1.2 Instruction, exécution et résultat

```
> 2 + 2;
```

$$4 \quad (1.1)$$

```
> 2 + 2
```

$$4 \quad (1.2)$$

```
> 2 + 2 :
> 2 + 3;
  4·5;
```

$$5$$

$$20 \quad (1.3)$$

```
>
```

1.3 Opérations de bases

```
>
```

```
> sqrt(2 + 3·7/5);
```

$$\frac{1}{5}\sqrt{155} \quad (2.1)$$

```
> abs(1 − 2··5);
```

$$31 \quad (2.2)$$

```
> cos(Pi/2);
```

$$0 \quad (2.3)$$

```
> exp(1); ln(1);
```

$$e$$

$$0 \quad (2.4)$$

```
> evalf(exp(1));
```

$$2.718281828 \quad (2.5)$$

```
> ?plot
> exp(1);
```

$$e \quad (2.6)$$

```
>
```

1.4 Les variables

```
> Digits;
```

```
evalf(Pi);
```

$$10$$

$$3.141592654 \quad \textbf{(3.1)}$$

```
> a;
```

$$a \quad \textbf{(3.2)}$$

```
> a := 1; a;
```

$$a := 1$$

$$1 \quad \textbf{(3.3)}$$

```
> a := 1;
  unassign('a');
  a;
```

$$a := 1$$

$$a \quad \textbf{(3.4)}$$

```
> Digits := 50;
  evalf(Pi);
```

$$Digits := 50$$

$$3.1415926535897932384626433832795028841971693993751 \quad \textbf{(3.5)}$$

```
> Digits = 20;
  Digits;
```

$$50 = 20$$

$$50 \quad \textbf{(3.6)}$$

```
> ?unassign
> Pi := 3.14;
Error, attempting to assign to `Pi` which is protected
> a := 10 : b := 43 :
  a; b;
```

$$10$$

$$43 \quad \textbf{(3.7)}$$

```
> restart;
> a; b;
```

$$a$$

$$b \quad \textbf{(3.8)}$$

```
> a := 167;
```

$$a := 167 \quad \textbf{(3.9)}$$

```
> a := a/10;
```

$$a := \frac{167}{100000} \quad \textbf{(3.10)}$$

```
>
```

1.5 Boucles et instructions conditionnelles

1.5.1 Boucles

```
> montant := 0 :
  for i from 1 to 32
  do montant := montant + 10;
  od:
  montant;
```

320 **(4.1.1)**

```
> montant := 0 :
  for i  to 32
  do montant := montant + 10;
  od:
  montant;
```

320 **(4.1.2)**

```
>
```

1.5.2 Tests if/while

```
> a := 10 : b := 0 :
  if (a > 10) then
  print(Coucou);
  else b := evalf(sqrt(a));
  fi:
  a; b;
```

10

3.162277660 **(4.2.1)**

```
> n := 0 :
  while (n·n < 1000)
  do n := n + 1 :
  od:
  n;
```

32 **(4.2.2)**

```
> 32·32;
```

1024 **(4.2.3)**

```
> 31·31;
```

961 **(4.2.4)**

```
> n := 0 :
  while (n·n < 1000)
  do n := n + 1 :
  print(n);
  od:
```

n;

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
32 **(4.2.5)**

>

1.5.3 Opérateurs logiques

```
> a := 11 : b := 5 : c := 0 :
  if ((a > 10) and (b = 7)) then
  c := 40 :
  fi:
  c;
```

$$0 \qquad (4.3.1)$$

```
> a := 11 : b := 5 : c := 0 :
  if ((a > 10) or (b = 7)) then
  c := 40 :
  fi:
  c;
```

$$40 \qquad (4.3.2)$$

```
> a := 10 :
  if (a ≥ 10) then
  print(Coucou) :
  fi:
```

$$Coucou \qquad (4.3.3)$$

```
>
```

1.6 Les fonctions

1.6.1 Fonctions et expressions

```
> f := x→x^2 + 1;
```

$$f := (x, y) \rightarrow x^2 + y + 1 \qquad (5.1.1)$$

```
> f(0.1, 0.5);
```

$$1.51 \qquad (5.1.2)$$

```
> f(0.5);
```

$$1.25 \qquad (5.1.3)$$

```
> restart;
> f := x^2 + 1;
```

$$f := x^2 + 1 \qquad (5.1.4)$$

```
> f(0.1);
```

$$x(0.1)^2 + 1 \qquad (5.1.5)$$

```
> subs(x = 0.1, f);
```

$$1.01 \qquad (5.1.6)$$

```
> f := (x, y)→x^2 + y + 1;
```

$$f := (x, y) \rightarrow x^2 + y + 1 \qquad (5.1.7)$$

```
> f(0.1, 0.5);
```

$$1.51 \qquad \textbf{(5.1.8)}$$

```
>
> restart;
> f := x^2 + y + 1;
```

$$f := x^2 + y + 1 \qquad \textbf{(5.1.9)}$$

$$x\,(0.1)^2 + 1 \qquad \textbf{(5.1.10)}$$

```
> subs(x = 0.1, y = 0.5, f);
```

$$1.51 \qquad \textbf{(5.1.11)}$$

```
> f := x^2 + 1 :
  f := unapply(f, x);
```

$$f := x \rightarrow x^2 + 1 \qquad \textbf{(5.1.12)}$$

```
> restart;
> f := x→x^2 + 1 :
  f := f(x);
```

$$f := x^2 + 1 \qquad \textbf{(5.1.13)}$$

```
>
```

1.6.2 Procédures

```
> toto := proc(f, N)
  local intrec, i;
  intrec := 0;
  for i from 1 to N
  do intrec := intrec + evalf(subs(x = i/N, f))/N;
  od:
  intrec :
  end proc:
> f := x^2 + 1 : N := 100 :
  toto(f, N); evalf(4/3);
```

$$1.338350000$$

$$1.333333333 \qquad \textbf{(5.2.1)}$$

```
> f := x : N := 10000 :
  toto(f, N);
```

$$0.5000500000 \qquad \textbf{(5.2.2)}$$

```
>
```

1.7 Les tableaux

1.7.1 La commande array

```
> N := 100 : c := array(1..N);
```

$$c := array(1..100, [\,])$$ **(6.1.1)**

```
> c[6] := 1;
```

$$c_6 := 1$$ **(6.1.2)**

```
> c[6];
```

$$1$$ **(6.1.3)**

```
> c[5];
```

$$c_5$$ **(6.1.4)**

```
> deb := -10 : fin := 15 : c := array(deb..fin);
  c[-5];
```

$$c := array(-10..15, [\,])$$

$$c_{-5}$$ **(6.1.5)**

```
> N := 100 :
  c := array(1..N) :
  c[1] := 1/7 :
  for k from 2 to N do
  T := add(c[j], j = 1..k-1) :
  S := add(j·c[j]·c[k-j], j = 1..k-1) :
  c[k] := 6/(5·k+9)·(S + 1/3 - T) :
  od:
> evalf(c[100]);
```

