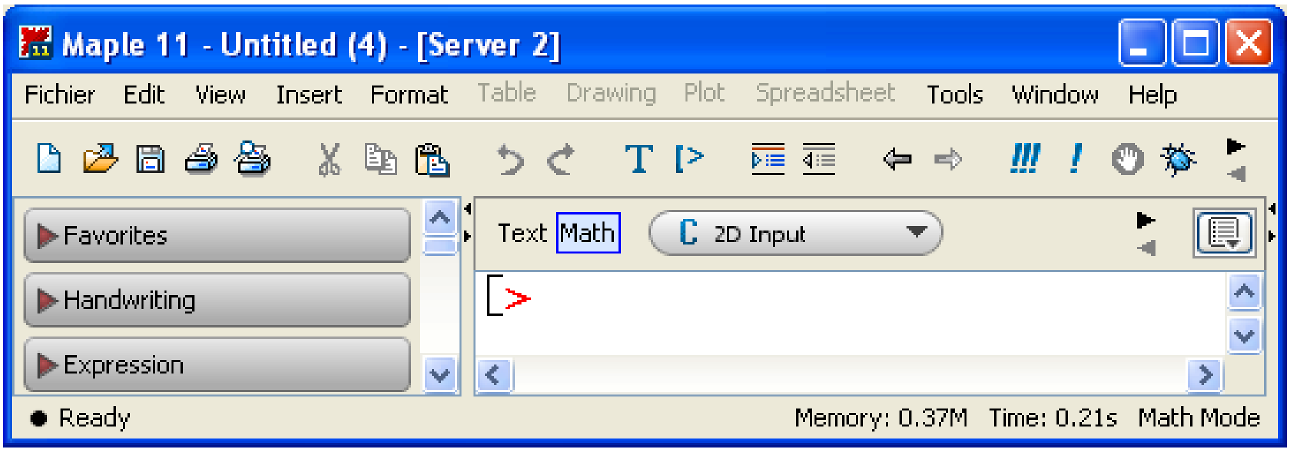
TP Maple 1 | Les premiers pas. Fonctions. Expressions

Maple est né en 1981 au sein du Symbolic Computation Group de l'université de Waterloo dans l'Ontario (Canada). Le logiciel fut commercialisé dès 1985 par la firme Maplesoft (voir le site www.maplesoft.com pour de plus amples informations). La version la plus récente du logiciel est Maple 11 et date de 2007.

1. Découverte du système Maple

1.1. Ouverture d'une session

On débute une *session* de Maple en ouvrant une nouvelle *feuille de calcul* (a Maple worksheet) dans le menu déroulant **File** au moyen de l'onglet **New** en selectionnant Worksheet Mode. La session peut alors commencer, le *prompt* $>$ apparaît et le curseur | clignote en attendant les commandes.



Ces dernières sont exécutées après validation par la touche **Entrée**. Dans l'exemple suivant nous calculons le carré du nombre complexe $1 + i$.

```
> (1+I)^2;                    #  Attention !  Majuscule I pour le nombre complexe i
```

$$2I$$

La partie de la ligne après # est ignorée par Maple et peut servir au programmateur de laisser des notes. Une commande Maple finit soit par un : soit par un ; comme ci-dessus. Si on finit par le double-point la commande est prise en compte mais le résultat n'est pas affiché.

Si on souhaite écrire plusieurs lignes consécutives dans le même prompt on peut passer à une nouvelle ligne par la combinaison de touches **Shift+Entrée**.

```
> exp(Pi*I);
  sqrt(5);
  evalf(sqrt(5));             #  évaluation, arrondie à par défaut 10 chiffres
  evalf(sqrt(5),3);           #  arrondie à 3 chiffres
```

$$-1$$

$$\sqrt{5}$$

$$2.236067977$$

$$2.24$$

1.2. Utilisation du symbole % lors d'un calcul

Le symbole pour-cent %, employé comme argument, désigne le dernier résultat (*chronologiquement !*) calculé par Maple. De même %% désigne l'avant-dernier résultat calculé et %%% l'antépénultième. Ce procédé de rappel s'arrête là. Cette technique évite l'emploi du copier-coller, ce qui est parfois appréciable :

```
> 8; %*6; (2+%%)*%; (20+%)*%%+%%%;
```

