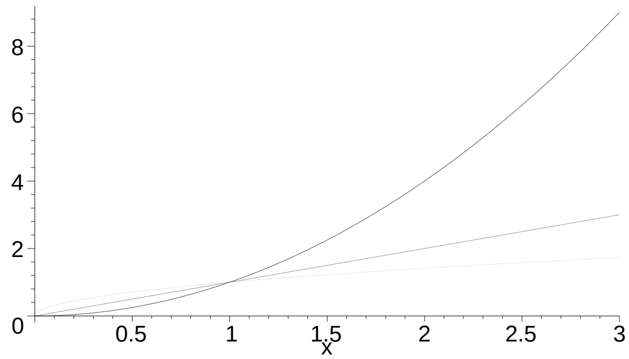

```
> plot({x,x^2,sqrt(x)},x=0..1);
```



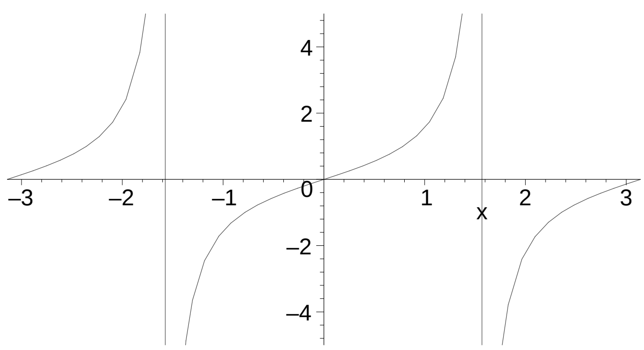

```
> plot({x,x^2,sqrt(x)},x=1..3);
```



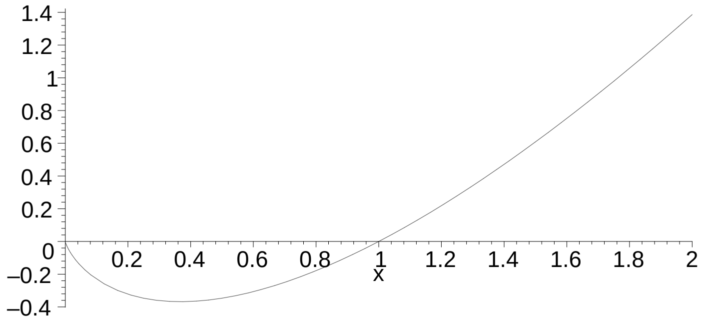

```
> plot({x,x^2,sqrt(x)},x=0..3);
```




```
> plot(tan(x),x=-Pi..Pi,-5..5);
```




```
> plot(x*ln(x),x=0..2);
```




```
> restart;
```

1.2 Définir une fonction

```
> f:=x->exp(1/(x+3))*(x-2);
```

$$f := x \to \mathbf{e}^{\left(\frac{1}{x+3}\right)}(x-2)$$

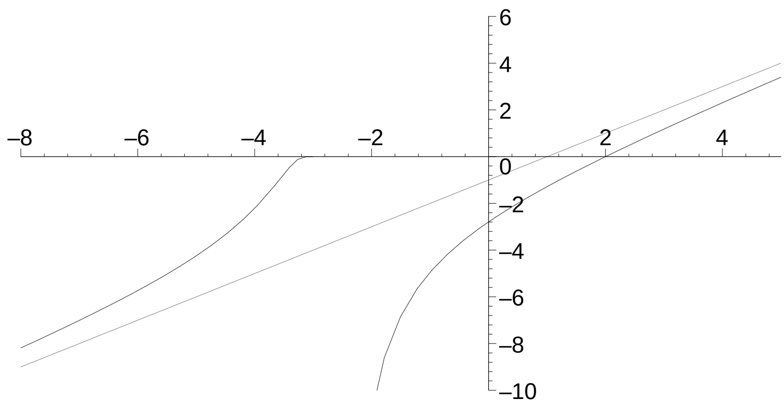
```
> f(x),f(t),f(3);
```

$$\mathbf{e}^{\left(\frac{1}{x+3}\right)}(x-2),\ \mathbf{e}^{\left(\frac{1}{t+3}\right)}(t-2),\ \mathbf{e}^{(1/6)}$$

```
> g:=x->x-1;
```

$$g := x \to x-1$$

```
> plot({f,g},-8..5,-10..6);
```



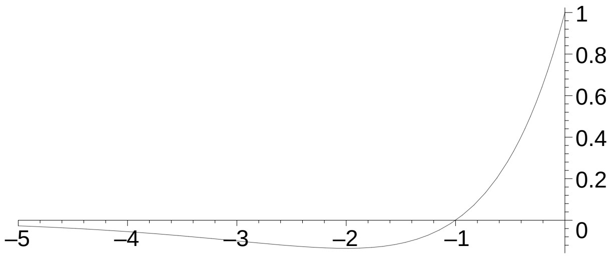

```
> asympt(f(x),x);
```

$$x-1-\frac{9}{2}\frac{1}{x}+\frac{\frac{67}{6}}{x^2}-\frac{655}{24}\frac{1}{x^3}+\frac{\frac{2617}{40}}{x^4}+O\left(\frac{1}{x^5}\right)$$

```
> restart;
```