$$2.989229284\ 10^{-10}$$ **(6.1.6)**

```
>
>
```

1.7.2 Les listes

```
> L := [1, 4, 7];
```

$$L := [1, 4, 7]$$ **(6.2.1)**

```
> nops(L);
```

$$3$$ **(6.2.2)**

```
> L := [3, 6, 7];
```

$$L := [3, 6, 7] \tag{6.2.3}$$

```
> L[2];
```

$$6 \tag{6.2.4}$$

```
> op(2, L);
```

$$6 \tag{6.2.5}$$

```
> L := [seq(i·i, i = 1 ..5)];
```

$$L := [1, 4, 9, 16, 25] \tag{6.2.6}$$

```
> N := 10 :
  L := [seq(k/N, k = 1 ..N)];
  f := x→exp(x) :
  fL := map(f, L);
```

$$L := \left[\frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1\right]$$

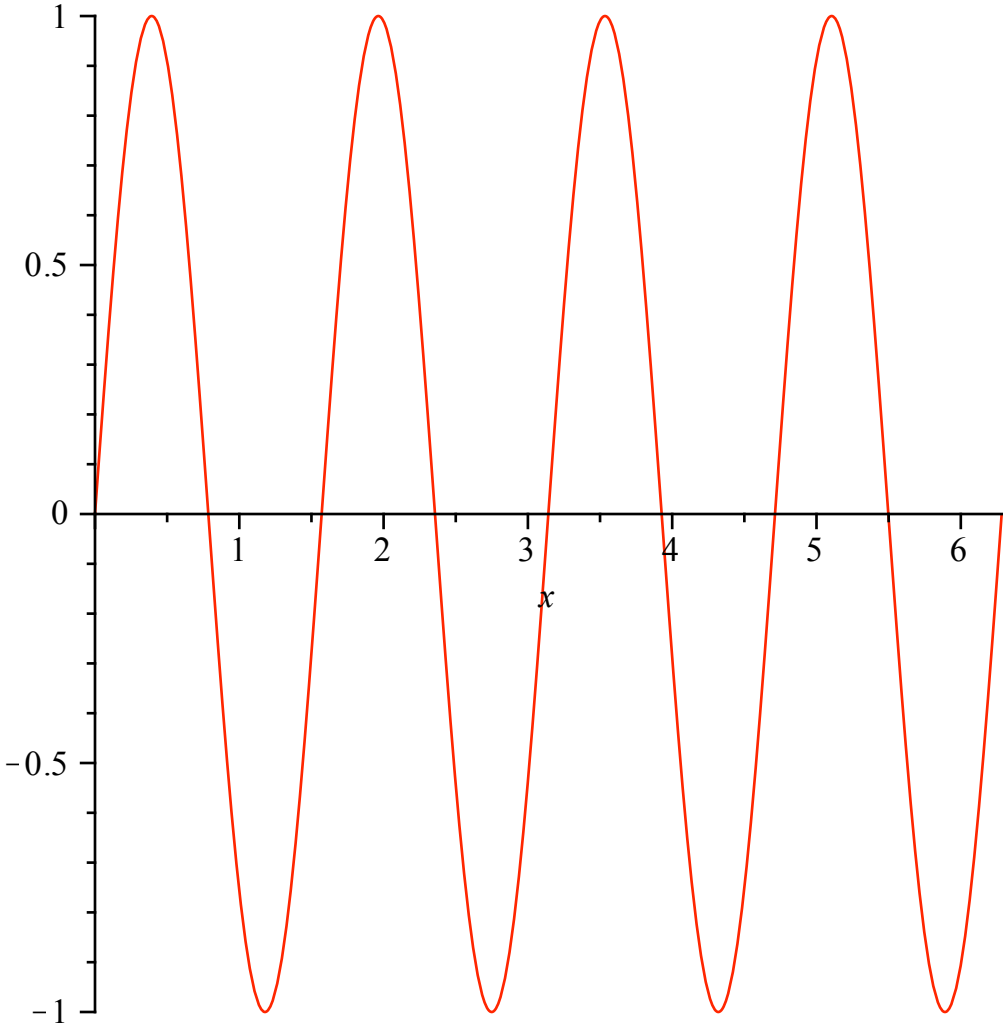
$$fL := \left[e^{\frac{1}{10}}, e^{\frac{1}{5}}, e^{\frac{3}{10}}, e^{\frac{2}{5}}, e^{\frac{1}{2}}, e^{\frac{3}{5}}, e^{\frac{7}{10}}, e^{\frac{4}{5}}, e^{\frac{9}{10}}, e\right] \tag{6.2.7}$$

```
>
```