$$8$$

$$48$$

$$480$$

$$24008$$

1.3. Ordonner son travail en sections

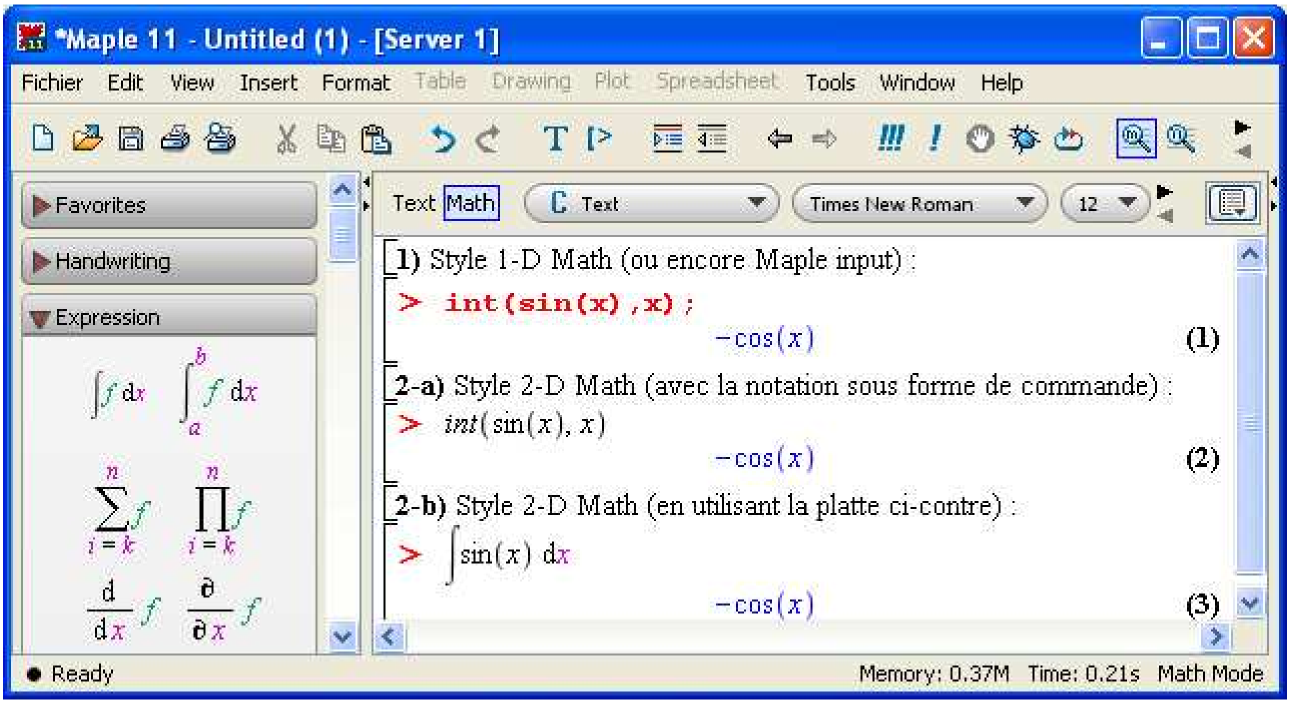
Les calculs et leurs résultats peuvent devenir assez longs. Il est conseillé de structurer son travail (par exemple exercice par exercice) dans des SECTIONS qu'on peut ensuite cacher ou faire réapparaître par un simple clic. Rendez-vous dans le menu **Insert/Sections**.

1.4. L'aide de Maple

L'utilisateur ne doit aucunement connaître par cœur toutes les fonctionnalités de Maple. A l'oral des concours, il aura le droit de consulter l'aide du logiciel. Il devra se familiariser avec la recherche d'outils en Anglais. Dans le menu déroulant **Help**, il trouvera la rubrique **Maple Help** qui lui permettra de rechercher les entrées de l'aide correspondant à sa demande.

