

# RÉSOLUTION D'ÉQUATIONS ET D'INÉQUATIONS

Bernard Dupont

[Bernard.Dupont@univ-lille1.fr](mailto:Bernard.Dupont@univ-lille1.fr)

Résoudre des équations et des inéquations est le pain quotidien du modélisateur, théoricien ou technicien. Ce chapitre est consacré pour l'essentiel à l'utilisation des commandes universelles **solve** et **fsolve**, respectivement dédiées à la résolution exacte (section 1) et la résolution approchée (section 2) d'une équation. Pour les systèmes d'équations, on utilise les mêmes commandes, soit **solve** pour la résolution exacte (section 3) et **fsolve** pour la résolution approchée (section 4). Les inéquations sont aussi résolues avec **solve** (section 5). Enfin on donne des compléments indispensables pour mener à bien une résolution, en particulier en économie où les inconnues sont des réels alors que Maple "travaille" systématiquement dans le corps des complexes (section 6).

## Résolution exacte d'une équation

Soit  $F(x) = 0$  une équation où  $x$  est inconnue. Si  $F$  est un polynôme, La commande **solve** dispose d'une batterie de possibilités pour la résoudre dans le corps des nombres complexes. Dans le cas contraire, une résolution exacte est plutôt l'exception que la règle.

## Équation polynômiale

Les limites de Maple sont celles de la théorie mathématique. Si le degré du polynôme est inférieur ou égal à 4, il est théoriquement possible de donner une solution à l'aide de radicaux. Au delà de 5, c'est exceptionnel.

### Polynôme de degré inférieur ou égal à 3

La commande **solve(Xp=0,x)** où **Xp** est une expression polynômiale de degré inférieur ou égal à 3 de la variable **x** renvoie les racines réelles et complexes.

```
> restart;
```

```
xp:=a*x+b=0;solve(xp,x);#polynôme du premier degré
```

$$xp := a x + b = 0$$

$$-\frac{b}{a}$$

```
> xp2:=a*x^2+b*x+c=0;solve(xp2,x);#polynôme "symbolique" du second degré
```

$$xp2 := a x^2 + b x + c = 0$$

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

```
> xp3:=5*x^2+3*x+6=0;
```

```
solve(xp3,x);#polynôme "numérique" du second degré
```

$$xp3 := 5 x^2 + 3 x + 6 = 0$$

$$-\frac{3}{10} + \frac{1}{10} I\sqrt{111}, -\frac{3}{10} - \frac{1}{10} I\sqrt{111}$$

Maple renvoie une séquence composée ici de deux éléments. On récupère un élément de la séquence au moyen de l'opérateur d'indexage **[numero]** :

**> %[1];%%[2];**

$$-\frac{3}{10} + \frac{1}{10} I\sqrt{111}$$

$$-\frac{3}{10} - \frac{1}{10} I\sqrt{111}$$

Il est astucieux d'attribuer un nom à la séquence des racines puis d'isoler chacune d'elles en les assignant :

**> soluce:=solve(xp3,x);soluce1:=soluce[1];soluce2:=soluce[2];**

$$soluce := -\frac{3}{10} + \frac{1}{10} I\sqrt{111}, -\frac{3}{10} - \frac{1}{10} I\sqrt{111}$$

$$soluce1 := -\frac{3}{10} + \frac{1}{10} I\sqrt{111}$$

$$soluce2 := -\frac{3}{10} - \frac{1}{10} I\sqrt{111}$$

Si l'équation comporte des nombres flottants, Maple renvoie systématiquement des solutions sous forme de nombres flottants avec  $n$  chiffres significatifs,  $n$  étant paramétré par la variable

**Digits:=n.**

**> xp4:=5.\*x^2+3\*x+6=0;Digits:=3;sol:=solve(xp4,x);sol1:=sol[1];sol2:=sol[2];**

$$xp4 := 5. x^2 + 3 x + 6 = 0$$

$$Digits := 3$$

$$sol := -0.300 + 1.05 I, -0.300 - 1.05 I$$

$$sol1 := -0.300 + 1.05 I$$

$$sol2 := -0.300 - 1.05 I$$

Les principes sont les mêmes pour des polynômes du troisième degré.

**> xp5:=a\*x^3+b\*x^2+c\*x+d=0;solve(xp5,x);#polynôme "symbolique" du troisième degré**

$$xp5 := a x^3 + b x^2 + c x + d = 0$$

$$\frac{1}{6} \frac{1}{a} (36 c b a - 108 d a^2 - 8 b^3$$

$$+ 12 \sqrt{3} \sqrt{4 a c^3 - c^2 b^2 - 18 c b a d + 27 d^2 a^2 + 4 d b^3 a})^{1/3} - \frac{2}{3} (3 a c$$

$$- b^2) / (a (36 c b a - 108 d a^2 - 8 b^3$$

$$+ 12 \sqrt{3} \sqrt{4 a c^3 - c^2 b^2 - 18 c b a d + 27 d^2 a^2 + 4 d b^3 a})^{1/3}) - \frac{1}{3} \frac{b}{a},$$

$$- \frac{1}{12} \frac{1}{a} (36 c b a - 108 d a^2 - 8 b^3$$

$$\begin{aligned}
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} + \frac{1}{3} (3ac \\
& - b^2) / \left( a (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} \left. \right) - \frac{1}{3} \frac{b}{a} \\
& + \frac{1}{2} I\sqrt{3} \left( \frac{1}{6} \frac{1}{a} (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} + \frac{2}{3} (3ac \\
& - b^2) / \left( a (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} \left. \right) \left. \right), \\
& - \frac{1}{12} \frac{1}{a} (36cba - 108da^2 - 8b^3 \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} + \frac{1}{3} (3ac \\
& - b^2) / \left( a (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} \left. \right) - \frac{1}{3} \frac{b}{a} \\
& - \frac{1}{2} I\sqrt{3} \left( \frac{1}{6} \frac{1}{a} (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} + \frac{2}{3} (3ac \\
& - b^2) / \left( a (36cba - 108da^2 - 8b^3 \right. \\
& + 12\sqrt{3} \sqrt{4ac^3 - c^2b^2 - 18cbad + 27d^2a^2 + 4db^3} a)^{1/3} \left. \right) \left. \right)
\end{aligned}$$

**> xp6:=2\*x^3+3\*x^2-6\*x+7=0;#les coefficients du polynôme de degré 3 sont des entiers**

**sol:=solve(xp6,x);**

$$\begin{aligned}
& xp6 := 2x^3 + 3x^2 - 6x + 7 = 0 \\
sol := & -\frac{1}{2} (21 + 2\sqrt{79})^{1/3} - \frac{5}{2(21 + 2\sqrt{79})^{1/3}} - \frac{1}{2}, \frac{1}{4} (21 + 2\sqrt{79})^{1/3} \\
& + \frac{5}{4(21 + 2\sqrt{79})^{1/3}} - \frac{1}{2} + \frac{1}{2} I\sqrt{3} \left( -\frac{1}{2} (21 + 2\sqrt{79})^{1/3} \right. \\
& \left. + \frac{5}{2(21 + 2\sqrt{79})^{1/3}} \right), \frac{1}{4} (21 + 2\sqrt{79})^{1/3} + \frac{5}{4(21 + 2\sqrt{79})^{1/3}} - \frac{1}{2}
\end{aligned}$$

$$-\frac{1}{2} I\sqrt{3} \left( -\frac{1}{2} (21 + 2\sqrt{79})^{1/3} + \frac{5}{2(21 + 2\sqrt{79})^{1/3}} \right)$$

> `xp7:=2.*x^3+3*x^2-6*x+7=0;`#le premier coefficient de ce polynôme du 3° degré est un nombre flottant

`sol:=solve(xp7,x);`

`sol[1];sol[2];sol[3];`#assignation des trois racines

$$\begin{aligned} xp7 &:= 2x^3 + 3x^2 - 6x + 7 = 0 \\ sol &:= 0.716 + 0.826 I, -2.93, 0.716 - 0.826 I \\ & \quad 0.716 + 0.826 I \\ & \quad -2.93 \\ & \quad 0.716 - 0.826 I \end{aligned}$$

### Polynômes de degré 4

Des nouveautés apparaissent :

> `xp8:=a*x^4+b*x^3+c*x^2+d*x+e=0;``sol:=solve(xp8,x);`

$$xp8 := a x^4 + b x^3 + c x^2 + d x + e = 0$$

$$sol := \text{RootOf}(a \_Z^4 + b \_Z^3 + c \_Z^2 + d \_Z + e)$$

Par défaut, Maple renvoie en écho une fonction `RootOf`, signifiant que les racines du polynôme de degré 4 sont .. les racines de ce polynôme. C'est frustrant puisqu'en théorie il est possible d'expliciter toutes les solutions à l'aide de radicaux. Ceci étant, on peut forcer le système à renvoyer les solutions en fixant son environnement avec l'instruction

`_EnvExplicit:=true`

> `_EnvExplicit:=true;`

`xp9:=a*x^4+b*x^3+c*x^2+d*x+e=0;`

`sol:=solve(xp9,x);`#l'affichage de la sortie est volontairement bloqué.

`_EnvExplicit := true`

$$xp9 := a x^4 + b x^3 + c x^2 + d x + e = 0$$

De fait, l'output développe des dizaines de lignes peu digestes. Evidemment, si les coefficients sont des nombres du type flottant, le problème ne se pose pas :

> `xp10:=1.*x^4+5*x^3-5*x^2+2*x+6=0;``sol:=solve(xp10,x);`

$$xp10 := 1. x^4 + 5 x^3 - 5 x^2 + 2 x + 6 = 0$$

$$sol := 0.811 + 0.845 I, -0.744, -5.88, 0.811 - 0.845 I$$

### Polynômes de degré supérieur ou égal à 5

Soit Maple parvient à exprimer les racines à l'aide de radicaux ou de fonctions trigonométriques, ce qui est tout à fait exceptionnel, soit il répond avec des fonctions `RootOf`.