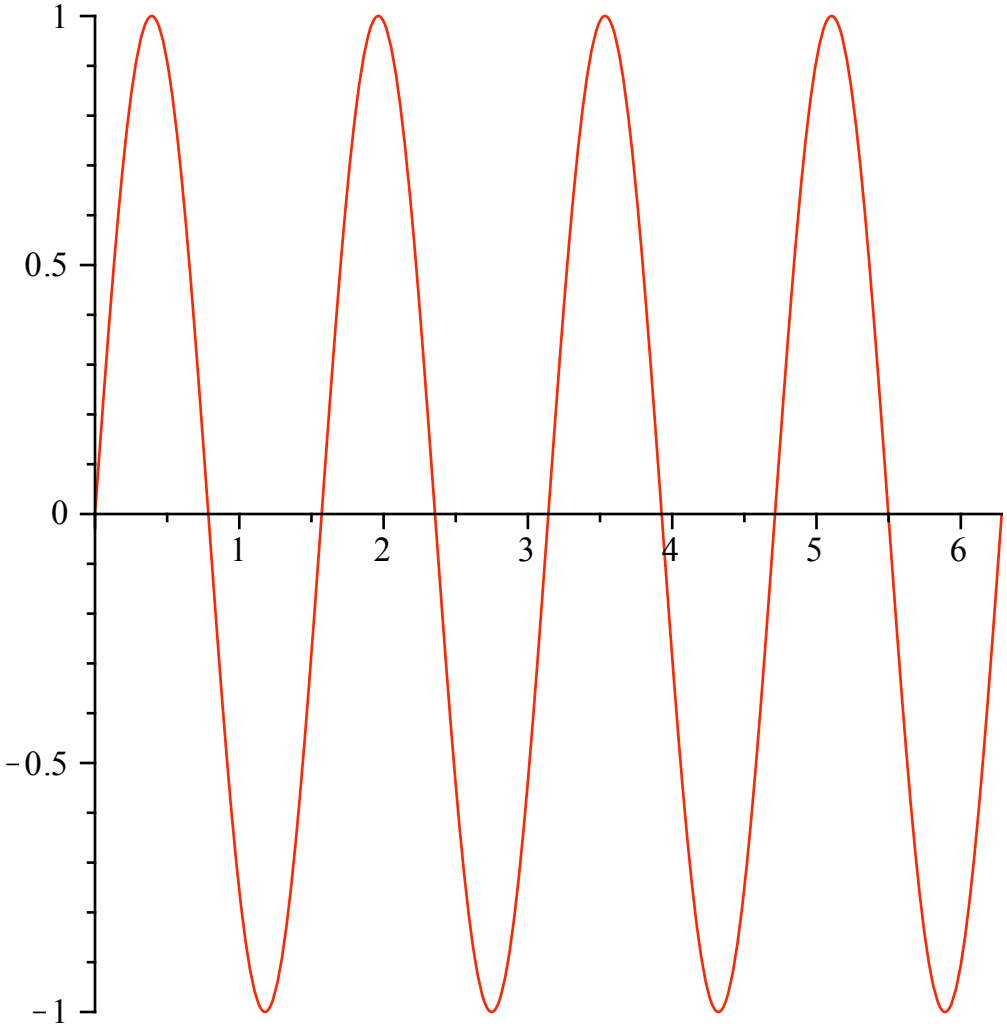
1.8 Graphiques

1.8.1 Des fonctions

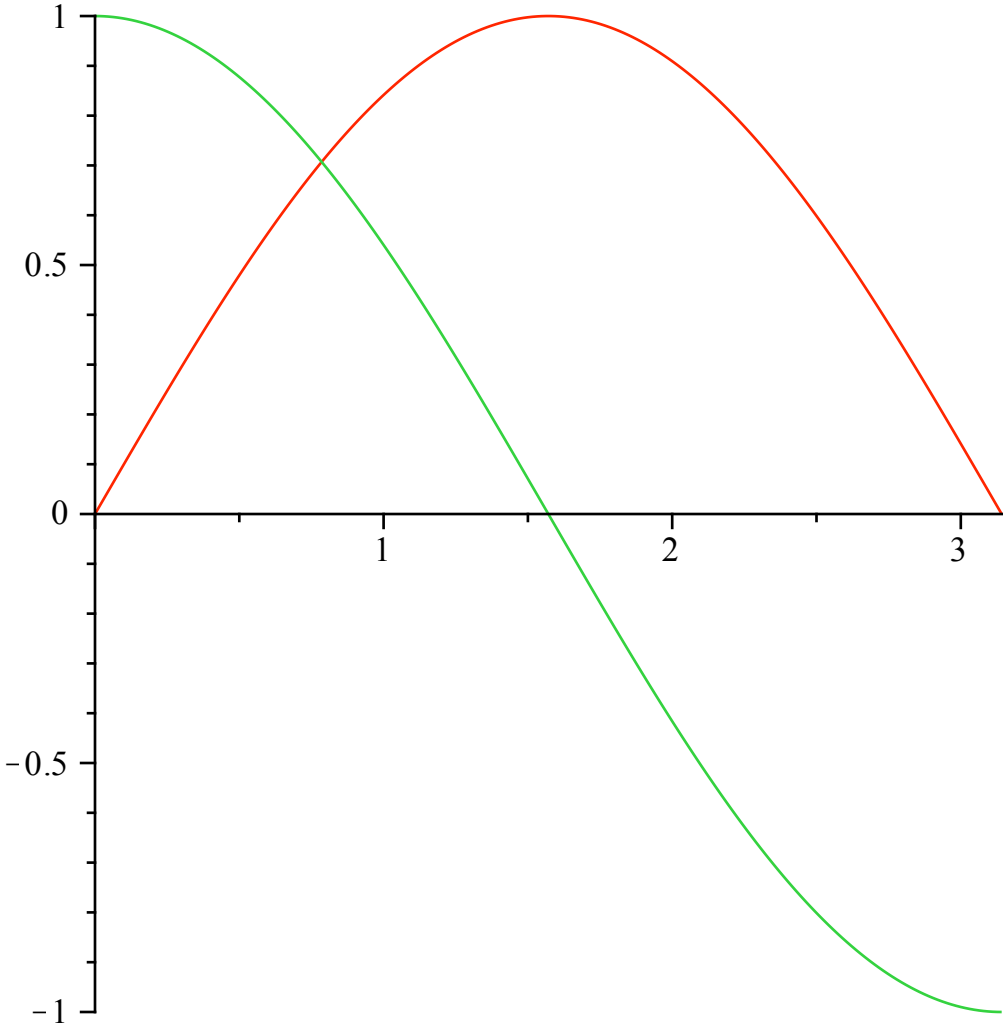
```
> f := sin(4·x) :
  plot(f, x = 0 ..2·Pi);
```



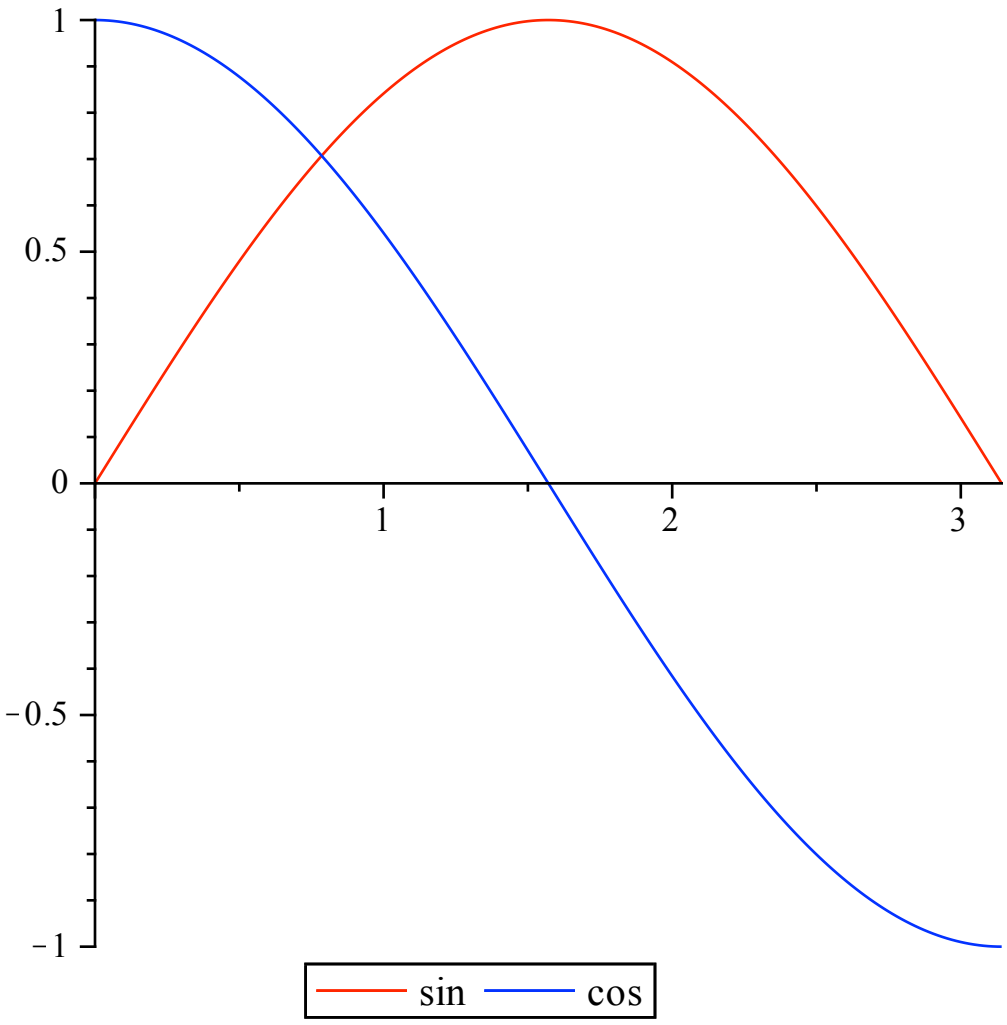

```
> f := x→sin(4·x) :
  plot( f, 0..2·Pi );
```