1.5. Les styles d'entrée

Depuis la version 10 de Maple, deux styles différents de notations sont disponibles pour les *entrées*. L'ancien style, appelé le Maple Input (ou 1-D Math), est propre au logiciel contrairement au style 2-D Math qui donne *en plus* accès à la notation standard mathématique. En guise d'exemple, voici le calcul des primitives du sinus programmé dans les deux styles.



Afin d'entraîner le lecteur à la rigueur syntaxique (indispensable lorsque l'on aborde des langages plus puissants (comme **C**, **Fortran**, etc.) et pour des raisons de compatibilité avec les versions antérieures de Maple, nous choisirons de programmer dans le style Maple Input dans la suite de ce cours.

L'utilisateur pourra fixer cette configuration en changeant les paramètres du logiciel dans la suite d'onglets Tools - Options... - Display : *Maple Notation* pour input display et *2-D math Notation* pour output display.

1.6. Feuille de calcul Vs. noyau d'une session

Après lancement de l'application Maple est automatiquement créée dans l'ordinateur une zone de mémoire, extensible au fur et à mesure des besoins, pour stocker les données propres au créateur de la session (noms des variables, expressions, etc.) pendant sa période d'utilisation du logiciel : c'est ce qu'on appelle le noyau (*the kernel*). Il est important de distinguer le contenu du noyau de la feuille de calcul : le noyau est toujours cohérent alors que la feuille de calcul peut ne pas l'être... Examinons le cas où l'utilisateur revient sur ses pas et modifie le début de sa feuille de calcul sans en modifier la fin : dans cette situation, la feuille de calcul, lue dans son intégralité, sera *mathématiquement* incohérente. Considérons la séquence suivante :

```
> a:=4:  # ligne validée par un return
> a^2;   # ligne validée par un return

                                    16
```

Il faut la comprendre de la manière suivante : la case-mémoire de l'ordinateur désignée par la lettre a contient la valeur numérique 4. Lorsque l'on demande le calcul de a^2, le logiciel retourne la valeur $4^2 = 16$. Si l'on modifie la première ligne de cette séquence par $a := 3$ on obtient après validation par la touche *Return* :

```
> a:=3:   # ligne re-validée par un return
> a^2;    # ligne non re-validée

                                    16
```

Si l'on enrigistrait la feuille de calcul à cet instant, il pourrait sembler que le logiciel ne sait pas calculer... Pour obtenir le résultat escompté, il faut valider à nouveau la ligne concernant a^2 au moyen de la touche *Return* du clavier :

```
> a:=3:
> a^2;  # ligne re-validée par un return

                                    9
```

Ouf ! On en déduit la règle fondamentale suivante :

Modification d'une ligne de commandes

Toute modification d'une ligne de commandes doit s'accompagner d'une validation (par la touche *Return*) des lignes suivantes sous peine d'ambiguité.

Lorsque l'on ouvre une ancienne session, il ne reste que la trace de la feuille de calcul : le noyau initial (conçu à la création de la feuille) a été effacé. Si l'on veut le reconstituer, il faudra resaisir chacune des lignes de commandes de la feuille en question. C'est ce qu'on le fera – en cas de doute – lors de l'ouverture d'une session antérieure.

2. Les variables

Une variable est une chaîne de caractères associée à un emplacement de la mémoire utilisée par Maple [1]. Cette « *case de mémoire* » peut contenir une donnée choisie par l'utilisateur. Toutes les chaînes de caractéres ne sont pas permises, certaines sont réservées (comme **Pi** qui contient une valeur approchée du nombre réel π ou encore **I** qui désigne le nombre complexe i). Mis à part quelques mots réservés [2] On retiendra la règle suivante :

1. Comme une adresse est associée à un bâtiment . . .
2. Par exemple : **Pi**, **I**, **O**, **D** et **gamma**.

Noms de variables sous Maple

Les noms de variable doivent commencer par une lettre mais peuvent ensuite contenir des chiffres ainsi que le caractère « _ » . Le logiciel distingue les minuscules des majuscules.