> `xp11:=x^8+x^4+1=0;``solve(xp11,x);`#polynôme de degré 8 avec des racines explicitées

`xp12:=x^7-1=0;``solve(xp12,x);`#polynôme de degré 7 avec des racines explicitées

`xp13:=x^5+3*x-1=0;``solve(xp13,x);`#polynôme de degré 5 avec des racines non explicitées

$$xp11 := x^8 + x^4 + 1 = 0$$

$$-\frac{1}{2} + \frac{1}{2} I\sqrt{3}, -\frac{1}{2} - \frac{1}{2} I\sqrt{3}, \frac{1}{2} + \frac{1}{2} I\sqrt{3}, \frac{1}{2} - \frac{1}{2} I\sqrt{3},$$

$$-\frac{1}{2} \sqrt{2+2I\sqrt{3}}, \frac{1}{2} \sqrt{2+2I\sqrt{3}}, -\frac{1}{2} \sqrt{2-2I\sqrt{3}}, \frac{1}{2} \sqrt{2-2I\sqrt{3}}$$

$$xp12 := x^7 - 1 = 0$$

$$1, \cos\left(\frac{2}{7}\pi\right) + I\sin\left(\frac{2}{7}\pi\right), -\cos\left(\frac{3}{7}\pi\right) + I\sin\left(\frac{3}{7}\pi\right), -\cos\left(\frac{1}{7}\pi\right)$$

$$+ I\sin\left(\frac{1}{7}\pi\right), -\cos\left(\frac{1}{7}\pi\right) - I\sin\left(\frac{1}{7}\pi\right), -\cos\left(\frac{3}{7}\pi\right) - I\sin\left(\frac{3}{7}\pi\right),$$

$$\cos\left(\frac{2}{7}\pi\right) - I\sin\left(\frac{2}{7}\pi\right)$$

$$xp13 := x^5 + 3x - 1 = 0$$

$$\text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=1), \text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=2), \text{RootOf}(\_Z^5$$

$$+ 3\_Z - 1, \text{index}=3), \text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=4), \text{RootOf}(\_Z^5 + 3\_Z$$

$$- 1, \text{index}=5)$$

Maple signale qu'il existe 5 racines pour la dernière équation. La commande `_EnvExplicit:=true` n'améliore pas l'information obtenue.

```
> _EnvExplicit:=true;solve(xp13,x);
```

$$\_EnvExplicit := true$$

$$\text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=1), \text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=2), \text{RootOf}(\_Z^5$$

$$+ 3\_Z - 1, \text{index}=3), \text{RootOf}(\_Z^5 + 3\_Z - 1, \text{index}=4), \text{RootOf}(\_Z^5 + 3\_Z$$

$$- 1, \text{index}=5)$$

Mais, dès que les coefficients sont tous des nombres du type flottant, toutes les racines sont explicitées :

```
> xp14:=x^5+3.*x-1=0;solve(xp14,x);
```

$$xp14 := x^5 + 3. x - 1 = 0$$

$$0.332, 0.839 + 0.944 I, -1.00 + 0.937 I, -1.00 - 0.937 I, 0.839 - 0.944 I$$

## Équation non polynômiale

Maple peut résoudre explicitement quelques équations non polynômiales. Lorsqu'il renvoie un résultat, il convient de se méfier car un certain nombre de racines peuvent être absentes .. Il est bon d'explorer le problème à l'aide d'une représentation graphique.

```
> restart;
```

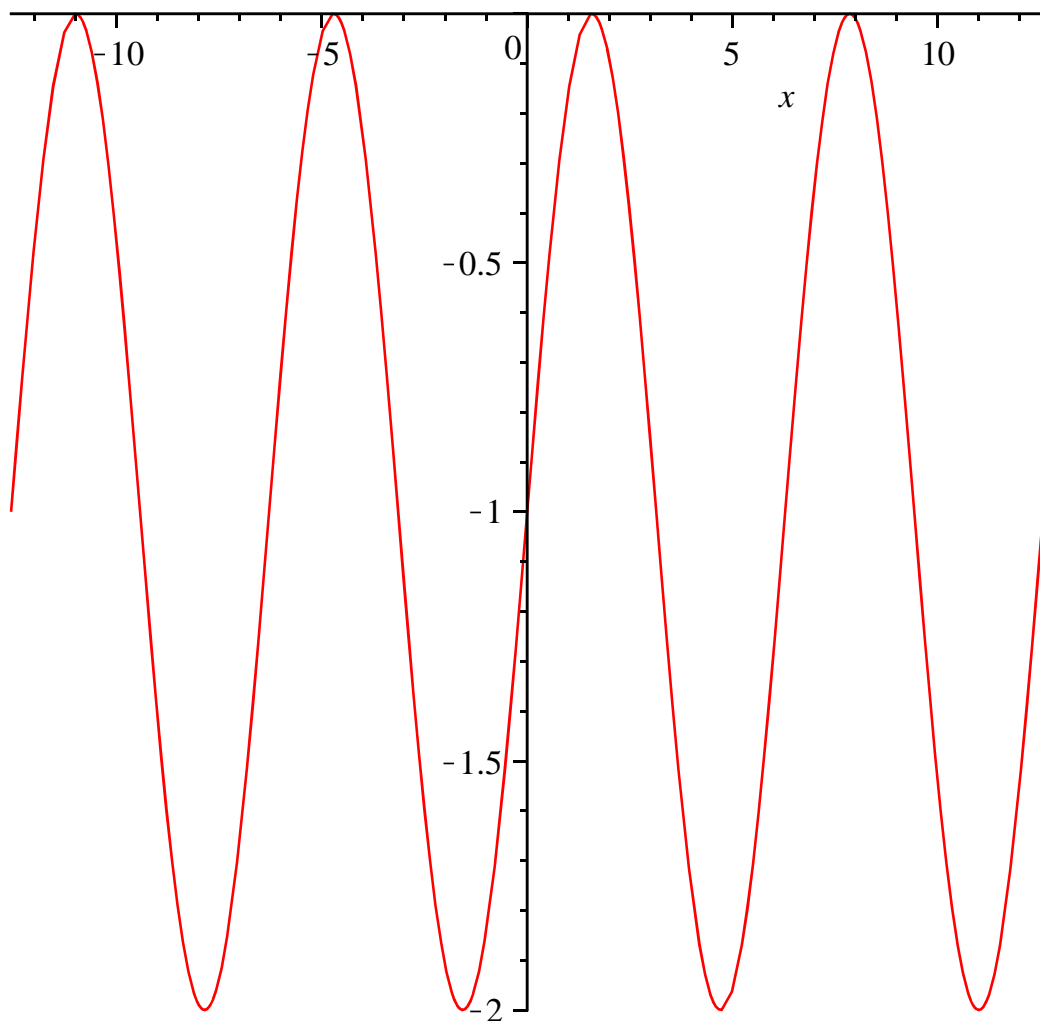
```
xp1:=sin(x)-1;
```

```
solve(xp1=0,x);
```

```
plot(xp1,x=-4*Pi..4*Pi);
```

$$xp1 := \sin(x) - 1$$

$$\frac{1}{2} \pi$$



En réalité, l'équation admet une infinité de solutions sur son domaine de définition. Maple a renvoyé une racine et il faut rectifier ce résultat en ajoutant l'expression bien connue "à  $2k\pi$  près".

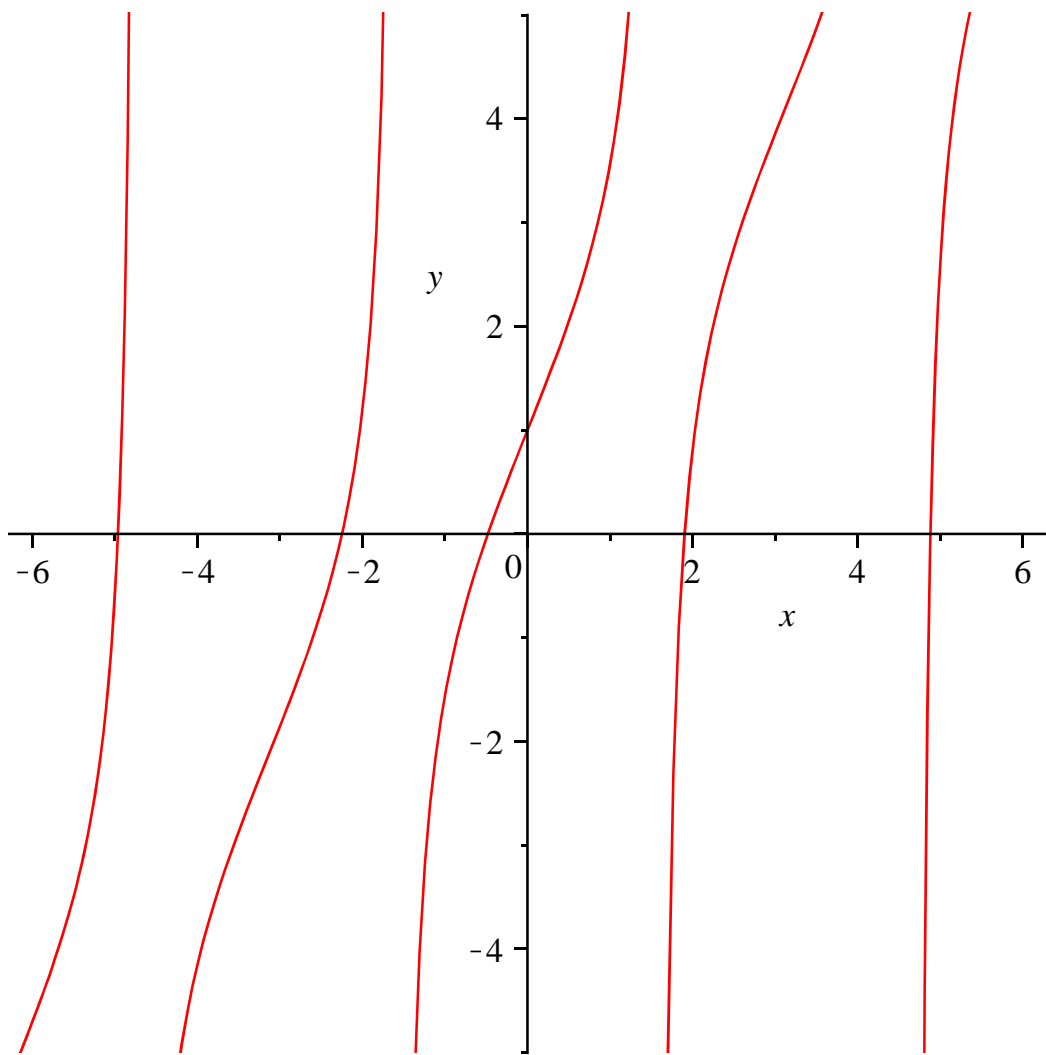
**Moralité : il faut être extrêmement vigilant et rigoureux quand on fait appel à l'instruction `solve`.**

Explorons d'autres singularités de cette instruction.