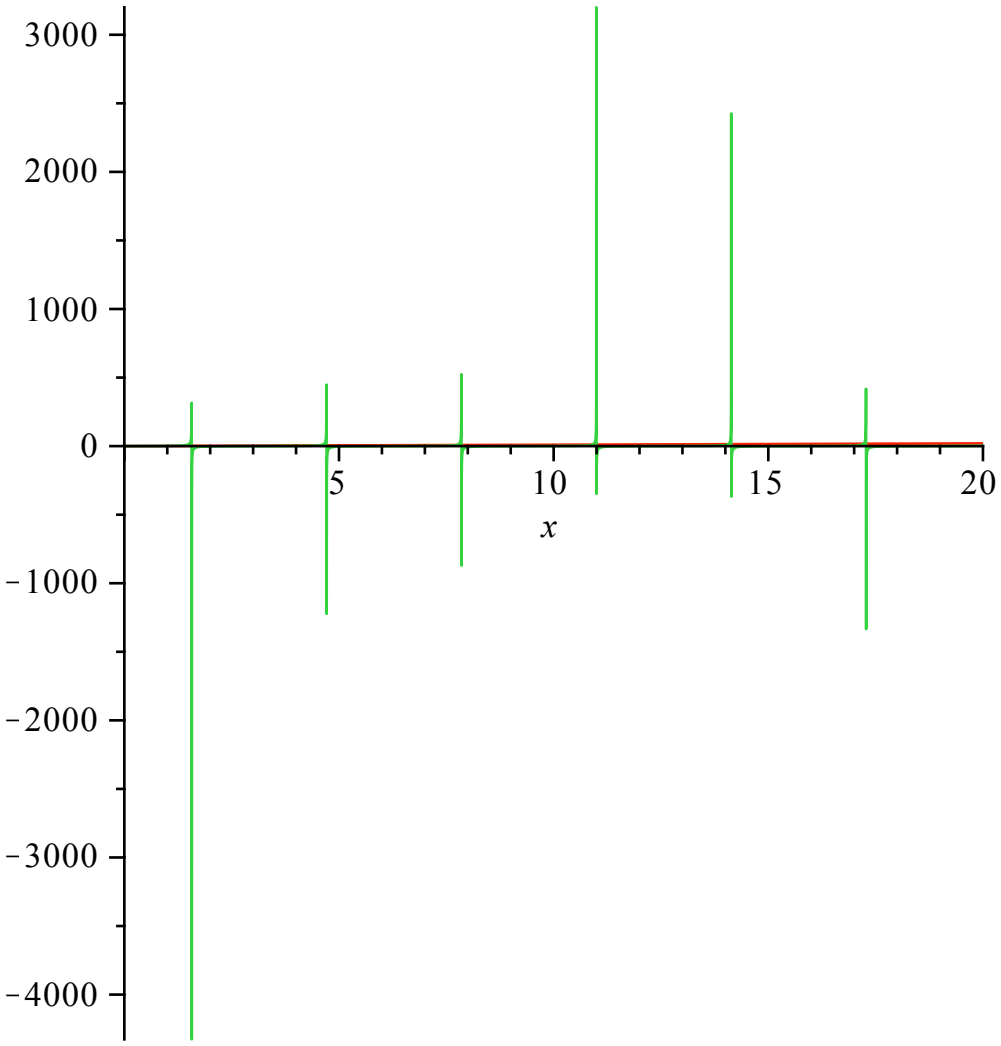

```
> plot([sin, cos], 0..Pi);
```



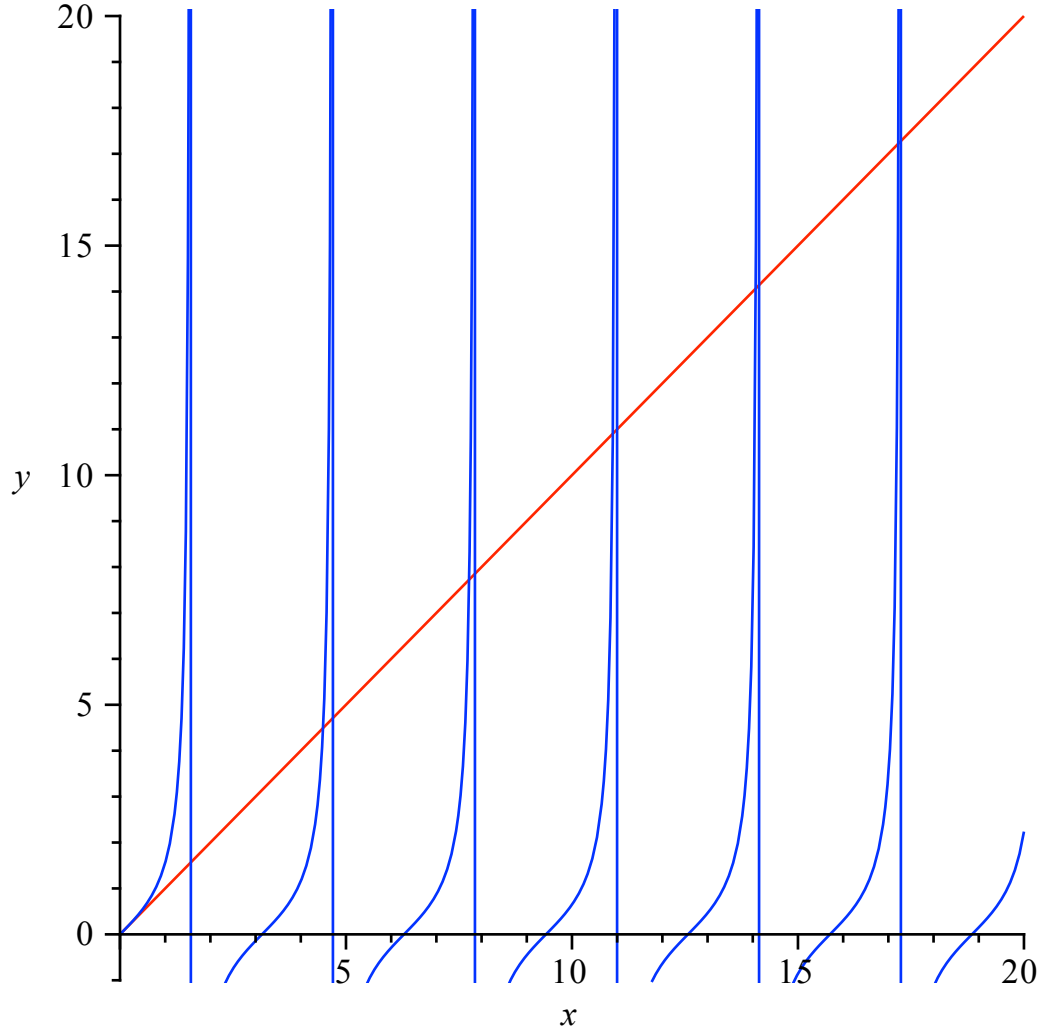

```
> plot([sin, cos], 0..Pi, color = [red, blue], legend = ["sin", "cos"]);
```