2.1. Assignations et évaluations

Par exemple, on peut *ranger* la valeur 8 dans la case de mémoire se trouvant à l'*adresse* a ; on utilise pour cela l'opération d'*affectation* suivante : **a :=8**. La variable a est dite *affectée* ou encore *que l'on a affecté la valeur* 8 *à la variable* a. Dans la suite des calculs, Maple remplacera systématiquement la variable a par son contenu 8. On peut alors *représenter* la variable a comme un « objet » *pointant* vers la valeur 8 :

$$a \;\rightarrow\; 8$$

On peut accéder (on dit parfois *évaluer*) au *contenu* d'une variable par simple appel de son nom.

```
> a:=8;
```

$$a := 8$$

```
> a;
```

$$8$$

On peut également *changer le contenu d'une variable* par une nouvelle affectation (qui « écrase » l'ancien contenu !).

```
> a:=12:a;
```

$$12$$

Des affectations simultanées sont possibles.

```
> b,c,d:=1,2,5:d,c,b;
```

$$5, 2, 1$$

L'utilisateur peut *empêcher l'accès en enregistrement* à une variable au moyen de la commande **protect('*nom de la variable*')**.

```
> a:=5:protect('a'):a:=11;
Error, attempting to assign to `a` which is protected
> a;
```

$$5$$

L'opération inverse se fait avec la commande **unprotect('*nom de la variable*')**.

```
> unprotect('a'):a:=11;
```

$$11$$

Avec la commande **restart** toutes le variables sont vidées de leurs affectations. On peut aussi vider une variable seule par la commande ***nom de la variable*='*nom de la variable*'**.

Evidemment l'utilisation de variables est très utile lorsqu'on ne veut pas trainer des expressions longs et compliquées. Testez le résultat des commandes suivantes.

```
> E:=solve(a*x^2+b*x+c=0,x);
  x1:=E[1]; x2:=E[2];
  x1*x2; expand(x1*x2);
```

Recommandation

On commencera chaque nouvel exercice ou thème de calcul par une commande **restart**.

3. Les fonctions mathématiques

Maple permet à l'utilisateur de créer ses propres fonctions au moyen d'une syntaxe inspirée de celle des mathématiques.

Définition d'une fonction f

$f := x$ -> *expression dépendant de* x

L'utilisateur accédera aux valeurs de la fonction f en utilisant la syntaxe usuelle des mathématiciens.

```
> f:=x->x+sin(x); f(Pi); evalf(f(Pi),3);
```

$$f := x \to x + \sin(x)$$

$$\pi$$

$$3.14$$

Remarquons que la variable x est *muette*, on peut la remplacer par n'importe quelle autre lettre.

```
> f:=x->sin(x)^2; g:=y->cos(y)^2; h:=f+g; h(t); simplify(h(t));
```

$$f := x \to \sin^2(x)$$
$$g := y \to \cos^2(y)$$
$$h := f + g$$
$$\sin^2(t) + \sin^2(t)$$
$$1$$

```
> f:=u->u+sin(u): evalf(f(Pi));
```

$$3.141592654$$

Si f est une fonction, « $f(x)$ » sera considéré comme une expression par **Maple**. Cette expression peur très bien dépendre d'autres paramètres que x.

```
> f:=u->sin(t*u): t:=1: evalf(f(Pi));
```

$$0$$

```
> t:=0.5: f(Pi); t:=1.5: f(Pi);
```

$$1, -1$$

3.1. Fonctions usuelles

Maple dispose d'une librairie de fonctions usuelles prédéfinies. En voici un aperçu non exhaustif ...

