

# Aide-mémoire MAPLE

(Version 23 février 2005)

© Pierre Lantagne

<http://math.cmaisonneuve.qc.ca/plantagne>  
[plantagne@cmaisonneuve.qc.ca](mailto:plantagne@cmaisonneuve.qc.ca)

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>abs</code>	Fonction valeur absolue	<code>abs(x^2+5*x-3);</code>
<code>add</code>	Pour additionner numériquement une séquence de nombres.	<code>add(i^2,i=1..5);</code> <code>add(sqrt(i),i=1..10);</code>
<code>alias</code>	Pour définir des noms raccourcis.	<code>alias(C=binomial);</code> <code>C(5,8);</code> <code>alias(C=C); # Pour annuler l'alias</code>
<code>algsubs</code>	Pour substituer syntaxiquement chaque occurrence d'une expression A par une expression B dans une autre expression. Cette macro-commande généralise la macro-commande <code>subs</code> avec l'exception qu'elle ne permet pas de substitutions simultanées. (Voir <code>subs</code> )	<code>E:=1+a+b+c;</code> <code>subs(a+b+c=d,E);</code> <code>algsubs(a+b+c=d,E);</code> <code>E:=x^3;</code> <code>subs(x^2=1-b^2,E);</code> <code>algsubs(x^2=1-b^2,E);</code>
<code>allvalues</code>	Pour imposer l'évaluation des racines d'une équation formulée en termes de <code>RootOf</code> .  Dans le cas d'une équation polynomiale de degré $\leq 4$ , les racines sont affichées symboliquement.	<code>RootOf(_Z^2+5*_Z+5=0);</code> <code>allvalues(RootOf(_Z^2+5*_Z+5));</code> <code>E:=x^3+12*x^2-5*x-12=0;</code> <code>Racines:=[allvalues(RootOf(E))];</code> <code>evalc(Racines);</code> <code>evalf(evalc(Racines));</code> <code>evalf(Racines);</code> <code>evalf(Racines,20);</code>
<code>arcsin, arccos, arctan, arccot, arcsec, arccsc</code>	Fonctions trigonométriques réciproques	<code>?invtrig</code> <code>arcsin(1/2);</code> <code>plot([x,arcsin(x),x=-1..1]);</code> <code>plot([sin(x),x,x=-Pi/2..Pi/2]);</code>
<code>assign</code>	Pour assigner le membre de droite au membre de gauche d'une égalité.	<code>Sol:={x=4},{x=-2}; assign(Sol);</code> <code>x;y; # x et y ne sont plus libres</code>
<code>assume</code>	Pour conditionner une variable.	<code>assume(n, odd);</code> <code>cos(n*Pi);</code>
<code>about</code>	Pour obtenir les propriétés d'un objet ou d'une variable conditionnée.	<code>assume(y, nonneg);</code> <code>about(y);</code>
<code>combine</code>	Pour réduire un expression à l'aide de domaines de règles de simplification.	<code>f:=exp(x)^2*exp(y);</code> <code>combine(f,exp);</code>
<code>collect</code>	Pour structurer une expression en pseudo-polynôme sur la base d'une de ses indéterminées.	<code>f:=a*ln(x)-ln(x)*x-x;</code> <code>collect(f,ln(x));</code>