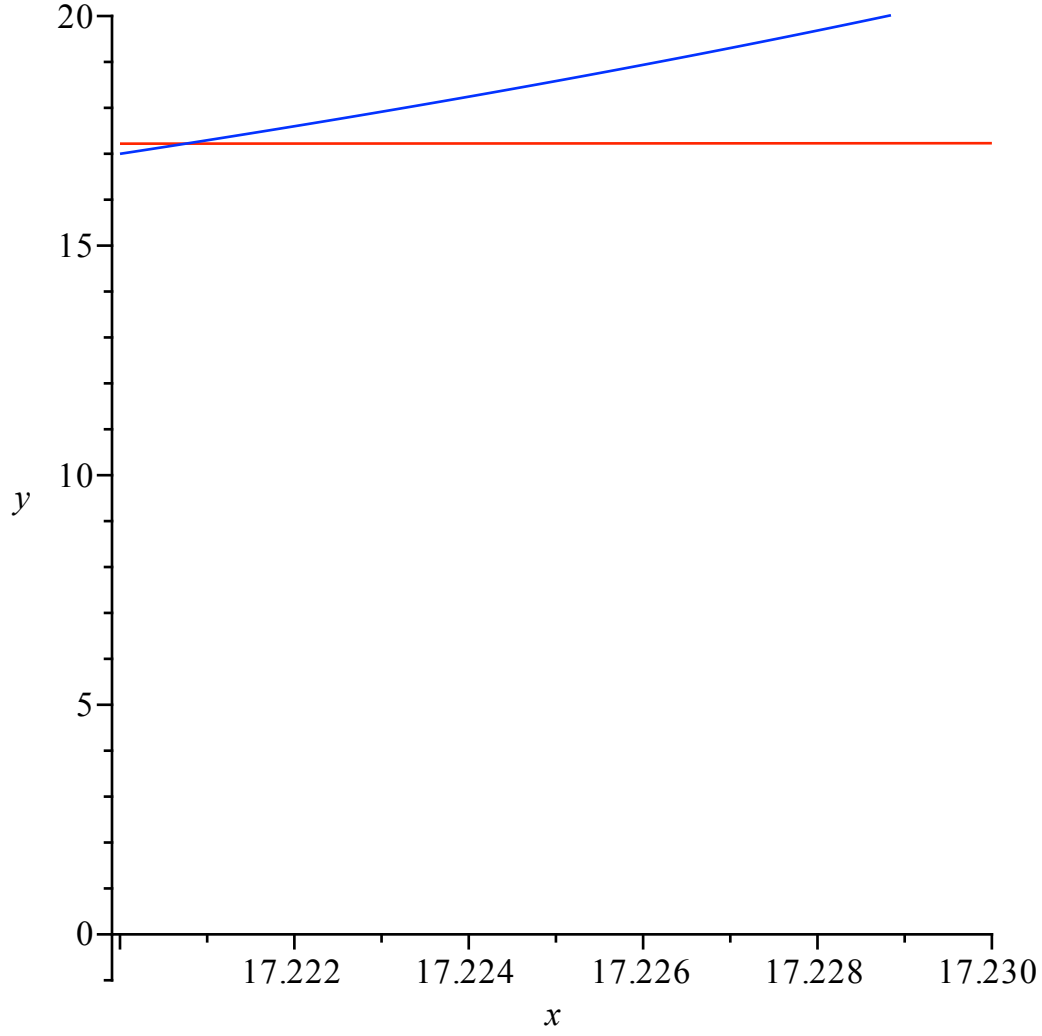

```
> ?plot3d
> #Exercice
> plot([x, tan(x)], x = 0..20);
```




```
> plot([x, tan(x)], x = 0..20, y = -1..20, color = [red, blue]);
```




```
> plot([x, tan(x)], x = 17.22..17.23, y = -1..20, color = [red, blue]);
```




```
> fsolve(x = tan(x), x = 17.22..17.23);
```

17.22075527 **(7.1.1)**

```
> #Fin exercice
> with(plots);
```

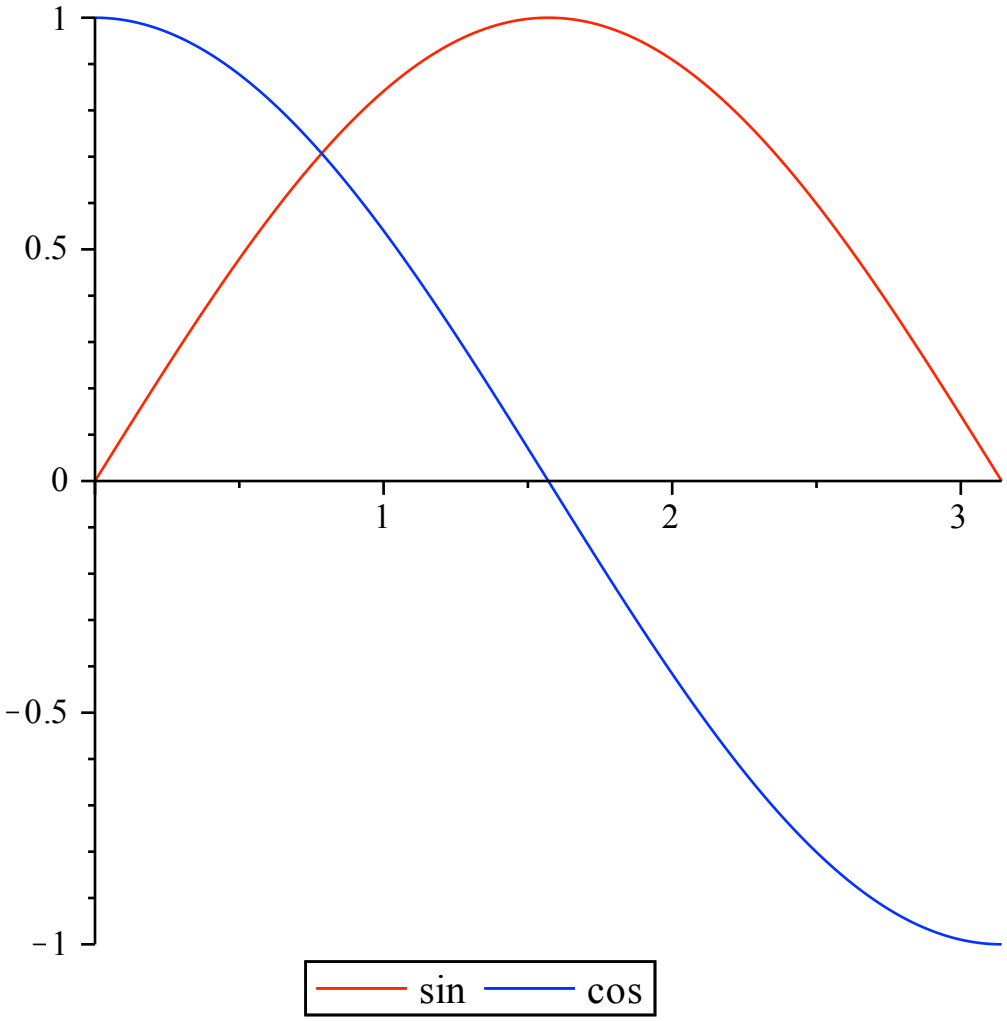
[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *densityplot*, *display*, *dualaxisplot*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*, *implicitplot3d*, *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*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*] **(7.1.2)**

```
> G1 := plot(sin, 0..Pi, color = red, legend = "sin");
  G2 := plot(cos, 0..Pi, color = blue, legend = "cos");
```