Fonctions usuelles sous Maple

- L'exponentielle : **exp** ;
- La racine carrée : **sqrt** ;
- Le logarithme népérien : **ln** *ou* **log** ;
- $\sqrt[n]{x}$ pour $x \in \mathbb{R}$ et n impair : **surd(x,n)** ;
- Fonctions triginométriques : **cos, sin, tan, cot** ;
- Fonctions hyperboliques : **cosh, sinh, tanh** ;
- Réciproques des fonctions triginométriques : **arccos, arcsin, arctan** ;
- Réciproques des fonctions hyperboliques : **arccosh, arcsinh, arctanh**.

On se souviendra que les nombres réels e et π sont respectivement codés **exp(1)** et **Pi** sous Maple.

3.2. Opérations sur les fonctions

Maple est capable d'effectuer les opérations algébriques élémentaires sur les fonctions telles que la somme, le produit, etc.

```
> restart: f:=x->x: g:=x->sin(x): h:=f+g, h(Pi), i:=f*g, i(Pi);
```

$$h := f + g, \pi, i := fg, 0$$

Opérations sur les fonctions

- **f + g** : somme de f et g ;
- **f * g** : produit de f et g ;
- **f / g** : quotient de f par g ;
- **f @ g** : composée $f \circ g$;
- **f @@ n** : composée $f \circ \cdots \circ f$ (n fois).

```
> restart: f:=x->cos(x): h:=f@@3; h(x);
```

$$h := f^{(3)}$$

$$\cos(\cos(\cos(x)))$$

On voit que, pour noter une composée n-fois, Maple utilise des parenthèses – malheureusement cela peut porter confusion avec la dérivée n-ième.

Evidemment Maple trace aussi des courbes des fonctions (voir commande **plot** dans l'aide).

4. Manipulation des expressions usuelles

Voici en vrac quelques commandes permettant le calcul des expressions sous Maple.

```
> cos(2*x):expand(%);
```

$$2\cos(x)^2 - 1$$

```
> (a+b)^4:expand(%);
```

$$a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

```
> expand(exp(a+ln(b)));
```

$$e^a b$$

```
> (X+3)/(X^2+4*X+3):normal(%);
```

$$\frac{1}{X+1}$$

```
> cos(x)^3:combine(%);
```

$$\frac{1}{4}\cos(3x) + \frac{3}{4}\cos(x)$$

```
> simplify(exp(a+ln(b*exp(c))));
```

$$be^{a+c}$$

```
> f := x*(x+1)+y*(x+1); collect(f,x)
```

$$f := x(x+1) + y(x+1)$$
$$x^2 + (1+y) * x + y$$

```
> P:=1+X^8+X^9+X+6*X^2+6*X^3+2*X^9+5*X^2-6*X-2; sort(P);
```

$$P := 3X^9 + X^8 + 6X^3 + 11X^2 - 5X - 1$$
$$3X^9 + X^8 + 6X^3 + 11X^2 - 5X - 1$$

```
> factor(r^3-3*r^2*s+3*r*s^2-s^3);
```

$$(r-s)^3$$

Si on ajoute l'option **radical**, **trig** ou **exp** à la commande **simplify** alors on restreint la simplification :

```
> a:=16^(1/2)*(cos(x)^2+sin(x)^2): simplify(a,radical); simplify(a,trig);
```

$$4\cos(x)^2 + 4\sin(x)^2$$
$$\sqrt{16}$$

Manipulation des expressions usuelles

▷ **expand** : développe par distributivité des expressions polynomiales, calcule les fonctions circulaires en nx en fonction de leur valeur en x, applique les règles de calcul sur les logarithmes et les exponentielles ;

▷ **factor** : factorise lorsque c'est possible ;

▷ **normal** : simplifie les fractions rationnelles ;

▷ **combine** : linéarise s'utilise avec différentes options (cf. l'aide de **Maple**). Permet par exemple d'exprimer le sinus et le cosinus à l'aide de l'exponentielle complexe (ie applique les formules d'Euler !) ;

▷ **convert** : pour faire passer une expression d'une forme à une autre (par exemple de l'écriture décimale à l'écriture binaire pour les entiers). Voir l'aide en ligne pour les autres fonctionnalités ;

▷ **simplify** : permet de simplifier des expressions algébriques. Cf. l'aide en ligne ;

▷ **collect** : effectue des regroupements ;

▷ **sort** : écrit un polynôme selon les puissances décroissantes.