Il se peut que Maple retourne la question posée sous la forme laconique `RootOf` alors que des solutions existent :

```
> xp2:=tan(x)+x+1;solve(xp2=0,x);plot(xp2,x=-2*Pi..2*Pi,y=-5.5,discont=true);
```

```
      xp2 := tan(x) + x + 1
      RootOf(tan(_Z) + _Z + 1)
```

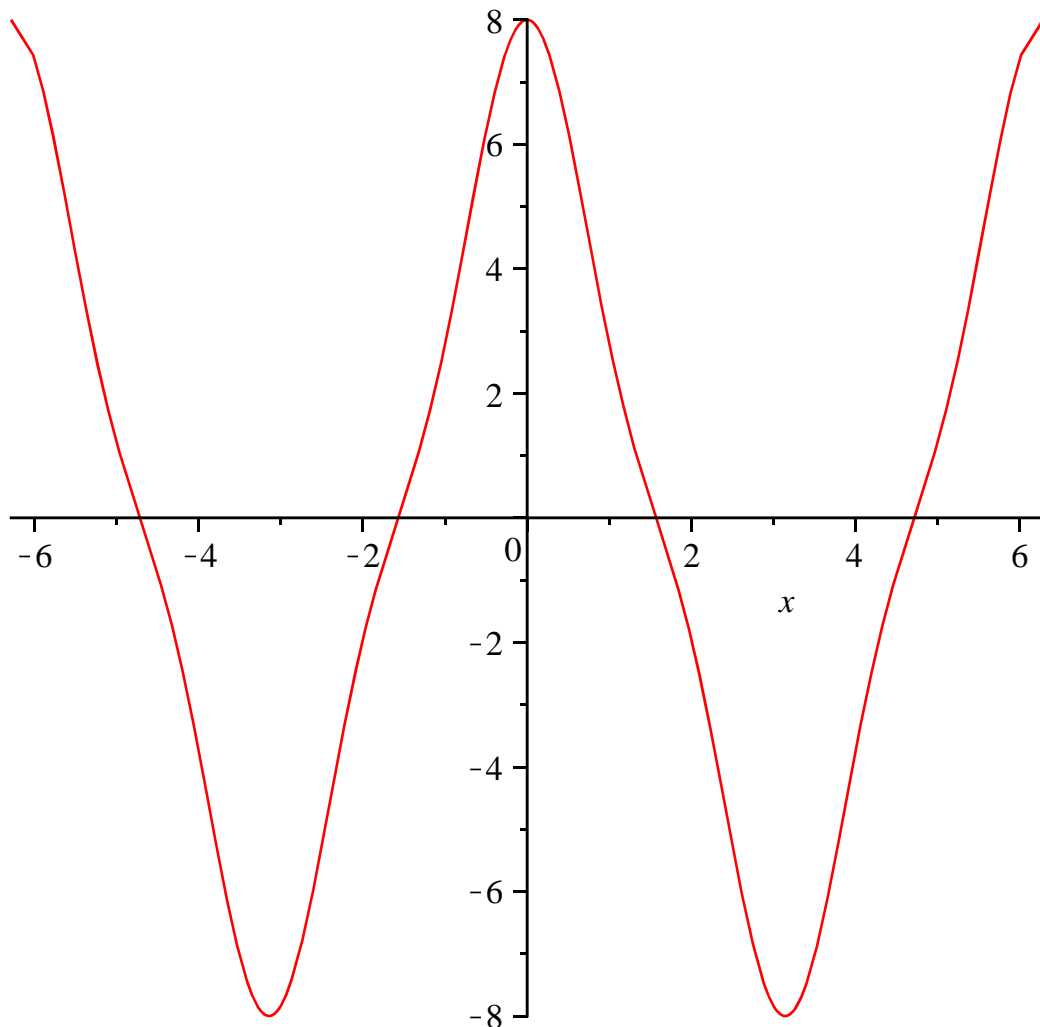


Des solutions complexes peuvent être renvoyées alors qu'on attend des solutions réelles : ce phénomène provient du fait que Maple cherche des solutions dans le corps des complexes.

```
> xp3:=cos(3*x)+7*cos(x);solve(xp3=0,x);plot(xp3,x=-2*Pi..2*Pi)
;
```

$$xp3 := \cos(3x) + 7 \cos(x)$$

$$\arccos(I), \arccos(-I), \frac{1}{2} \pi$$



En fait, la résolution d'une équation non polynômiale doit être généralement menée comme on le ferait "à la main". De plus, il faut avoir l'esprit critique vis à vis des réponses fournies par le logiciel.

## ▼ Résolution approchée d'une équation

La commande **fsolve** (abréviation de **floating solve**) permet de résoudre numériquement les équations. De nombreuses options permettent d'en améliorer les performances, notamment décevantes quand elle est utilisée de façon pure.

### ▼ Équation polynômiale

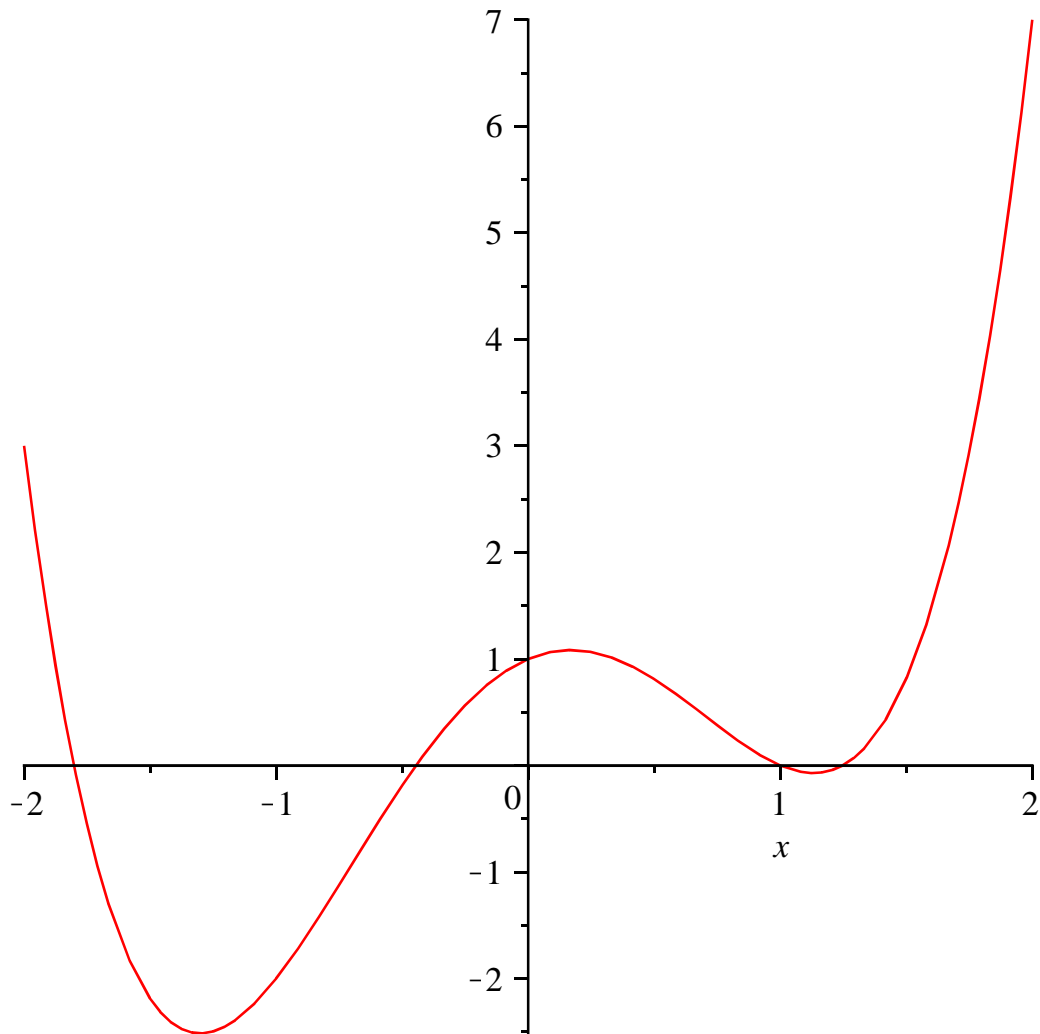
Soit  $F(x) = 0$  une équation polynômiale où  $x$  est inconnue et tous les coefficients sont numériques. En principe, **fsolve(F(x)=0,x)** retourne toutes les racines réelles de l'équation.

```
> restart;xp1:=x^4-3*x^2+x+1;
fsolve(xp1=0,x);#fsolve affiche 4 racines réelles
plot(xp1,x=-2..2);#vérification graphique du résultat
```

$$xp1 := x^4 - 3x^2 + x + 1$$

-1.801937736, -0.4450418679, 1., 1.246979604





Si nécessaire, on peut restreindre l'intervalle de résolution. Cette précision affine la recherche de solution. D'ailleurs, il arrive que Maple soit incapable de trouver une racine de l'équation quand **fsolve** est utilisée de façon pure et qu'il détecte une racine en lui indiquant un intervalle.

```
> fsolve(xp1=0,x=-1..0.5);
-0.4450418679
```