tableau 5: Macro-commandes MAPLE

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>convert</code>	Pour convertir un objet d'un certain type vers un autre.	<code>convert(30*degrees,radians);</code> <code>convert(3.14159,fraction);</code> <code>convert(sin(Pi/120),radical);</code>
<code>cos</code>	Fonction trigonométrique cosinus	<code>cos(Pi/5);</code>
<code>csc</code>	Fonction trigonométrique cosécante	<code>csc(Pi/5);</code>
<code>denom</code>	Pour pointer vers le dénominateur d'une fraction ou le dénominateur commun d'une somme de fractions.	<code>F:=1/x-2/x^2+3/(1+x); denom(F);</code>
<code>diff</code> (Voir ?D)	Pour obtenir la dérivée d'une expression.	<code>diff(x^2*cos(x),x);</code> <code>diff(x*cos(x),x\$3);</code> # dérivée successive d'ordre 3
<code>display</code>	Pour superposer différentes structures graphiques dans un même graphique. Cette macro-commande appartient à l'extension <code>plots</code> .	<code>with(plots,display,polarplot);</code> <code>C1:=plot([1+cos(t),sin(t),t=0..2*Pi]);</code> <code>C2:=polarplot(2*cos(3*t),t=-Pi..Pi);</code> <code>display([C1,C2],scaling=constrained);</code>
<code>dsolve</code>	Pour résoudre une équation différentielle ordinaire.	Équation: $y(t)*diff(y(t),t)+sin(t)=0$ ; <code>dsolve(Équation,y(t));</code>
<code>eval</code>	Pour évaluer une formule exactement.	<code>eval(1+x^3,x=sqrt(5));</code> <code>eval(x^2*y,[x=2,y=3]);</code>
<code>evalb</code>	Pour obtenir la valeur de vérité d'une proposition. Attention, la macro-commande ne fait aucune simplification et détermine la valeur de vérité de manière informatique. Il est alors possible que le résultat soit faux bien que la réponse devrait être vraie.	<code>evalb(1=1.0);</code> <code>evalb(sqrt(5)&gt;2);</code> <code>evalb(evalf(sqrt(5))&gt;2);</code> <code>evalb((sin^2)(x)+(cos^2)(x)=1);</code>
<code>evalc</code>	Pour obtenir la forme algébrique d'un nombre complexe.	<code>evalc(polar(-3,Pi/5));</code> <code>evalc(exp(Pi/3*I));</code> <code>evalc(ln(2+I));</code>
<code>evalf</code>	Pour obtenir une approximation d'une valeur exacte (réelle ou complexe).	<code>evalf(Pi/2);</code> <code>evalf(ln(2+I),20);</code>
<code>exp</code>	Fonction exponentielle de base $e$	<code>exp(x^2+1);</code>
<code>expand</code>	Pour développer une expression, en développant la distributivité de la multiplication sur l'addition ou en développant des formules trigonométriques et autres.	<code>expand((x+1)/(x+2));</code> <code>expand(cos(2*x));</code> <code>expand(exp(5*x));</code> <code>expand(ln(a/b));</code> <code>assume(a,positive,b,positive);</code> <code>expand(ln(a/b));</code>
<code>factor</code>	Pour factoriser toute expression pouvant être vue comme un polynôme ou une fraction rationnelle en ses indéterminées.	<code>E:=2*cos(x)^2*exp(2*x)-exp(x)*\cos(x)^2+cos(x)*exp(3*x);</code> <code>factor(expand(E));</code>
<code>fsolve</code>	Pour résoudre par approximation numérique une équation ou un système d'équations. On peut lui indiquer un intervalle de recherche ou lui indiquer une valeur approximative de départ. Ces informations peuvent être obtenues en traçant un graphique.	<code>Eq:=cos(x)-2*x^3+5*x^2+x-5;</code> <code>fsolve(Eq,x=-1);</code> <code>fsolve(Eq,x=1);</code> <code>fsolve(Eq,x=2);</code>