La commande **convert** permet de convertir une expression en une autre. Elle permet par exemple de *transformer* un développement de Taylor en une expression du type *polynom* en extrayant sa partie principale. Elle permet également la conversion des entiers d'une base de numération à une autre, la décomposition en éléments simples de sfractions rationnelles : on l'aura compris, la commande **convert** offre de très nombreuses fonctionnalités, on consultera l'aide en ligne afin d'explorer l'étendue de ses possibilités.

```
> convert(9, binary);
```

$$1001$$

```
> convert(1.892,fraction);
```

$$\frac{473}{250}$$

```
> f := sinh(x)+sin(x): convert(f, exp);
```

$$\frac{1}{2}e^{x} - \frac{1}{2}e^{-x} - \frac{1}{2I}(e^{Ix} - e^{-Ix})$$

5. Exercices

Exercice 1.

Effectuer, afficher et commenter la séquence suivante :

```
> b:=cos(a): a:=Pi/2: b;
```

$$0$$

```
> a:=2*Pi/3: b;
```

$$-\frac{1}{2}$$

```
> c:=cos(a): a:=3*Pi/4: b, c;
```

$$-\frac{1}{2}\sqrt{2},\ -\frac{1}{2}$$

Exercice 2.

On part de deux variables

```
> a:=1: b:=2:
```

On souhaite échanger le contenu de ces deux variables. Comment s'y prendre ?

Exercice 3.

En utilisant l'aide en ligne, trouver comment calculer le module d'un nombre complexe.

Exercice 4.

En utilisant l'aide en ligne, trouver comment dériver vingt fois par rapport à x l'expression $\ell n(1+x^2)$.

Exercice 5.

A l'aide de Maple tracer les courbes des trois fonctions définies par $f(x) = x^2$, $g(x) = -x^2$ et $h(x) = x^2\sin(1/x)$ dans un même dessin.

Exercice 6.

Vérifier avec Maple que le produit de 4 nombres entiers en progression arithmétique auquel on ajoute la puissance quatrième de la raison est toujours un carré.

Exercice 7.

Déterminer une équation à coefficients entiers du septième degré dont $2\cos(\pi/7)$ est solution. On pourra partir de **expand(cos(7*t))**.

Exercice 8.

Simplifier

$$\frac{\dfrac{\frac{1}{a}+\frac{1}{b}}{\frac{1}{a}-\frac{1}{b}}+\dfrac{\frac{1}{a}-\frac{1}{b}}{\frac{1}{a}+\frac{1}{b}}}{\dfrac{\frac{1}{a}+\frac{1}{b}}{\frac{1}{a}-\frac{1}{b}}-\dfrac{\frac{1}{a}-\frac{1}{b}}{\frac{1}{a}+\frac{1}{b}}}.$$

Exercice 9.

Trouver l'expression de

$$\cos^3(2x)\sin^2(x)$$

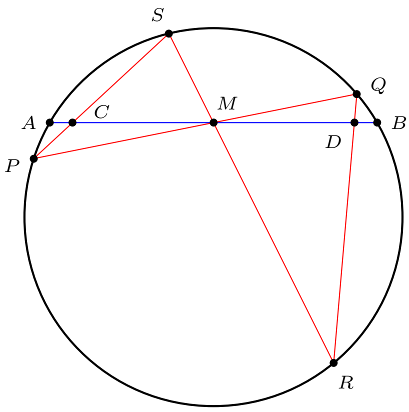
en fonction de $t=\tan(x/2)$. On obtiendra une fraction rationnelle en la variable t que l'on factorisera au maximum.

Exercice 10.

On définit la suite $(u_n)_{n\in\mathbb{N}}$ par $u_0=1.5$ et $\forall n\geqslant 0,\ u_{n+1}=\sin(u_n)$. Calculer u_{2010}.

Exercice 11.