Le théorème de Cayley-Hamilton garantit qu'une équation de degré  $n$  a  $n$  racines complexes. On peut forcer Maple à calculer toutes les racines, réelles et complexes, en ajoutant l'option

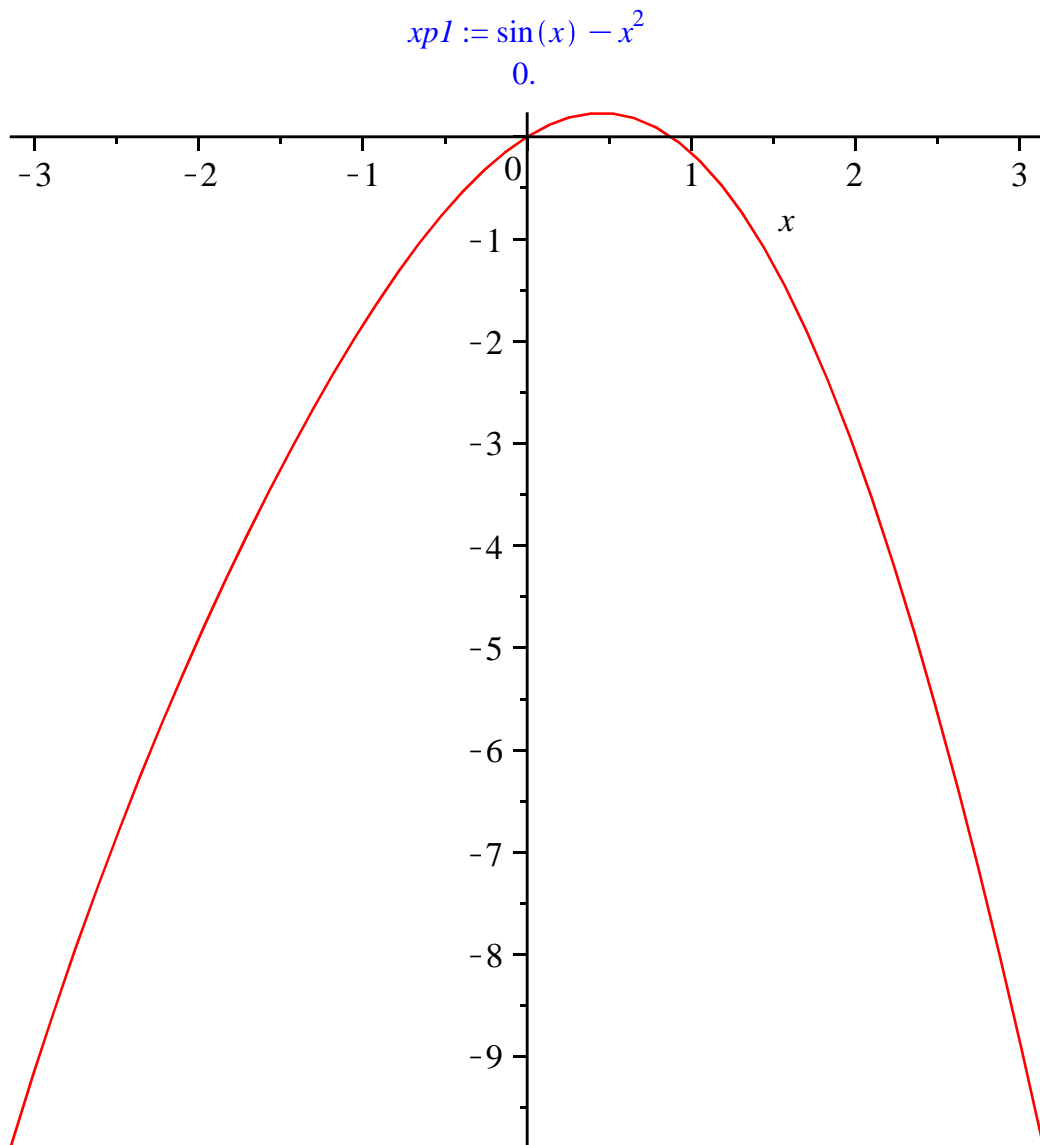
**complex** :

```
> xp2:=x^4-3*x^2+x+sqrt(2);
fsolve(xp2=0,x);#pour afficher les racines réelles
fsolve(xp2,x,complex);#pour afficher toutes les racines
xp2 := x4 - 3x2 + x + √2
-1.764214201, -0.5632117414
-1.764214201, -0.5632117414, 1.163712971 - 0.2627879113 I, 1.163712971
+ 0.2627879113 I
```

## ▼ Équation non polynômiale

Soit  $F(x) = 0$  une équation non polynômiale où  $x$  est inconnue et tous les coefficients sont numériques. Dans ce cas, `fsolve(F(x)=0,x)` retourne **au plus une** racine réelle de l'équation.

```
> restart;xp1:=sin(x)-x^2;#équation transcendentale
fsolve(xp1=0,x);#fsolve renvoie une racine
plot(xp1,x=-Pi..Pi);#la représentation graphique montre qu'il
y a deux racines.
```



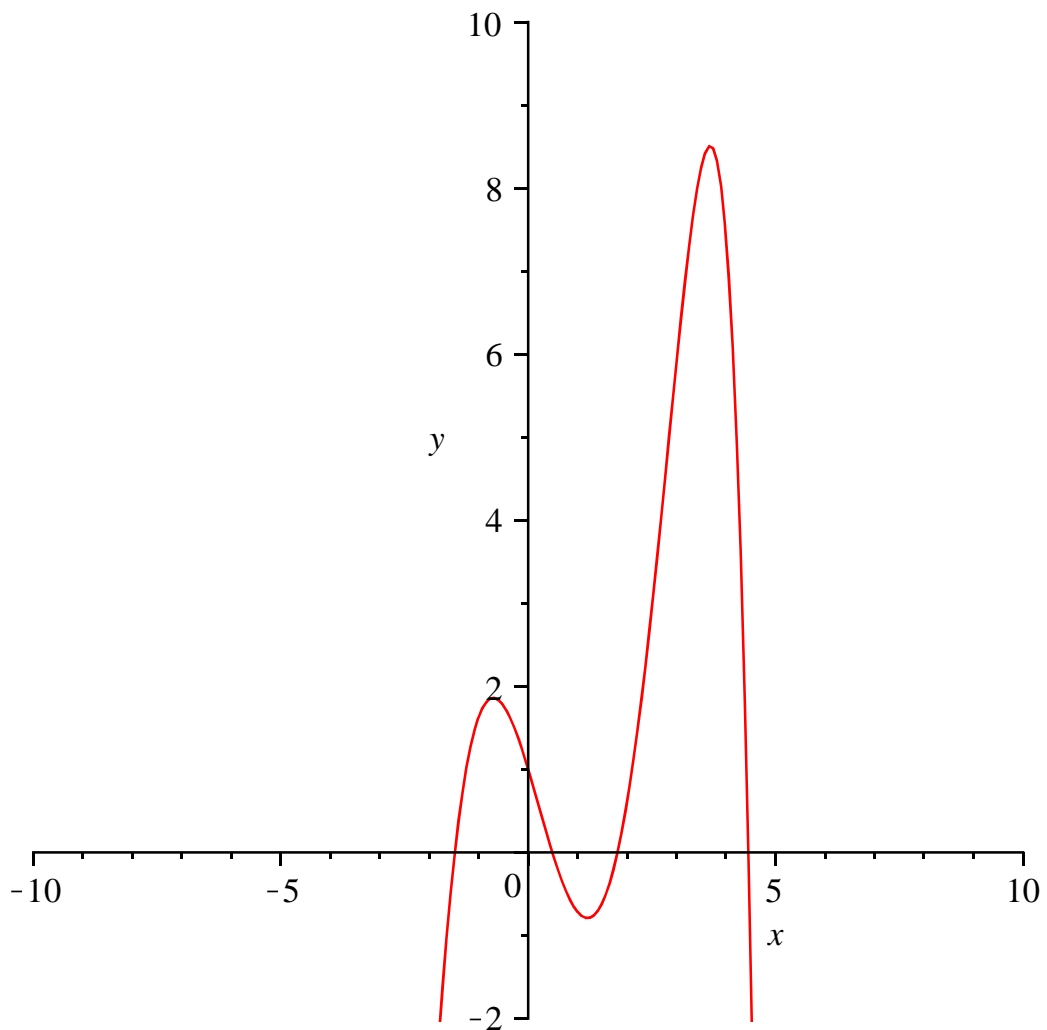
On aide Maple à détecter l'autre racine en intervenant sur l'intervalle de résolution :

```
> fsolve(xp1=0,x=0.8..1.2);
0.8767262154
```

L'utilisation de l'option `avoid` est parfois indispensable pour la recherche de toutes les solutions :

```
> xp2:=x^3-x+2-exp(x);sol1:=fsolve(xp2,x);#première utilisation
de fsolve
plot(xp2,x=-10..10,y=-2..10,numpoints=250);
```

```
xp2 := x3 - x + 2 - ex  
sol1 := 0.4875610348
```



```
> sol2:=fsolve(xp2,x,avoid={x=sol1});#deuxième utilisation de  
fsolve, en écartant la première solution  
sol2 := 4.451620297  
=>  
> sol3:=fsolve(xp2,x=0..10,avoid={x=sol1,x=sol2});#troisième  
utilisation de fsolve, en écartant la première et la deuxième  
solutions  
sol3 := 1.806570269  
=>  
> sol4:=fsolve(xp2,x=-infinity..0,avoid={x=sol1,x=sol2,x=sol3})  
;#quatrième utilisation de fsolve, en écartant les solutions  
précédentes  
sol4 := -1.481974815
```

## ▼ Résolution exacte d'un système d'équations

La résolution exacte d'un système d'équations est encore effectuée par la commande "à tout faire"

**solve(équations, inconnues)** où **equations** est un ensemble ou une liste d'équations et **inconnues** est un ensemble ou une liste d'inconnues. Il faut distinguer la résolution des systèmes linéaires des systèmes non linéaires.