tableau 5: Macro-commandes MAPLE

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>ifactor</code>	Pour obtenir la factorisation complète d'un nombre entier ou d'un nombre rationnel.	<code>ifactor(20!);</code> <code>ifactor(57/25546);</code>
<code>interface</code>	Pour modifier les paramètres de l'interface usager.	<code>interface(imaginaryunit=i);</code> <code>i^2;</code>
<code>Im</code>	Pour obtenir la partie imaginaire d'un nombre complexe.	<code>Im(-3-3*I);</code> <code>Im(polar(-5,Pi/4));</code> <code>Im(exp(5*i*Pi/4));</code>
<code>implicitdiff</code>	Pour dériver une fonction définie implicitement dans une formule.	<code>Éq:=x^2*y+x*y^3=1;</code> <code>`y`:=implicitdiff(Éq,y,x);</code>
<code>implicitplot</code>	Pour tracer le lieu géométrique définie implicitement par une équation. Cette macro-commande appartient à l'extension <code>plots</code> .	<code>Éq:=(x+2)^2+(y-1)^2=9;</code> <code>with(plots,implicitplot);</code> <code>implicitplot(Éq,x=-5..2,y=-2..4);</code>
<code>int</code>	Pour calculer une intégrale indéfinie ou une intégrale définie.	<code>int(x/(x^3-1),x);</code> <code>int(1/x^2,x=1..infinity);</code> <code>int(int(x^2*sin(y),y),x); # int. double</code>
<code>lhs</code>	Pour pointer vers le membre de gauche d'une équation ou d'un intervalle.	<code>lhs(y+1= 2*x);</code> <code>lhs( -3..5 );</code>
<code>limit</code>	Pour calculer une limite ou une limite directionnelle.	<code>limit(sin(x)/x, x=0);</code> <code>limit(1/x,x=0);</code> <code>limit(1/x,x=0,left);</code>
<code>ln</code>	Fonction logarithme naturel	<code>ln(x);</code> <code>ln(-1);</code>
<code>log</code>	Fonction logarithme naturel	<code>log(x);</code> <code>log(-1)</code>
<code>log[b]</code>	Fonction logarithme de base b	<code>log[2](5);</code>
<code>log10</code>	Fonction logarithme de base 10.	<code>log10(10000);</code>
<code>map</code>	Pour appliquer une procédure à chaque opérande d'une expression.	<code>Éq:=sqrt(1-x)=sqrt(7*x-11);</code> <code>map(x-&gt;x^2,Éq);</code>
<code>Matrix</code>	Pour créer des matrices compatibles avec les macro-commandes de l'extension <code>LinearAlgebra</code> .	<code>Matrix([[1,2,3],[4,5,6]]);</code> <code>&lt;&lt;1,4&gt; &lt;2,5&gt; &lt;3,6&gt;; # Écriture abrégée</code>
<code>max</code>	Pour obtenir le maximum d'une séquence de nombres réels.	<code>max(3/5,ln(2),9/13,-infinity);</code> <code>max();</code>
<code>min</code>	Pour obtenir le minimum d'une séquence de nombres réels.	<code>min(3/5,ln(2),9/13,-infinity);</code> <code>min();</code>
<code>normal</code>	Pour obtenir les numérateur et dénominateur relativement premier dans une expression rationnelle.	<code>E:=(x^2-y^2)/(x-y)^3;</code> <code>normal(E);</code>
<code>numer</code>	Pour extraire le numérateur d'une fraction ou le numérateur d'une somme de fractions.	<code>F:=1/x-2/x^2+3/(1+x);</code> <code>numer(F);</code>
<code>odeadvisor</code>	Pour déterminer la classe d'une équation différentielle. Cette macro-commande appartient à l'extension <code>DEtools</code> .	<code>Équation := diff(y(t),t)+sin(t)*y(t)=0;</code> <code>with(DEtools,odeadvisor):</code> <code>odeadvisor(Équation,y(t));</code>