Soit $\mathscr{C}$ un cercle, A, B deux points distincts sur $\mathscr{C}$ et M le milieu de la corde $[AB]$. Soient $[PQ]$ et $[SR]$ deux autres cordes passant par M. On note C (resp. D) le point d'intersection de $[AB]$ avec $[PS]$ (resp. $[RQ]$).



Démontrer que M est aussi le milieu de $[CD]$.
(Faites une preuve par un calcul analytique en coordonnées ; consacrez-vous à l'idée de la preuve et déléguez à Maple le travail calculatoire.)

Solutions des exercices

Solution 1.

Il semble contradictoire que dans le dernière ligne b et c sont des valeurs distinctes. Mais il faut tenir compte que b est introduit dès le départ comme une expression qui dépend de la variable a ce qui n'est pas le cas de c. En effet, au moment où on a défini c le nombre a était une valeur réelle et non une variable, donc c est un nombre et pas une expression.

Solution 2.

```
> a:=1; b:=2; help:=a;  a:=b; b:=help; help:='help';
```

Solution 3.

```
> abs(1-I);
```

Solution 4.

```
> diff(ln(1+x^2), x$20);
```

Solution 5.

On utilise une liste d'expressions :

```
> L:=[x^2,-x^2,x^2*sin(1/x)]: plot(L,x=-0.01..0.01);
```

Ou on utilise une liste de fonctions :

```
> f:=x->x^2: g:=-f: h:=x->x^2*sin(1/x): L:=[f,g,h]: plot(L,-0.01..0.01);
```

Solution 6.

```
> factor(n*(n+k)*(n+2*k)*(n+3*k)+k^4);
```

Solution 7.

```
> expand(cos(7*t));
```

$$64\cos(t)^7 - 112\cos(t)^5 + 56\cos(t)^3 - 7\cos(t)$$

```
> expand(cos(7*arccos(x/2)))+1;  #C'est le polynôme cherché.
```

$$\frac{1}{2}x^7 - \frac{7}{2}x^5 + 7x^3 - \frac{7}{2}x + 1$$

Solution 8.

```
> A:=1/a+1/b: B:=1/a-1/b: A1:=A/B+B/A: B1:=A/B-B/A: simplify(A1/B1);
```

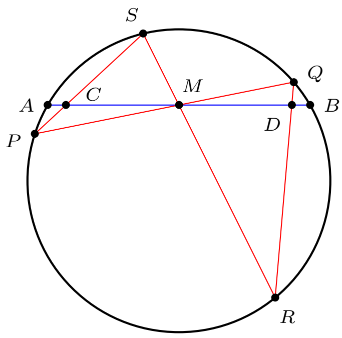
Solution 9.

```
> factor(expand(cos(4*arctan(t))^3*sin(2*arctan(t))^2));
```

Solution 10.

```
> ?composition: (sin@@2010)(1.5);
```

Solution 11.




```
> # Repère orthonormé tel que M est l'origine est (AB) est l'axe des abscisses. Unite = rayon du cercle
  # Equation du cercle C: x^2+(y-m)^2=1
  # Equation des droites  (AB): y=0    (PQ): x=ay   (RS): x=by
  # Calcul des coordonnées de P ou Q en fonction de a :
  S:=solve((a*y)^2+(y-m)^2=1, y);
  yQ:=S[1]; yP:=S[2]; xQ:=a*yQ; xP:=a*yP;
> # Calcul des coordonnées de R ou S en fonction de a :
  yS:=solve((b*y)^2+(y-m)^2=1, y)[1]; yR:=solve((b*y)^2+(y-m)^2=1, y)[2]; xS:=b*yS; xR:=b*yR;
> # Le point D(xD;0) est caractérisé par une condition d'alignement :
  xD:=solve((xQ-x)/(xQ-xR)=(yQ-0)/(yQ-yR),x);
> # De même pour le point C(xC;0) :
  xC:=solve((yS-yP)/(xS-xP)=(yS-0)/(xS-x),x);
> # On montre que l'origine M est le milieu de [CD]
  simplify(xC+xD);
```