## Systemes lineaires

En principe, Maple résout tous les systèmes linéaires dont les coefficients sont des nombres.

```
> restart;
sol1:=solve({x+y+z=5,2*x+sqrt(2)*y-4*z=1,x-5*y=4},{x,y,z});
#la solution est exprimée dans un ensemble
sol2:=solve([x+y+z=5,2*x+sqrt(2)*y-4*z=1,x-5*y=4],[x,y,z]);
#la solution est exprimée dans une liste
sol1 := {x = 2053/577 + 15/1154*sqrt(2), y = -51/577 + 3/1154*sqrt(2), z = 883/577 - 9/577*sqrt(2)}
sol2 := [[x = 2053/577 + 15/1154*sqrt(2), y = -51/577 + 3/1154*sqrt(2), z = 883/577 - 9/577*sqrt(2)]]
```

Les équations sont écrites entre accolades ou crochets ainsi que les inconnues. Le résultat est lui aussi entre accolades ou crochets. Pour Maple, tout élément ou séquence d'éléments entre accolades représente un ensemble d'éléments non ordonnés. Il n'est pas possible de prévoir l'ordre dans lequel vont apparaître les racines.

Au vu du résultat, on peut récupérer isolément les racines à l'aide de **subs** :

```
> solx:=subs(sol1,x);soly:=subs(sol1,y);solz:=subs(sol1,z);
solx := 2053/577 + 15/1154*sqrt(2)
soly := -51/577 + 3/1154*sqrt(2)
solz := 883/577 - 9/577*sqrt(2)
```

ou récupérer les solutions sous forme d'une liste ordonnée (soit sous forme d'une séquence d'éléments mis entre crochets) :

```
> ['solx','soly','solz']=subs(sol1,[x,y,z]);
[solx, soly, solz] = [2053/577 + 15/1154*sqrt(2), -51/577 + 3/1154*sqrt(2), 883/577 - 9/577*sqrt(2)]
```

Quand le système n'a pas de solution, Maple renvoie une séquence vide :

```
> sol3:=solve({2*x+y=1,2*x+y=5},{x,y});
sol3 :=
```

Quand le système admet une infinité de solutions (cas où les équations sont linéairement dépendantes), Maple renvoie une solution sous forme paramétrique dans laquelle une ou plusieurs inconnues servent de paramètres :

```
> sol4:=solve({x+y+z=1,2*x+2*y+2*z=2,x-y+2*z=4},{x,y,z});
sol4 := {x = -2 - 3*y, y = y, z = 3 + 2*y}
```

En ce qui concerne les systèmes dans lesquels certains ou tous les coefficients sont des paramètres, Maple cherche une solution "générale" qui ne tient pas compte des cas particuliers. Il convient alors d'être très vigilant.

```
> sol5:=solve({a*x+y=1,x+a*y=1},{x,y});
```

$$sol5 := \left\{ x = \frac{1}{a+1}, y = \frac{1}{a+1} \right\}$$

La solution générale fournie ici n'est pas correcte pour  $a = 1$  (car, dans ce cas, les équations sont linéairement dépendantes) et pour  $a = -1$  (car alors le dénominateur est nul, ce qui est impossible). Il faut étudier à part ces deux cas :

```
> sys:={a*x+y=1,x+a*y=1};
soluce_1:=solve(subs(a=1,sys),{x,y});
soluce_2:=solve(subs(a=-1,sys),{x,y});
sys := {x + a y = 1, a x + y = 1}
soluce_1 := {x = -y + 1, y = y}
soluce_2 :=
```

## ▼ Systèmes non linéaires

Nous sommes dans le règne du (quasi) arbitraire. Certains systèmes sont solubles et Maple renvoie la solution correcte :

```
> restart;
solve({x+2*y=4,x*y=2},{x,y});
{x=2, y=1}, {x=2, y=1}
```

D'autres sont solubles, mais Maple renvoie une solution décevante :

```
> solve({3*x+2*y=4,x*y=2},{x,y});
{x=2 RootOf(3 _Z^2 - 2 _Z + 1, label=_L2), y=-3 RootOf(3 _Z^2 - 2 _Z + 1, label
=_L2) + 2}
```

```
> evalf(%)#on force l'évaluation, mais Maple ne reconnaît pas
2/3 et sqrt(2)
{x=0.6666666666 + 0.9428090416 I, y=1.0000000000 - 1.414213562 I}
```

Signalons enfin que l'instruction **solve** a été programmée pour être performante sur les systèmes symétriques :

```
> restart;solve({x+y+z=6,x^2+y^2+z^2=14,x*y*z=6},{x,y,z});
{x=3, y=1, z=2}, {y=1, x=2, z=3}, {x=3, y=2, z=1}, {x=1, y=2, z=3}, {x=2,
z=1, y=3}, {x=1, z=2, y=3}
```

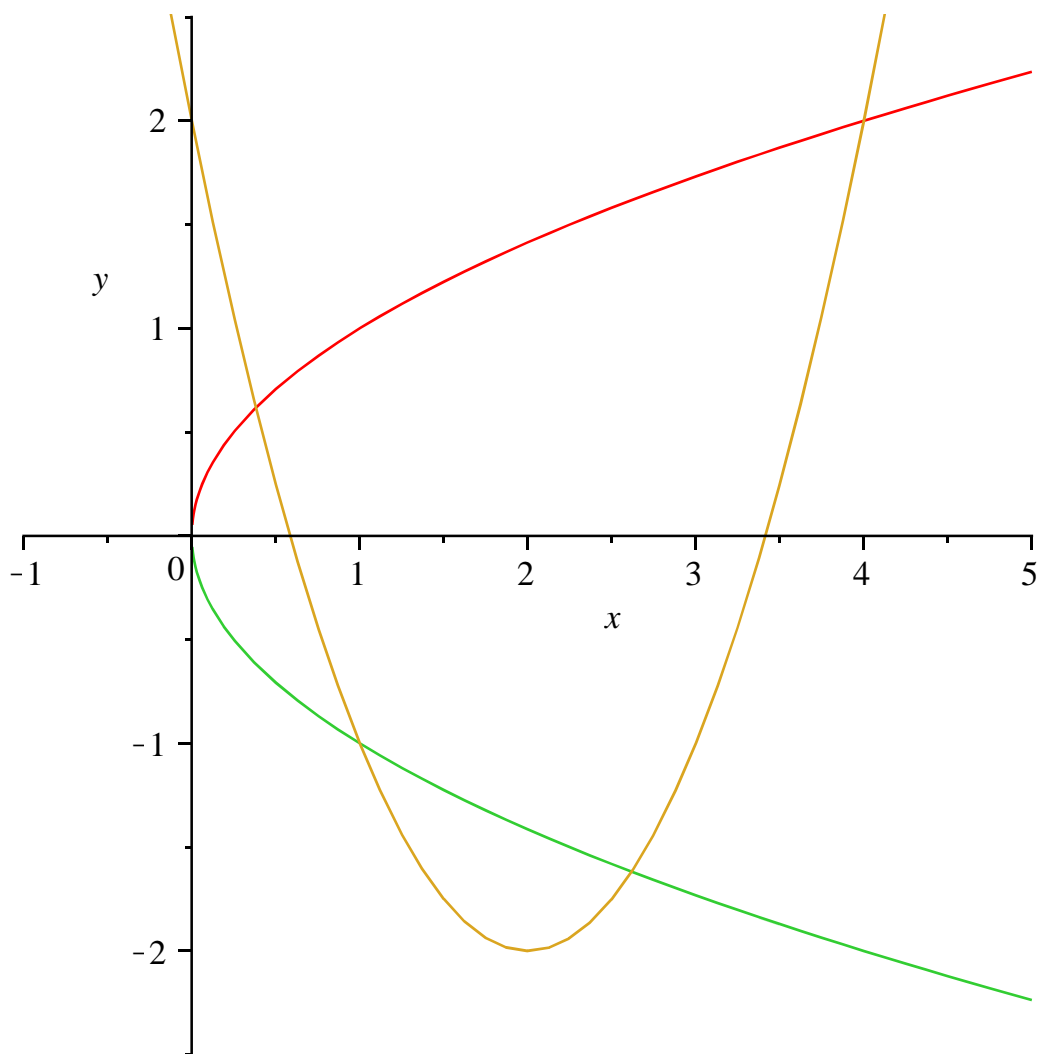
## ▼ Résolution approchée d'un système d'équations

Comme pour la résolution d'équations, on utilise la commande **fsolve** pour résoudre numériquement un système d'équations. Le premier argument est un ensemble de  $n$  équations. Le second argument est un ensemble de  $n$  inconnues. Certaines options permettent d'affiner la recherche.

```
> restart;
fsolve({y^2-x=0,y-x^2+4*x-2=0},{x,y});
{x=2.618033989, y=-1.618033989}
```

Une représentation graphique du problème permet d'aider le travail de Maple :

```
> plot({sqrt(x),-sqrt(x),x^2-4*x+2},x=-1..5,y=-2.5..2.5);
```



On précise dans un troisième argument le voisinage dans lequel Maple doit chercher une solution :

```
> fsolve({y^2-x=0,y-x^2+4*x-2=0},{x,y},{x=0.2..0.8,y=0.2..0.8});
      {x=0.3819660113,y=0.6180339887}
> fsolve({y^2-x=0,y-x^2+4*x-2=0},{x,y},{x=0.8..1.2,y=-1.2..-0.8})
;
      {x=1.000000000,y=-1.000000000}
> fsolve({y^2-x=0,y-x^2+4*x-2=0},{x,y},{x=3.8..4.2,y=1.8..2.2});
      {x=4.000000000,y=2.000000000}
```

Cet exemple n'a bien entendu aucune valeur universelle. Chaque cas un peu tordu est un cas d'espèce à étudier avec beaucoup de patience et de rigueur.