1.3 Dériver

```
> h:=x->(x+1)*exp(x):
> plot(h,-5..0);
```



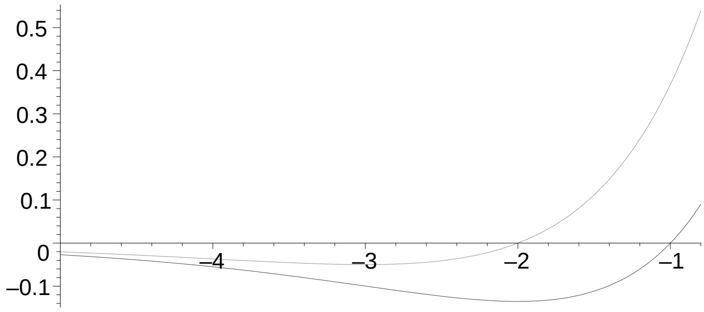

```
> D(h)(x);
```

$$e^x + (x+1)\, e^x$$

```
> D(h)(2),diff(h(x),x)(2),subs(x=2,diff(h(x),x));
```

$$4\, e^2,\ (e^x)(2) + (x(2)+1)\,(e^x)(2),\ 4\, e^2$$

```
> plot({h,D(h)},-5..-.8);
```




```
> restart:
```

2 Des équations différentielles

2.1 Pas trop difficile

```
> restart:dsolve(D(y)(t)=y(t),y(t));
```

$$y(t) = \_C1\, e^t$$

```
> y(t);
```

$$y(t)$$

```
> ?assign
```

- The functions **assign(a, B)** and **assign(a = B)** make the assignment **a := B;** and return **NULL**.

```
> dsolve(D(y)(t)=y(t),y(t));
```

$$y(t) = \_C1\, e^t$$

```
> assign(%);
> y(t);
```

$$\_C1\, e^t$$

```
> y(2);
```

$$y(2)$$

```
> subs(t=2,y(t));
```

$$\_C1\, e^2$$

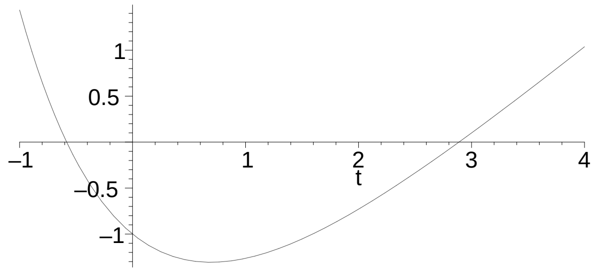
```
> restart;
```

2.2 Avec condition initiale

```
> restart:dsolve({D(y)(t)=-y(t)+t-2,y(0)=-1},y(t));assign(%)
  ;
```

$$y(t) = t - 3 + 2\,\mathbf{e}^{(-t)}$$

```
> plot(y(t),t=-1..4);
```




```
> solve(D(y)(t)=0);
```

$$\mathrm{RootOf}(\mathrm{D}(y)(\_Z))$$

```
> fsolve(D(y)(t)=0);
```

$$\mathrm{fsolve}(\mathrm{D}(y)(t) = 0,\ t)$$

```
> fsolve(D(y)(t)=0,t=0..1);
```

$$\mathrm{fsolve}(\mathrm{D}(y)(t) = 0,\ t,\ 0\ ..\ 1)$$

mouais...

```
> solve(y(t)=0);
```

$$\mathrm{LambertW}(-2\,\mathbf{e}^{(-3)}) + 3,\ \mathrm{LambertW}(-1,\ -2\,\mathbf{e}^{(-3)}) + 3$$

```
> evalf(%);
```

$$2.888703356,\ -.583073876$$

```
> fsolve(y(t)=0);
```

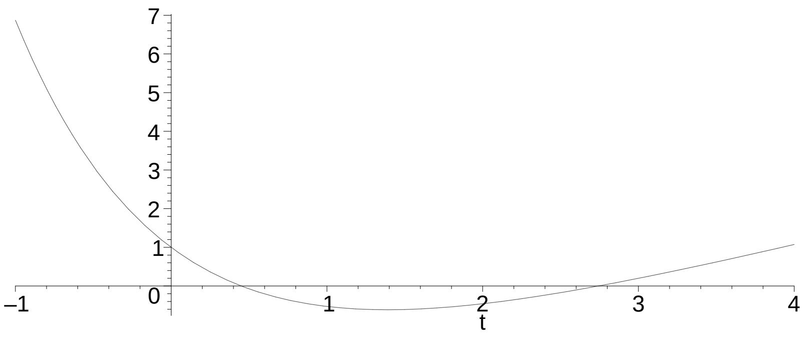
$$2.888703356$$

```
> fsolve(y(t)=0,t=-1..0);
```