G1 := *PLOT*(...) **(7.1.3)**

$G2 := PLOT(\ldots)$ **(7.1.3)**

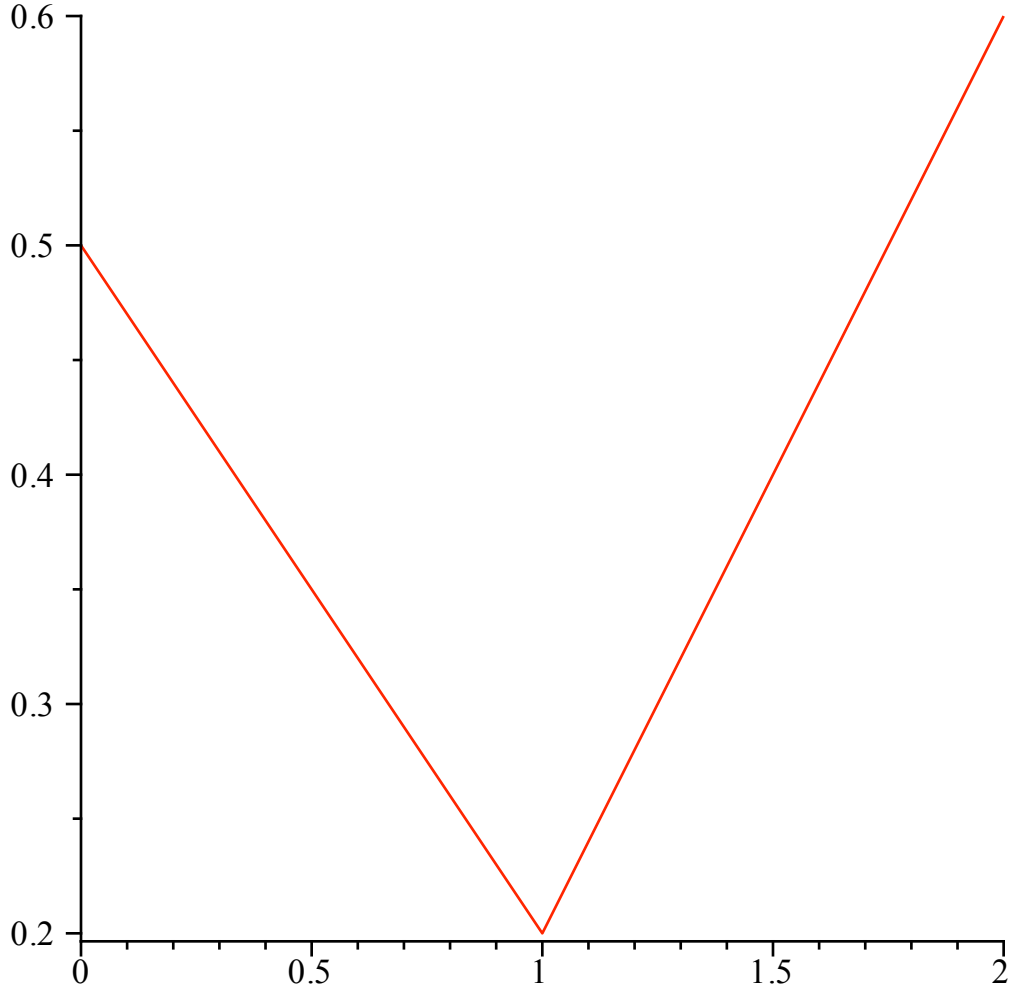
```
> display(G1, G2);
```



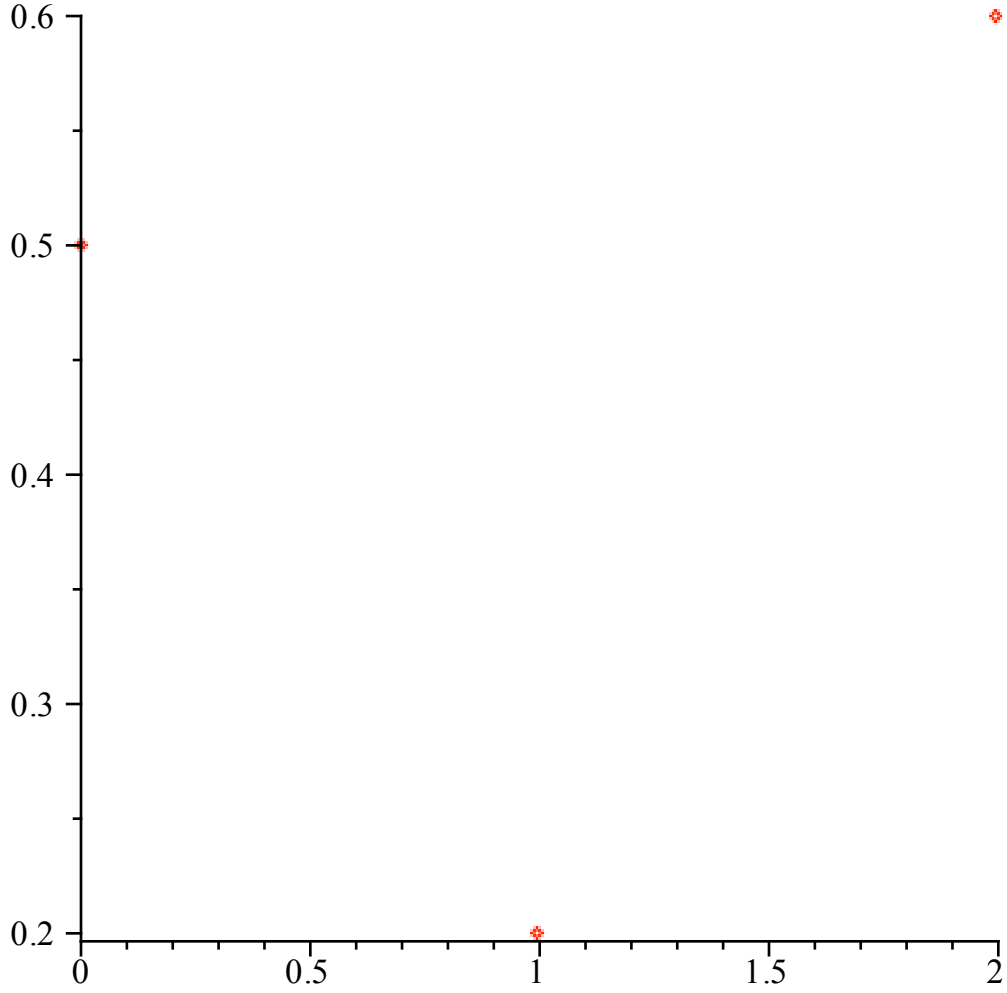

```
>
```

1.8.2 Un ensemble de points

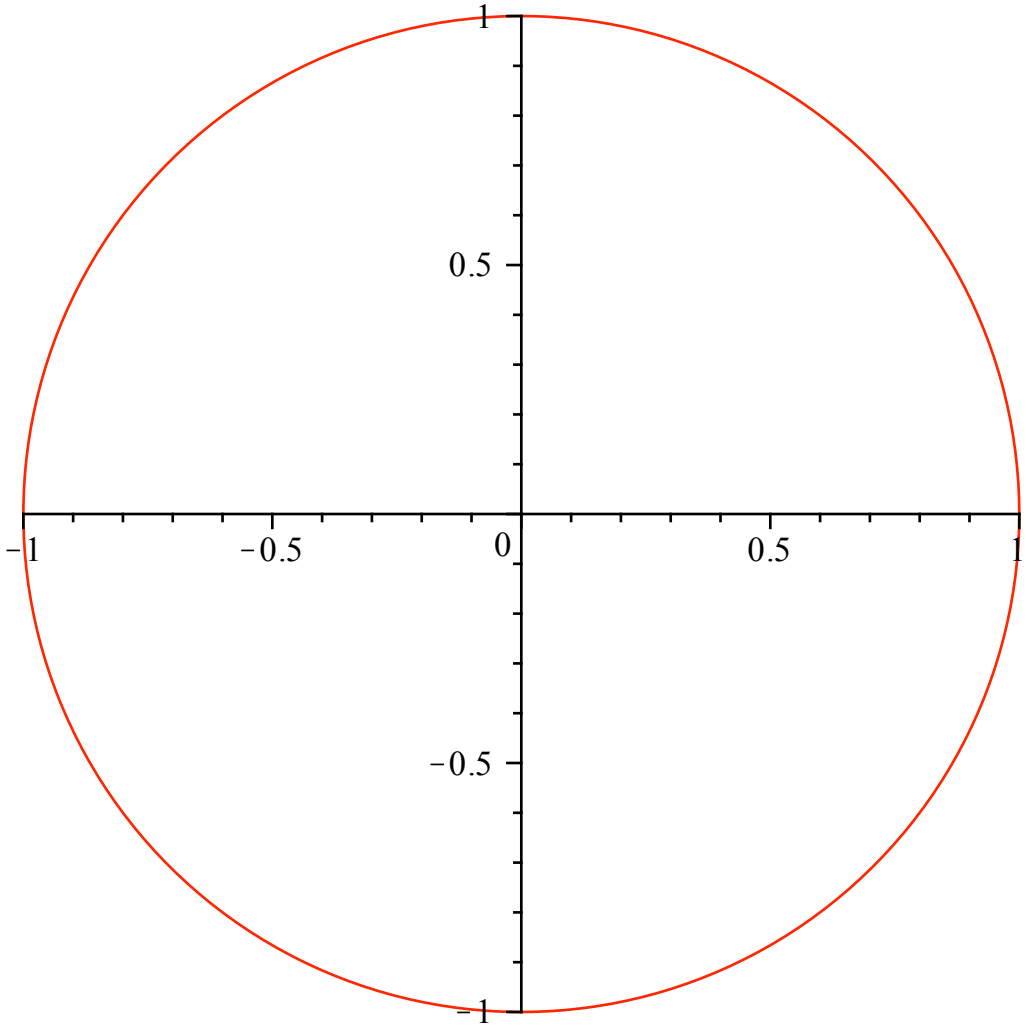
```
> plot([[0, 0.5], [1, 0.2], [2, 0.6]]);
```



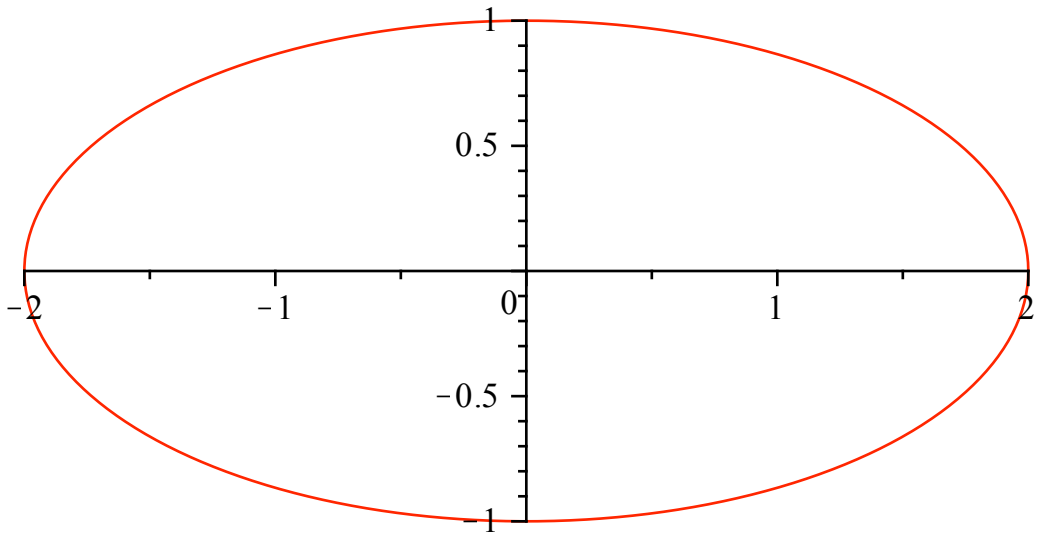

```
> plot([[0, 0.5], [1, 0.2], [2, 0.6]], style = point);
```




```
> plot([cos(t), sin(t), t = 0..2·Pi], scaling = constrained);
```




```
> plot([2·cos(t), sin(t), t = 0..2·π], scaling = constrained);
```



>

1.9 Résolution d'équations différentielles

> $f := x \to x^3$;
D(f);

$$f := x \to x^3$$

$$x \to 3x^2 \quad \textbf{(8.1)}$$

> D(f)(0);

$$0 \quad \textbf{(8.2)}$$

> *diff*(f(x), x);

$$3x^2 \quad \textbf{(8.3)}$$

> *subs*(x = 0, *diff*(f(x), x));

$$0 \quad \textbf{(8.4)}$$

```
> f := x^3;
  diff(f, x);
```

$$f := x^3$$

$$3x^2 \tag{8.5}$$

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

$$6x \tag{8.6}$$

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

$$\frac{1}{x} \tag{8.7}$$

```
> #ressort
> restart;
> m := 2; alpha := 0.5; k := 5;
```

$$m := 2$$

$$\alpha := 0.5$$

$$k := 5 \tag{8.8}$$

```
> ressort := {m·diff(x(t), t, t) + alpha·diff(x(t), t) + k·x(t) = 0};
```

$$ressort := \left\{2\left(\frac{d^2}{dt^2}x(t)\right) + 0.5\left(\frac{d}{dt}x(t)\right) + 5x(t) = 0\right\} \tag{8.9}$$

```
> CI := (a, b) → {x(0) = a, D(x)(0) = b};
```