## ▼ Résolution d'inéquations

Maple peut résoudre certaines inéquations et même certains systèmes d'inéquations à une variable à l'aide de **solve**. Les symboles d'inégalité reconnus par le logiciel sont : strictement inférieur à : **<**; strictement supérieur à : **>**; inférieur ou égal à : **<=**; supérieur ou égal à : **>=**. Au passage, signalons que l'opérateur "différent de" s'écrit : **<>**.

## Inéquation à une variable

L'exemple suivant familiarise avec la syntaxe et montre comment Maple exprime que le résultat est la réunion d'intervalles ouverts de  $\mathbb{R}$  :

```
> restart;
```

```
P:=x^3+4*x^2+2*x-1;solve(P>0,x);solve(P<0,x);
```

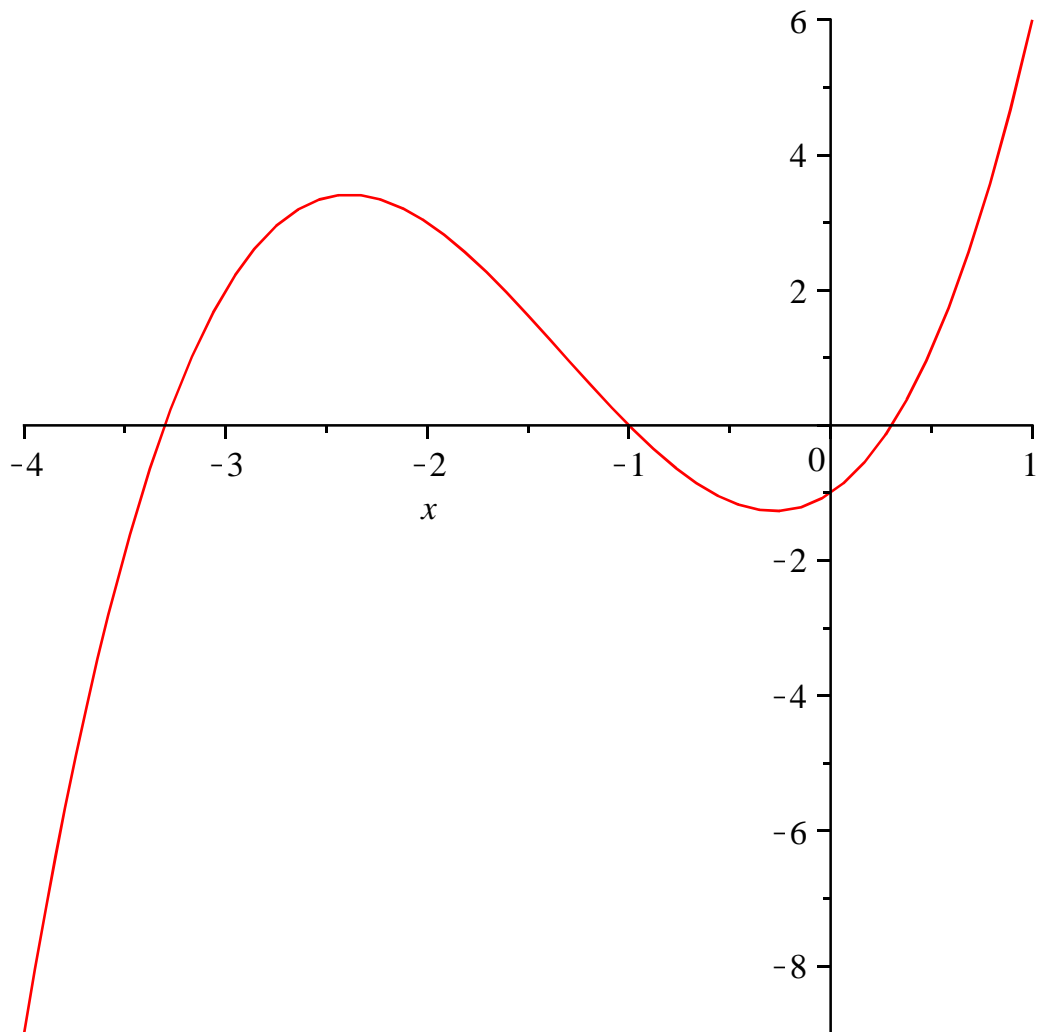
$$P := x^3 + 4x^2 + 2x - 1$$

```
RealRange(Open(-3/2 - 1/2*sqrt(13)), Open(-1)), RealRange(Open(-3/2 + 1/2*sqrt(13)), infinity)
```

```
RealRange(-infinity, Open(-3/2 - 1/2*sqrt(13))), RealRange(Open(-1), Open(-3/2 + 1/2*sqrt(13)))
```

Une représentation graphique permet de s'assurer de la justesse du résultat :

```
> plot(P,x=-4..1);
```



Si on désire voir figurer explicitement la variable  $x$  dans la réponse, il faut mettre en accolades

L'inéquation et la variable :

```
solve({P>0},{x});solve({P<0},{x});  
 $\left\{x < -1, -\frac{3}{2} - \frac{1}{2}\sqrt{13} < x\right\}, \left\{-\frac{3}{2} + \frac{1}{2}\sqrt{13} < x\right\}$   
 $\left\{x < -\frac{3}{2} - \frac{1}{2}\sqrt{13}\right\}, \left\{-1 < x, x < -\frac{3}{2} + \frac{1}{2}\sqrt{13}\right\}$ 
```

## ▼ Système d'inéquations linaires à deux inconnues

La commande **solve** permet de "résoudre" des systèmes d'inéquations à deux inconnues.

```
> restart;solve({x+y>0, x-y<=1},{x,y});  
 $\left\{x \leq \frac{1}{2}, -x < y\right\}, \left\{-1 + x \leq y, \frac{1}{2} < x\right\}$   
> solve({x+2*y=3, y+1/x>1}, {x,y});  
 $\left\{x = -2y + 3, \frac{1}{2} < y, y < \frac{3}{2}\right\}, \left\{x = -2y + 3, 2 < y\right\}$ 
```

Mais Maple offre surtout l'avantage spectaculaire d'obtenir une représentation graphique de l'ensemble des solutions possibles ("faisables") par l'instruction **inequal** du paquetage **plots**. Sa syntaxe est **inequal(ineqs, xspec, yspec, options)** où **ineqs** est un système d'inéquations écrit sous forme d'ensemble (donc, entre accolades); **xspec** précise l'intervalle de la variable  $x$  et **yspec** celui de  $y$ . Le graphe contient 4 éléments :

L'ensemble faisable (feasible region), qui réunit les zones pour lesquelles toutes les inégalités sont satisfaites;

L'ensemble exclu (excluded regions), qui réunit les zones où au moins une inégalité n'est pas respectée;

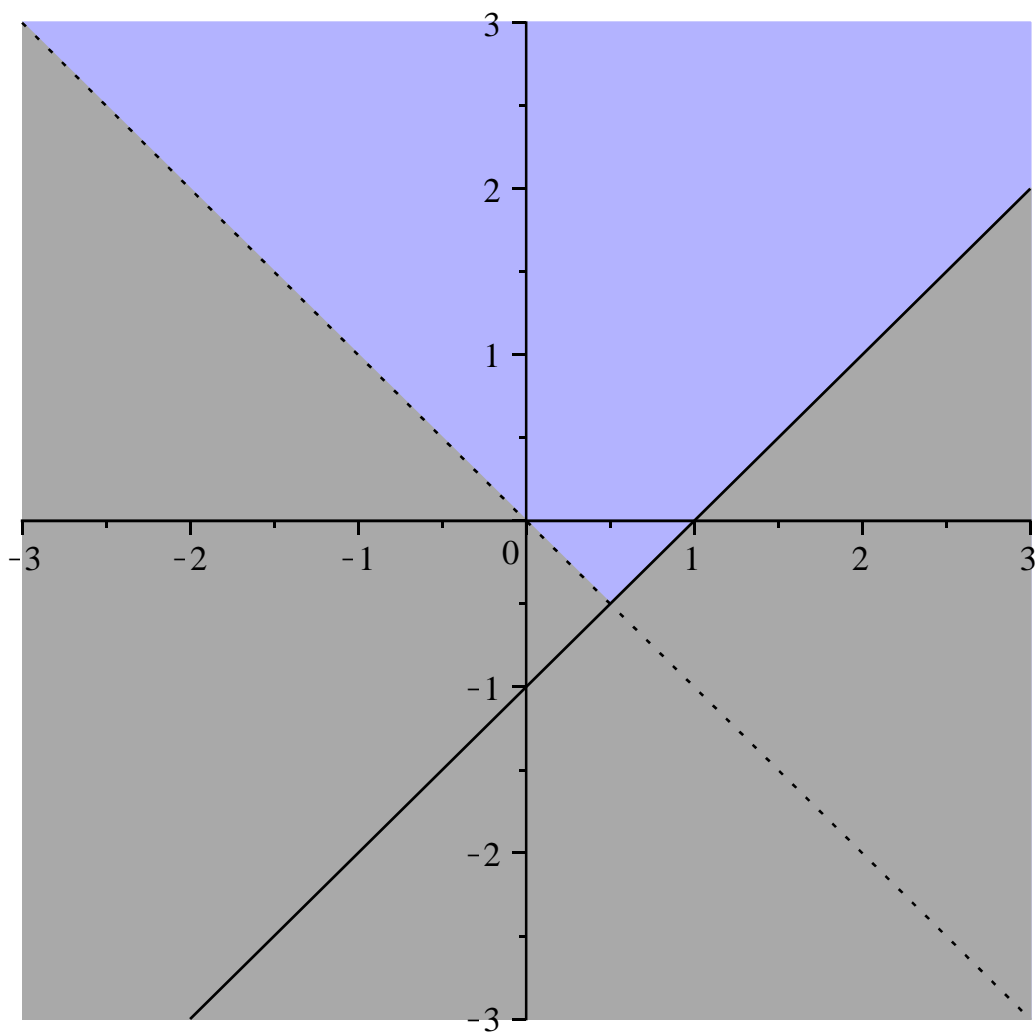
Les lignes "ouvertes" (open lines) qui sont les frontières des inégalités strictes;