$$-.5830738760$$

```
> restart:dsolve({D(y)(t)=-y(t)+t-2,y(0)=1},y(t));assign(%);
  plot(y(t),t=-1..4);
```

$$y(t) = t - 3 + 4\,\mathbf{e}^{(-t)}$$

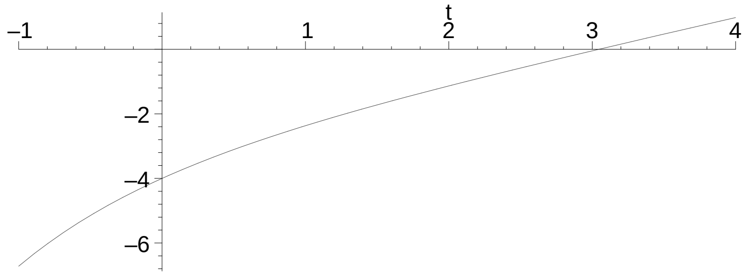



```
> restart:dsolve({D(y)(t)=-y(t)+t-2,y(0)=y0},y(t));
```

$$y(t) = t - 3 + \mathbf{e}^{(-t)}\,(3 + y0)$$

```
> restart:dsolve({D(y)(t)=-y(t)+t-2,y(0)=-4},y(t));assign(%)
  ;plot(y(t),t=-1..4);
```

$$y(t) = t - 3 - \mathbf{e}^{(-t)}$$

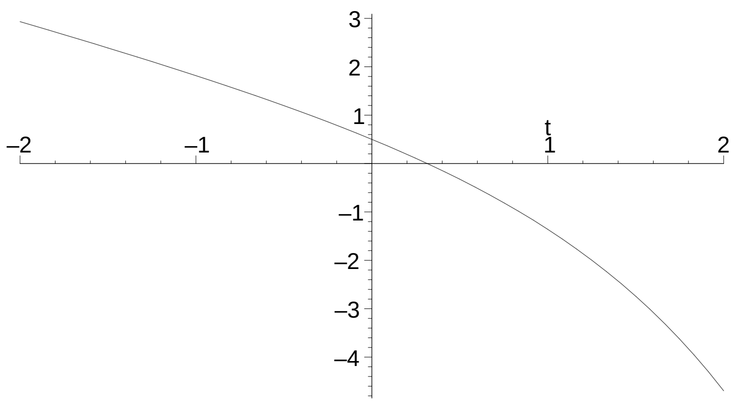



```
> restart:dsolve({D(y)(t)=y(t)+t-2,y(0)=y0},y(t));
```

$$y(t) = -t + 1 + \mathbf{e}^{t}\,(-1 + y0)$$

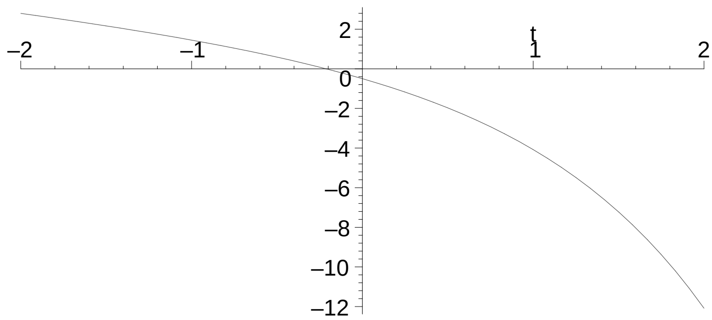
```
> restart:dsolve({D(y)(t)=y(t)+t-2,y(0)=.5},y(t));assign(%):
  plot(y(t),t=-2..2);
```