tableau 5: Macro-commandes MAPLE

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>piecewise</code>	Pour créer une expression définie par morceaux.	<code>piecewise(x&lt;2,x^2,x&gt;=3,x,non_définie)</code>
<code>polar</code>	Pour saisir un nombre complexe sous la forme géométrique.	<code>polar(1,Pi); polar(-sqrt(2),5*Pi/6)</code>
<code>polarplot</code>	Pour créer un tracé en coordonnées polaires. Cette macro- commande appartient à l'extension <code>plots</code> .	<code>with(plots,polarplot); polarplot(2*cos(3*theta),theta=-Pi..Pi);</code>
<code>plot</code>	Pour créer un tracé (structure graphique) à deux dimensions .	<code>plot([sin(t),cos(t),t=-Pi..Pi]);</code>
<code>plot3d</code>	Pour créer un tracé (structure graphique) à trois dimensions.	<code>plot3d([x,y,exp(x^3*y)],x=-1..1,y=-1..1);</code>
<code>plot_real_curve</code>	Pour tracer dans le plan cartésien des courbes définies implicitement par des expressions polynomiales en $x$ et $y$ . Cette macro-commande appartient à l'extension <code>algcures</code> et est beaucoup plus efficace que <code>implicplot</code> de l'extension <code>plots</code> .	<code>with(algcurves,plot_real_curve); Ex:=x^2+2*x*y+y^2-75/7*x-55/7*y+122/7; plot_real_curve(Ex,x,y);</code>
<code>print</code>	Pour forcer un affichage dans la zone des résultats.	<code>print(`2+1`=2+1); interface(verbosespec=2); print(abs); print(surd);</code>
<code>quo</code>	Pour obtenir le quotient de la division polynomiale. Si un quatrième paramètre est précisé, celui-ci sera affecté du reste à la condition que ce paramètre soit une variable libre.	<code>Divid:=x^3+x+1; Divis:=x^2+x+1; quo(Divid, Divis, x, 'reste'); reste;</code>
<code>radnormal</code>	Pour obtenir la réduction des radicaux dans un nombre et, sur demande, rationaliser le dénominateur.	<code>N:=(2+sqrt(6))/(sqrt(3)+sqrt(2)); simplify(N); radnormal(N); radnormal(N,rationalized);</code>
<code>Re</code>	Pour obtenir la partie réelle d'un nombre complexe.	<code>Re(3-3*I); Re(polar(-5,Pi/4)); Re(exp(-7*i*Pi/6));</code>
<code>rem</code>	Pour obtenir le reste de la division polynomiale. Si un quatrième paramètre est précisé, celui-ci sera affecté du quotient à la condition que ce paramètre soit une variable libre.	<code>Divid:=x^3+x+1; Divis:=x^2+x+1; rem(Divid, Divis, x,'quotient'); reste;</code>
<code>restart</code>	Pour faire un lavage de cerveaux à MAPLE. L'environnement de la session de travail est remis dans un état semblable à celui qu'il avait au début de la session, sauf ce qui a trait à la mémoire qui a servi: elle n'est jamais restituée. Toutes les variables assignées redeviennent libres et toutes les macro-commandes des extensions ne sont plus reconnues.	<code>restart;</code>

tableau 5: Macro-commandes MAPLE

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>rhs</code>	Pour pointer vers le membre de droite d'une équation ou d'un intervalle.	<code>rhs(y+1= 2*x);</code> <code>rhs( -3..5 );</code>
<code>root</code>	Pour obtenir la racine principale n <sup>e</sup> d'un nombre complexe.	<code>root(81,3);</code> <code>root(-8,3);</code> <code>evalc(%);</code>
<code>RootOf</code>	Représente les racines d'une équation.	<code>RootOf(x^3-5*x^2+x-1);</code> <code>convert(%, radical,1);</code>
<code>sec</code>	Fonction trigonométrique sécante	<code>sec(Pi/24);</code> <code>convert(sec(Pi/24), radical);</code>
<code>seq</code>	Pour générer une séquence d'expressions.	<code>seq(10^(-k),k=1..8);</code> <code>L:= [seq(sin(Pi*i/12), i=0..6)];</code> <code>map(x-&gt;convert(x, radical),L);</code>
<code>simplify</code>	Pour simplifier une expression. La simplification n'a de sens que dans un contexte donné. Pour obtenir le résultat attendu, précisez les règles de simplifications. Entre autres: trig, radical, power, exp, ln. De plus, on peut passer, en paramètre, un conditionnement des indéterminées. Enfin, avec les règles ln, power et radical on dispose d'une option supplémentaire, symbolic, qui permet un traitement syntaxique de la simplification.	<code>E:=sqrt((x+1)^3);</code> <code>simplify(E);</code> <code>simplify(E, radical);</code> <code>simplify(E, assume=real);</code>
<code>sin</code>	Fonction trigonométrique sinus	<code>sin(Pi/5);</code> <code>convert(%, radical);</code>
<code>smartplot</code>	Pour créer un graphique de tracés 2D et 3D qui sont directement modifiables avec le bouton droit de la souris. Une fonctionnalité remarquable pour l'exploration graphique.	<code>smartplot(cos(x)+sin(x));</code>
<code>sort</code>	Pour trier un polynôme en ordre décroissant des exposants. Pour trier une liste de chaînes de caractères. Pour trier une liste de valeurs numériques décimales. Pour trier une liste de valeurs symboliques avec un critère de tri.	<code>sort([Francine, Anne, Jacinthe]);</code> <code>sort([-12,15,4]);</code> <code>sort([3,Pi,1,sqrt(2)]);</code> <code>Critere:=(x,y)-&gt;is(x&lt;y);</code> <code>sort([3,Pi,1,sqrt(2)],Critere);</code>
<code>solve</code>	Pour résoudre symboliquement une équation ou un système d'équations.	<code>Probleme:= y=m*x+b;</code> <code>solve(Probleme,x);</code> <code>solve(Probleme,{x});</code>
<code>spacecurve</code>	Pour tracer des courbes dans l'espace.	<code>spacecurve([sin(t),cos(t),t],</code> <code>    t=-2*Pi..2*Pi,color=navy);</code> <code>f:=(x,y)-&gt;-sqrt(x^2+y^2);</code> <code>spacecurve([x,-x,f(x,-x)],x=-3..3);</code>
<code>sqrt</code>	Pour obtenir la racine carrée principale.	<code>sqrt((x+1)^2);</code>
<code>sum</code>	Pour additionner symboliquement.	<code>sum(k,k=1..n);</code> <code>factor(%);</code> <code>sum(1/k!, k=0..infinity);</code> <code>P:=3*x^3-5*x^2+x-1;</code> <code>sum(k,k=RootOf(P));</code>