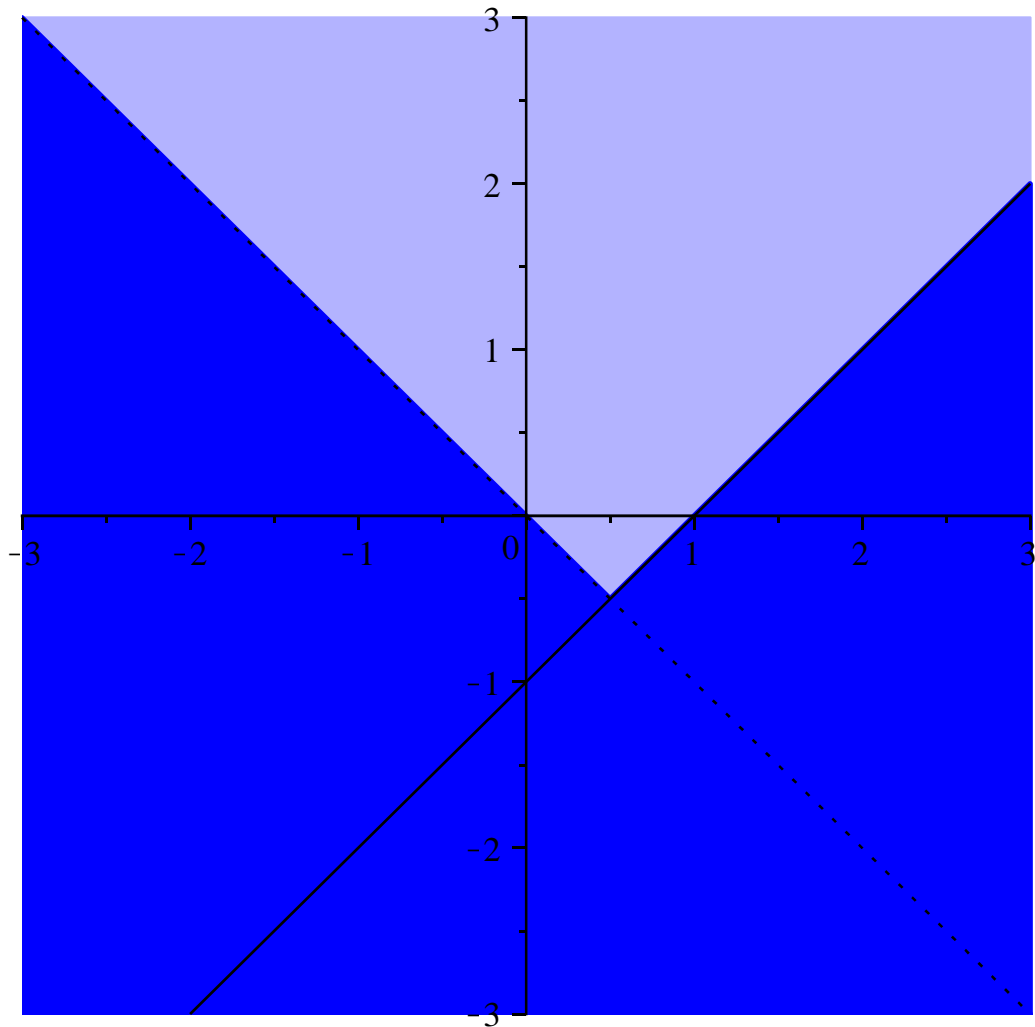
Les lignes "fermées" (closed lines) qui sont les frontières des inégalités larges et incluent les sommets.

Ces éléments sont paramétrables en options sous la forme : **optionsnom=(suite d'options de plot)**. Les **optionsnom** sont les suivants : **optionsfeasible**, **optionsexcluded**, **optionsopen**, et **optionsclosed**. Les options concernent surtout la colorisation des surfaces et des droites.

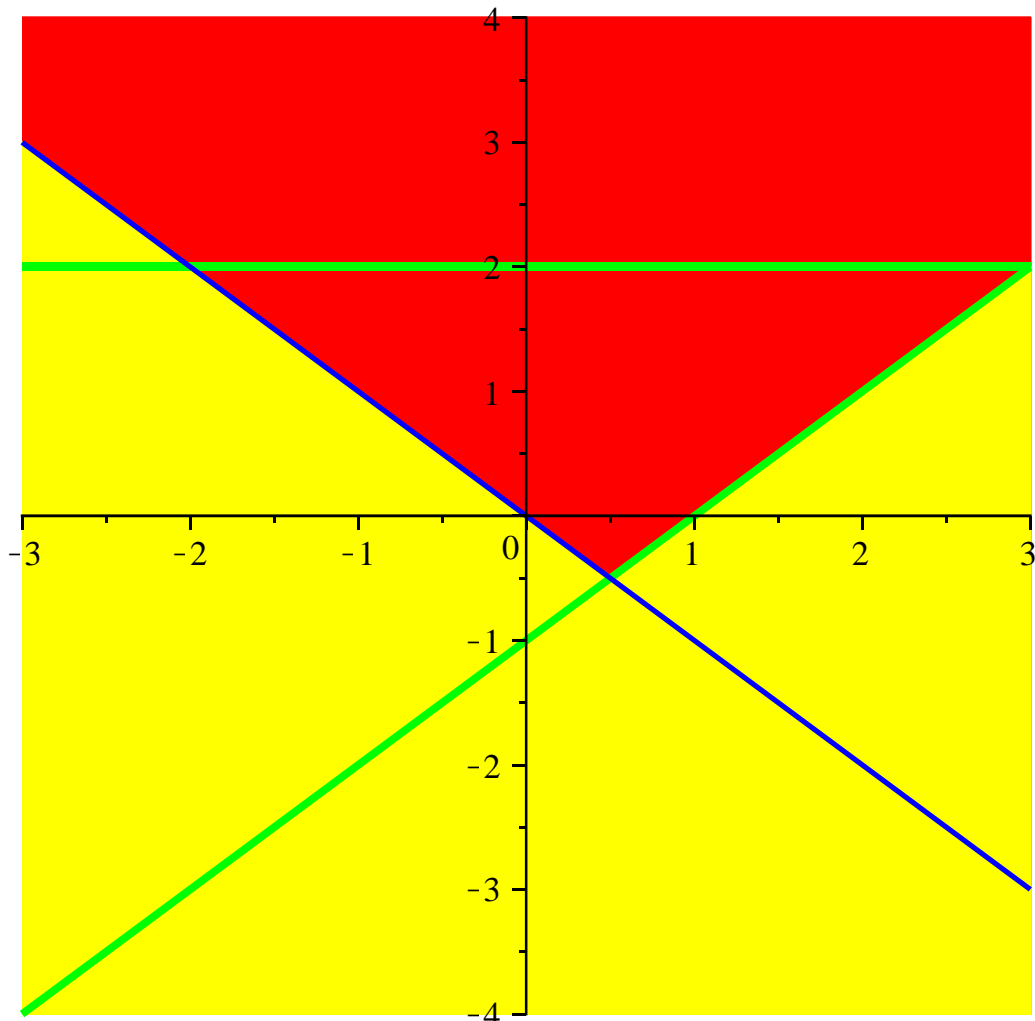
```
> restart;  
with(plots):#chargement du paquetage  
inequal( { x+y>0, x-y<=1}, x=-3..3, y=-3..3 );#aucune option  
inequal( { x+y>0, x-y<=1}, x=-3..3, y=-3..3,  
         optionsexcluded=(color=blue,thickness=2) );#colorisation  
         en bleu de la zone exclue
```







```
> inequal( { x+y>0, x-y<=1,y=2}, x=-3..3, y=-4..4,
  optionsfeasible=(color=red),
  optionsopen=(color=blue,thickness=2),
  optionsclosed=(color=green, thickness=3),
  optionsexcluded=(color=yellow) );#utilisation de toutes
les options
```



## Compléments indispensables

L'économiste sera parfois déçu par les résultats bruts qu'apporte la commande `solve`. La première raison est que Maple recherche les solutions d'une équation ou d'un système d'équations dans le corps des nombres complexes alors que les variables économiques sont "naturellement" réelles. La seconde tient au fait que les équations économiques sont souvent théoriques et font intervenir des paramètres soumis à des hypothèses que le logiciel ne prend pas en compte. Cette section donne des indications sur la manière de, sinon surmonter, du moins contourner ces difficultés. Elle présente également quelques astuces pour améliorer la recherche et le traitement des solutions.

### Suivre les étapes d'une résolution avec `infolevel[solve]`

En cas de besoin, on peut suivre les étapes de résolution par lesquelles passe `solve` en introduisant la commande `infolevel[solve]:=N`, où `N` est le niveau d'informations demandé (en pratique, `N=3` suffit).

```
> restart;
infolevel[solve]:=3;
solve(cos(x^3)-6*x=0,x);
```

```
infolevel_solve := 3
```

```
LinearEq: computing pseudo sub-resultant 20 16 x
```

```
PseudoResultant: computing pseudo resultant with lengths 20  
16 x
```

```
PseudoResultant: resultant computed and factored length = 29
```

$$\frac{1}{6} \cos(\text{RootOf}(216\_Z - \cos(_Z)^3))$$

## ▼ Récupération des solutions

La récupération des solutions est tout un art qu'il faut maîtriser car ces solutions feront généralement l'objet de manipulations immédiates : réinjection dans d'autres équations, exploitation à des fins graphiques, traitements analytiques, etc.. A bien des égards, il est préférable de procéder à des assignations en donnant à la ou les solutions un nom distinctif.