$$y(t) = -t + 1 - \frac{1}{2}\,\mathbf{e}^{t}$$



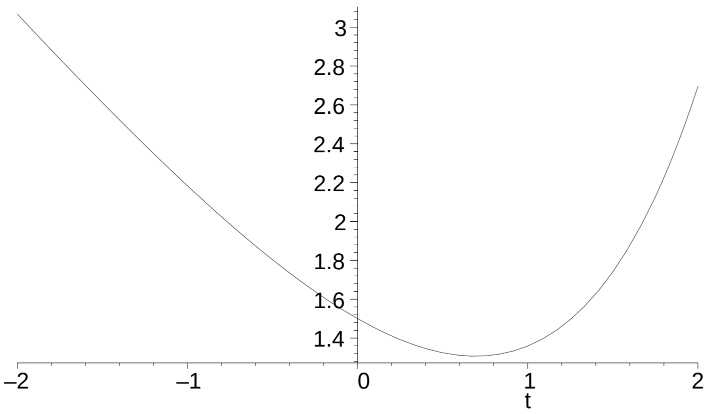

```
> restart:dsolve({D(y)(t)=y(t)+t-2,y(0)=-0.5},y(t));assign(%
  ):plot(y(t),t=-2..2);
```

$$y(t) = -t + 1 - \frac{3}{2}\,\mathbf{e}^{t}$$



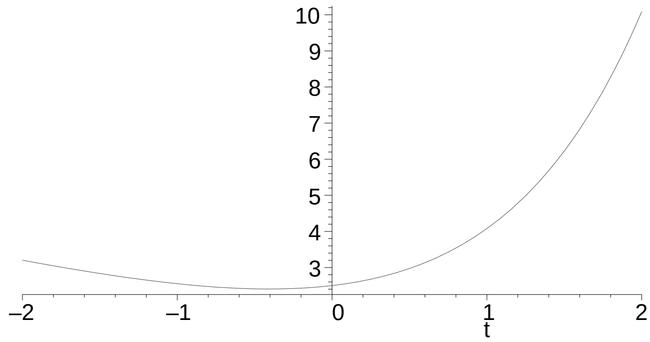

```
> restart:dsolve({D(y)(t)=y(t)+t-2,y(0)=1.5},y(t));assign(%)
  :plot(y(t),t=-2..2);
```

$$y(t) = -t + 1 + \frac{1}{2}\,\mathbf{e}^{t}$$




```
> restart:dsolve({D(y)(t)=y(t)+t-2,y(0)=2.5},y(t));assign(%)
  :plot(y(t),t=-2..2);
```

$$y(t) = -t + 1 + \frac{3}{2}\mathbf{e}^{t}$$




```
> restart;
```

2.3 Un problème non linéaire

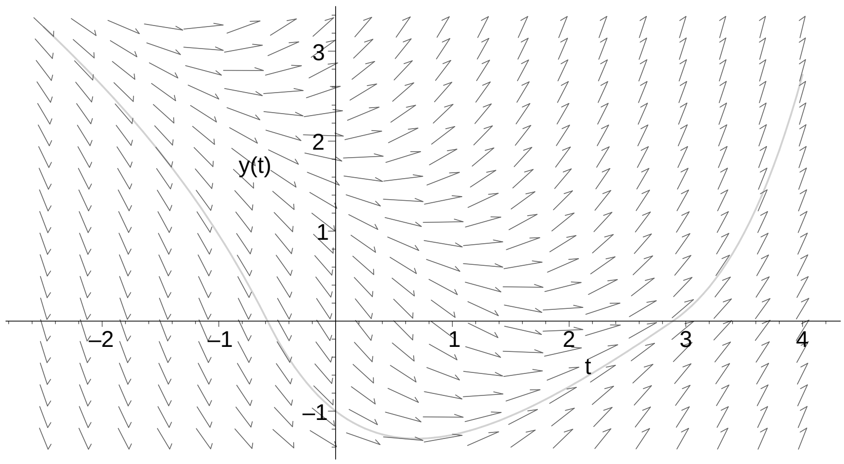
```
> dsolve({D(y)(t)=abs(y(t))+t-2,y(0)=-1},y(t));
> with(DEtools);
```

[*DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, GCRD, LCLM, MeijerGsols, PDEchangecoords, RiemannPsols, Xchange, Xcommutator, Xgauge, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diffop2de, 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, 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, odeadvisor, odepde, parametricsol, phaseportrait, poincare, polysols, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, solve_group, super_reduce,* *symgen*, *symmetric_power*, *symmetric_product*, *symtest*, *transinv*, *translate*, *untranslate*, *varparam*, *zoom*]

```
> ?DEplot
```

- Given a set or list of initial conditions (see below), and a system of first order differential equations or a single higher order differential equation, **DEplot** will plot solution curves, by numerical methods. A two-element system of first order differential equations will also produce a direction field plot, provided the system is determined to be autonomous. For non-autonomous systems, no direction field will be produced (only solution curves will be possible in such instances). There can be ONLY one independent variable.

```
> DEplot(D(y)(t)=abs(y(t))+t-2,y(t),t=-2.5..4,[[y(0)=-1]],st
  epsize=.05);
```




```
> DEplot(D(y)(t)=abs(y(t))+t-2,y(t),t=-5..4.5,[[y(0)=-1]],st
  epsize=.01,arrows=none);
```