tableau 5: Macro-commandes MAPLE

## Principales macro-commandes MAPLE

Symboles	Description	Exemples
<code>subs</code>	Pour substituer syntaxiquement une sous-expression correspondant à un opérande d'une autre expression. S'il y a plusieurs substitutions demandées, elles seront faites dans l'ordre énoncé. Si ces substitutions sont formulées dans une liste, elles seront faites simultanément.	<code>subs(x=a^2+b^2, a*x^2);</code> <code>E:=x=1/2+1/2*sqrt(-7+4*y);</code> <code>subs(x=y,y=x,E);</code> <code>subs([x=y,y=x],E);</code>
<code>surd</code>	Pour obtenir la racine n <sup>e</sup> impaire réelle d'un nombre négatif.	<code>surd(81,3);</code> <code>surd(-8,3);</code>
<code>tan</code>	Fonction trigonométrique tangente	<code>tan(Pi/5);</code> <code>convert(tan(Pi/5),radical);</code> <code>convert(tan(Pi/7),radical);</code>
<code>testeq</code>	Pour vérifier une équation sur une base probabiliste. À utiliser lorsqu'aucune simplification ne permet de vérifier l'équation. L'exemple ci-contre est donc simpliste.	<code>Éq:=1+(tan^2)(x)=(sec^2)(x);</code> <code>evalb(Éq);</code> <code>testeq(Éq);</code>
<code>unapply</code>	Pour créer une fonction (procédure) à partir d'une expression.	<code>E:=x^2+3*x+1;</code> <code>f:=unapply(E,x);</code> <code>f(infinity);</code> <code>f(-infinity);</code>
<code>unassign</code>	Pour rendre libres une ou plusieurs variables simultanément.	<code>a:=sin(Pi/4);</code> <code>b:=-5;</code> <code>a+b;</code> <code>unassign('a,b');</code> <code>a+b;</code>
<code>Vector</code>	Pour créer des vecteurs colonnes compatibles avec les macro-commandes de l'extension <code>LinearAlgebra</code> .	<code>Vector([1,-2,3,-3]);</code> <code>&lt;1,-2,3,-3&gt;; # Écriture abrégée</code>
<code>Vector[row]</code>	Pour créer des vecteurs lignes compatibles avec les macro-commandes de l'extension <code>LinearAlgebra</code> .	<code>Vector[row]([1,-2,3,-3]);</code> <code>&lt;1 -2 3 -3&gt;; # Écriture abrégée</code>
<code>type</code>	Pour vérifier le type d'une expression.	<code>type(5/2,rational);</code>
<code>whattype</code>	Pour obtenir le type d'une expression.	<code>whattype(x-y);</code> <code>whattype(x/y);</code>
<code>with</code>	Pour mémoriser le chemin de certaines ou de toutes les macro-commandes d'une extension au cours d'une session Maple.	<code>with(plots,[display,implicitplot]);</code> <code>with(LinearAlgebra);</code>

tableau 5: Macro-commandes MAPLE