### ▼ Cas de la résolution d'une équation

Pour une équation, Maple renvoie une suite (sequence) de solutions. Il est aisé de récupérer chaque solution après avoir pris la précaution d'assigner l'output.

```
> restart;  
soluce:=solve(x^4-a=0,x);#les solutions sont assignées à  
l'identificateur : soluce  
sol_1:=soluce[1];  
sol_2:=soluce[2];  
sol_3:=soluce[3];  
sol_4:=soluce[4];
```

$$\text{soluce} := a^{1/4}, Ia^{1/4}, -a^{1/4}, -Ia^{1/4}$$

$$\text{sol}_1 := a^{1/4}$$

$$\text{sol}_2 := Ia^{1/4}$$

$$\text{sol}_3 := -a^{1/4}$$

$$\text{sol}_4 := -Ia^{1/4}$$

Quand on ignore a priori le nombre de solutions, on peut utiliser le petit programme suivant pour obtenir automatiquement l'assignement des solutions.

```
> for k from 1 to nops([soluce]) do  
sol[k]:=soluce[k]  
end do;
```

$$\text{sol}_1 := a^{1/4}$$

$$\text{sol}_2 := Ia^{1/4}$$

$$\text{sol}_3 := -a^{1/4}$$

$$\text{sol}_4 := -Ia^{1/4}$$

### ▼ Cas d'un système d'équations

Dans le cas de la résolution d'un système d'équations, Maple retourne une suite/sequence d'ensembles/sets donnant les solutions sous la forme d'équations. L'ordre des ensembles varie

d'un ordinateur à l'autre et même, pour un même ordinateur, d'une session l'autre. Quand il s'agit d'un système linéaire, ce n'est pas gênant puisqu'on s'attend à ce que la solution soit unique. En revanche, pour un système non linéaire, il faut procéder avec prudence.

Dans le cas d'un système linéaire, la commande **assign** assigne à chaque variable inconnue sa solution. Autrement dit, elle transforme les égalités figurant dans la solution renvoyée par **solve** en une assignation multiple.

```
> sys:={2*x+3*y=1,x+2*y=0};
var:={x,y};
sol:=solve(sys,var);
assign(sol);
      sys := {x + 2 y = 0, 2 x + 3 y = 1}
      var := {x, y}
      sol := {x = 2, y = -1}
```

La commande **assign** ne renvoie pas d'écho, mais l'assignation a bien été faite :

```
> x;y;
      2
      -1
```

Dans le cas d'un système non linéaire, la commande **assign** assigne à chaque variable inconnue la **dernière** solution figurant dans la solution renvoyée par **solve**.

```
> restart;sys:={x-1=0,y^2-x=0};var:={x,y};sol:=solve(sys,
var);assign(sol);x;y;
      sys := {x - 1 = 0, y2 - x = 0}
      var := {x, y}
      sol := {x = 1, y = 1}, {x = 1, y = -1}
      1
      -1
```

Autant dire que l'assignation est aléatoire. Le petit programme suivant, qui repose sur les propriétés de la commande **subs**, permet de récupérer systématiquement toutes les solutions, même si on ignore a priori leur nombre.

```
> for k from 1 to nops([sol]) do
sol_x[k],sol_y[k]:=subs(sol[k],x),subs(sol[k],y);
#assignation "multiple"
end do;
'sol_x[1]'=sol_x[1];#récupération et documentation d'un
résultat particulier
'sol_x[2]'=sol_x[2];#idem
'sol_y[1]'=sol_y[1];#idem
'sol_y[2]'=sol_y[2];#idem
      sol_x1, sol_y1 := 1, 1
      sol_x2, sol_y2 := 1, -1
      sol_x1 = 1
```

```

sol_x2 = 1
sol_y1 = 1
sol_y2 = -1

```

## ▼ Vérification des résultats

Pour vérifier qu'une solution est correcte, il suffit de l'injecter dans l'équation ou le système d'équations de départ et regarder si on obtient ou non une identité.

```

> restart; sys := {x-1=0, y^2-x=0}; var := {x, y}; sol := solve(sys, var);
#étape de résolution
for k from 1 to nops([sol]) do
sol_x[k], sol_y[k] := subs(sol[k], x), subs(sol[k], y);
end do; #étape de récupération des solutions
subs({x=sol_x[1], y=sol_y[1]}, sys); #étape de vérification
subs({x=sol_x[2], y=sol_y[2]}, sys);
      sys := {x - 1 = 0, y2 - x = 0}
      var := {x, y}
sol := {x = 1, y = 1}, {x = 1, y = -1}
      sol_x1, sol_y1 := 1, 1
      sol_x2, sol_y2 := 1, -1
      {0 = 0}
      {0 = 0}

```

La commande **evalb** (pour **evaluate as a boolean expression**) peut être utilisée conjointement avec **subs**. La réponse est **true** si le test de vérification est mené avec succès, **false** sinon.

```

> evalb(op(subs({x=sol_x[1], y=sol_y[1]}, sys)));
      true

```

Alternativement, on invoquera **eval**.

```

> eval(sys, {x=sol_x[1], y=sol_y[1]});
      {0 = 0}

```

Un thème connexe à la vérification est celui de l'exploration des solutions potentielles. Il est en effet possible que Maple ne retourne pas de solution ou retourne une partie des solutions alors qu'il se peut qu'elles existent ou qu'il en existe d'autres. Il affiche alors l'avertissement :

[Warning, solutions may have been lost](#). Pour confirmer, il faut intercaler la variable globale **\_SolutionsMayBeLost**; . Si la réponse est **true**, il se peut que des solutions existent. Sinon, il reste peu d'espoir.

```

> restart; eq := cos(sqrt(x)) - x^2 = 0; sol := solve(eq, x); #Maple ne
sait pas résoudre cette équation
_SolutionsMayBeLost; #Maple s'confirme que des solutions
peuvent exister

```

$$eq := \cos(\sqrt{x}) - x^2 = 0$$

[Warning, solutions may have been lost](#)

```
sol :=
```

## Incompatibilité de **solve** et **assume**

Soit l'équation simple  $x^2 - 3 = 0$ , dont on veut connaître la solution positive. Le premier réflexe est de poser une hypothèse sur  $x$ .

```
> restart;
  assume(x>0);
  solve(x^2-3=0,x);
```

Warning, solve may be ignoring assumptions on the input variables.

$$\sqrt{3}, -\sqrt{3}$$

L'exemple est éloquent : **la commande solve ne tient** - malheureusement - **jamais compte des contraintes posées sur la variable inconnue**. On peut lever cette limitation en demandant la résolution d'un système formé de l'équation de départ et de l'inéquation exprimant la contrainte.

```
> solve({x^2-3=0,x>0},x);
```

Warning, solve may be ignoring assumptions on the input variables.

$$\{x \sim \sqrt{3}\}$$

En revanche, **assume** prend en compte les hypothèses faites sur les paramètres quand l'équation à résoudre est symbolique.

```
> restart; assume(s>0,n>0,alpha>0,alpha<1);
  solve({s*k^(alpha)-n*k=0,k>0},{k}); #à la recherche de la
  golden rule d'accumulation du capital
  simplify(%,exp);
```

$$\left\{ k = e^{-\frac{\ln\left(\frac{s}{n}\right)}{\alpha-1}} \right\}$$

$$\left\{ k = \left(\frac{s}{n}\right)^{-\frac{1}{\alpha-1}} \right\}$$

## Résolution d'une équation dans $\mathbb{R}$

En économie, on recherche la ou les solutions réelles d'une équation ou d'un système d'équations. Maple, quant à lui, résout dans le corps des complexes et retourne des solutions sans intérêt pour l'économiste, qui est contraint de trier les résultats. Un moyen simple, mais faillible, de forcer le logiciel à ne retenir que les solutions réelles consiste à adjoindre à l'équation étudiée l'inéquation  **$x < \infty$**  et placer ces deux conditions entre accolades. Parfois, ce procédé marche, parfois il ne marche pas.

```
> restart;
  solve(x^3=1,x);
  solve({x^3-1=0,x<infinity},x);
  solve(x^4=1,x);
```

```
solve({x^4=1,x<infinity},x);
```

$$1, -\frac{1}{2} + \frac{1}{2} I\sqrt{3}, -\frac{1}{2} - \frac{1}{2} I\sqrt{3}$$
$$\{x=1\}$$
$$1, -1, I, -I$$
$$\{x=1\}, \{x=-1\}$$

Le programme de tri suivant conserve les solutions réelles en écartant les solutions dont la partie imaginaire est différente de 0. Il n'est utile que si Maple sait expliciter les solutions avec la commande **allvalues** comme dans l'exemple qui suit. Ce n'est malheureusement pas toujours le cas.

```
> sol:=map(allvalues,{solve(x^3-1=0,x)});#reprise de l'exemple
ci-dessus
#début du programme de tri
S:={}:
for j from 1 to nops(sol) do
if Im(sol[j])=0 then S:=S union {sol[j]} else S:=S end if;
end do;
S;
```

$$sol := \left\{ 1, -\frac{1}{2} - \frac{1}{2} I\sqrt{3}, -\frac{1}{2} + \frac{1}{2} I\sqrt{3} \right\}$$
$$\{1\}$$

Dans l'exemple suivant, **allvalues** ne parvient pas à expliciter les solutions. Les tests ne peuvent donc pas aboutir.

```
> sol:=map(allvalues,{solve(x^5-2*x^3+10=0,x)});
S:={}:
for j from 1 to nops(sol) do
if Im(sol[j])=0 then S:=S union {sol[j]} else S:=S end if;
end do;
S;
sol := {RootOf(_Z^5 - 2 _Z^3 + 10, index=1), RootOf(_Z^5 - 2 _Z^3 + 10, index=2),
RootOf(_Z^5 - 2 _Z^3 + 10, index=3), RootOf(_Z^5 - 2 _Z^3 + 10, index=4),
RootOf(_Z^5 - 2 _Z^3 + 10, index=5)}
{RootOf(_Z^5 - 2 _Z^3 + 10, index=3)}
```