$$CI := (a, b) \rightarrow \{x(0) = a, D(x)(0) = b\} \tag{8.10}$$

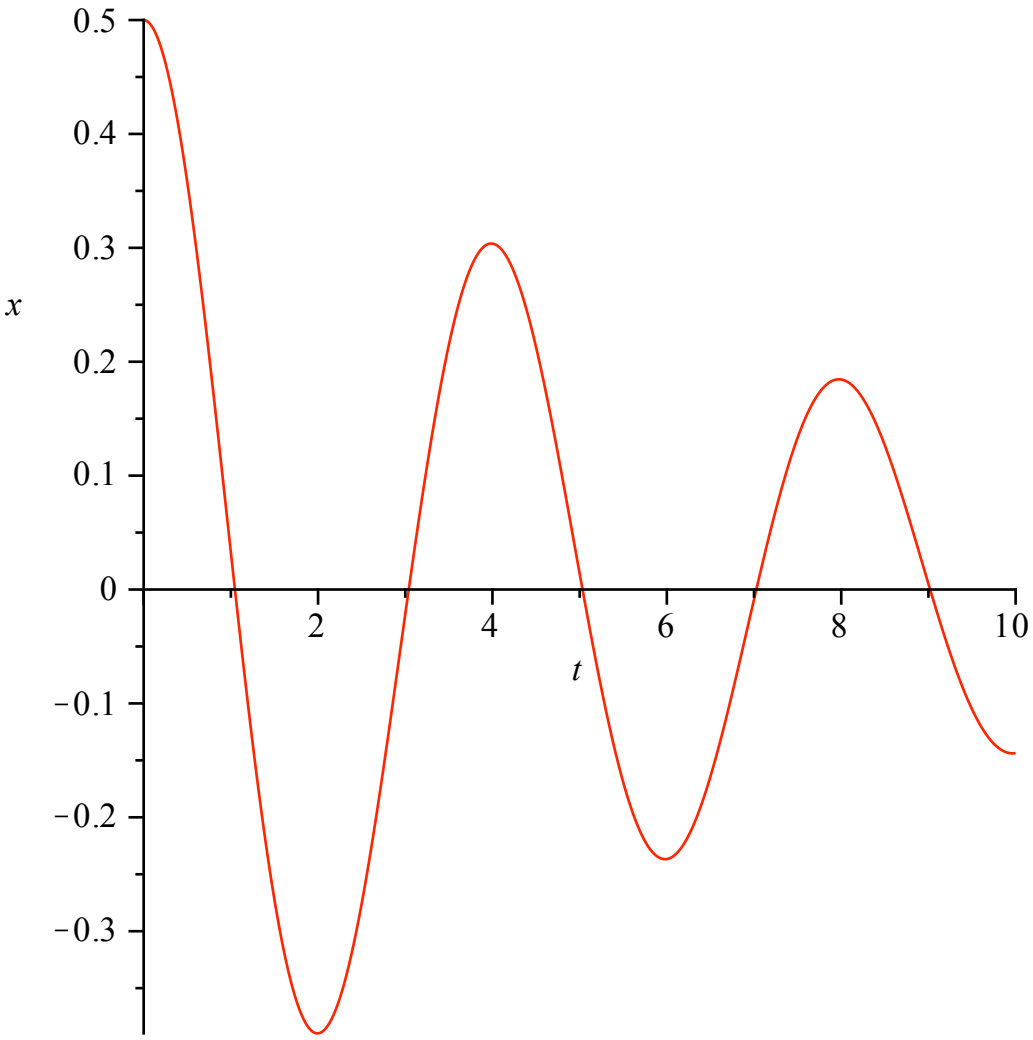
```
> sol := dsolve(ressort union CI(0.5, 0), x(t), numeric);
```

$$sol := \mathbf{proc}(x\_rkf45) \ \ldots \ \mathbf{end\ proc} \tag{8.11}$$

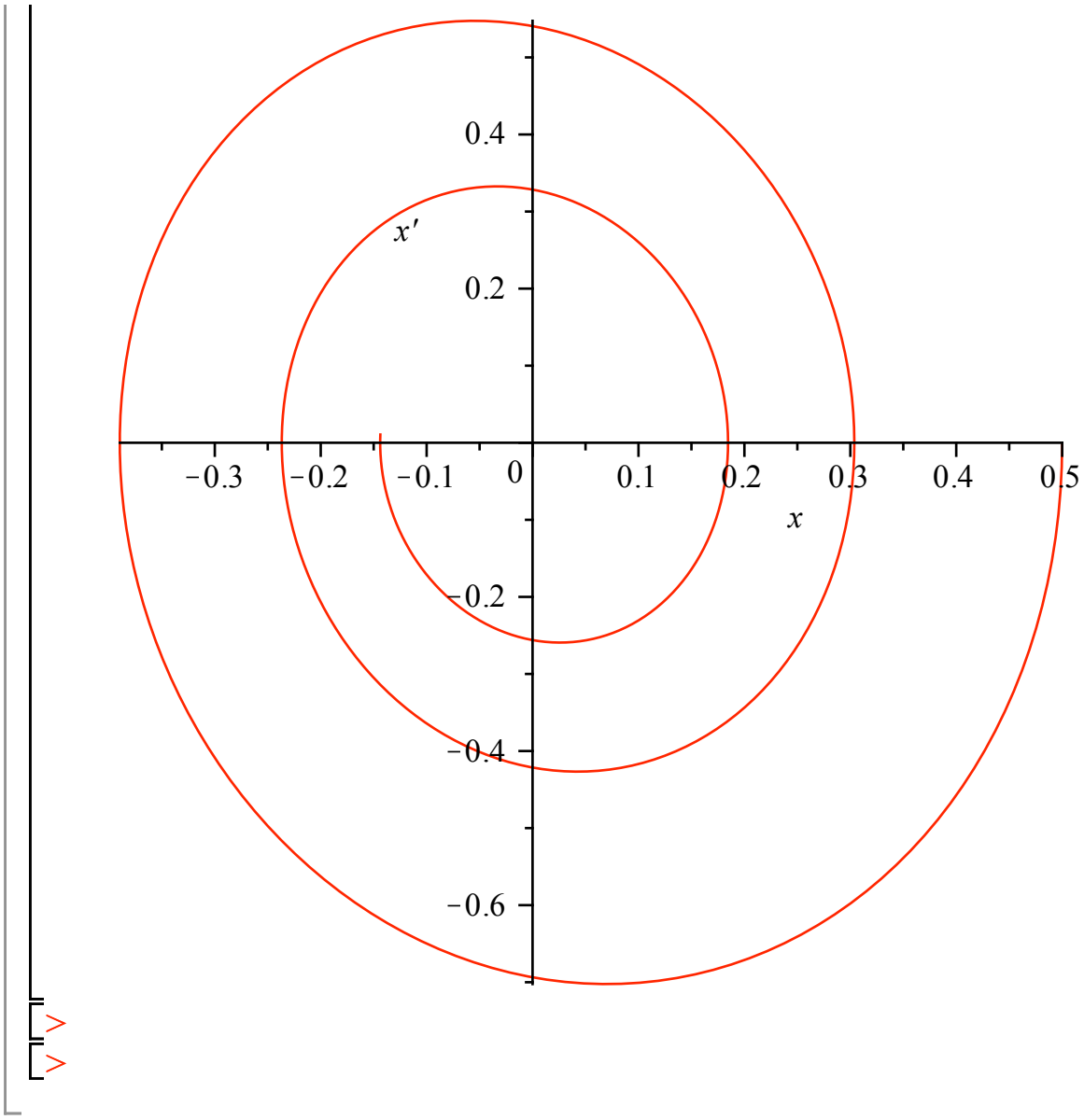
```
> with(plots);
```

[*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*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*] (8.12)

```
> odeplot(sol, [t, x(t)], 0..10, numpoints = 400);
```




```
> odeplot(sol, [x(t), diff(x(t), t)], 0..10, numpoints = 400);
```


0.4
0.2
x′
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
x
−0.2
−0.4
−0.6