

Aide-mémoire MAPLE
Principales macro-commandes MAPLE
(Document 7 de 7)

Pierre Lantagne, enseignant retraité

Principales macro-commandes MAPLE

Symboles	Description	Exemples
abs	Fonction valeur absolue	<code>abs(x^2+5*x-3);</code>
add	Pour additionner numériquement une séquence de nombres.	<code>add(i^2,i=1..5);</code> <code>add(sqrt(i),i=1..10);</code>
alias	Pour définir des noms raccourcis.	<code>alias(C=binomial);</code> <code>C(5,8);</code> <code>alias(C=C); # Pour annuler l'alias</code>
algsubs	Pour substituer syntaxiquement chaque occurrence d'une expression A par une expression B dans une autre expression. Cette macro-commande généralise la macro-commande subs avec l'exception qu'elle ne permet pas de substitutions simultanées. (Voir subs)	<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>
allvalues	Pour imposer l'évaluation des racines d'une équation formulée en termes de RootOf. Dans le cas d'une équation polynomiale de degré ≤ 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>
arcsin, arccos, arctan, arccot, arcsec, arccsc	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>
assign	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>
assume	Pour conditionner une variable.	<code>assume(n, odd);</code> <code>cos(n*Pi);</code>
about	Pour obtenir les propriétés d'un objet ou d'une variable conditionnée.	<code>assume(y, nonneg);</code> <code>about(y);</code>
combine	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>
collect	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>
convert	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>
cos	Fonction trigonométrique cosinus	<code>cos(Pi/5);</code>
csc	Fonction trigonométrique cosécante	<code>csc(Pi/5);</code>

TABLEAU 1: Macro-commandes MAPLE

Principales macro-commandes MAPLE

Symboles	Description	Exemples
denom	Pour pointer vers le dénominateur d'une fraction ou le dénominateur commun d'une somme de fractions.	$F:=1/x-2/x^2+3/(1+x)$; denom(F);
diff (Voir ?D)	Pour obtenir la dérivée d'une expression.	diff(x ² *cos(x), x); diff(x*cos(x), x\$3); # dérivée successive d'ordre 3
display	Pour superposer différentes structures graphiques dans un même graphique. Cette macro-commande appartient à l'extension plots.	with(plots, display, polarplot); C1:=plot([1+cos(t), sin(t), t=0..2*Pi]); C2:=polarplot(2*cos(3*t), t=-Pi..Pi); display([C1, C2], scaling=constrained);
dsolve	Pour résoudre une équation différentielle ordinaire.	Équation: =y(t)*diff(y(t), t)+sin(t)=0; dsolve(Équation, y(t));
eval	Pour évaluer une formule exactement.	eval(1+x ³ , x=sqrt(5)); eval(x ² *y, [x=2, y=3]);
evalb	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.	evalb(1=1.0); evalb(sqrt(5)>2); evalb(evalf(sqrt(5))>2); evalb((sin ² (x)+(cos ² (x))=1);
evalc	Pour obtenir la forme algébrique d'un nombre complexe.	evalc(polar(-3, Pi/5)); evalc(exp(Pi/3*I)); evalc(ln(2+I));
evalf	Pour obtenir une approximation d'une valeur exacte (réelle ou complexe).	evalf(Pi/2); evalf(ln(2+I), 20);
exp	Fonction exponentielle de base e	exp(x ² +1);
expand	Pour développer une expression, en développant la distributivité de la multiplication sur l'addition ou en développant des formules trigonométriques et autres.	expand((x+1)/(x+2)); expand(cos(2*x)); expand(exp(5*x)); expand(ln(a/b)); assume(a, positive, b, positive); expand(ln(a/b));
factor	Pour factoriser toute expression pouvant être vue comme un polynôme ou une fraction rationnelle en ses indéterminées.	E:=2*cos(x)^2*exp(2*x)-exp(x)*\cos(x)^2+cos(x)*exp(3*x); factor(expand(E));
fsolve	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.	Eq:=cos(x)-2*x^3+5*x^2+x-5; fsolve(Eq, x=-1); fsolve(Eq, x=1); fsolve(Eq, x=2);
ifactor	Pour obtenir la factorisation complète d'un nombre entier ou d'un nombre rationnel.	ifactor(20!); ifactor(57/25546);
interface	Pour modifier les paramètres de l'interface usager.	interface(imaginaryunit=i); i^2;

TABLEAU 1: Macro-commandes MAPLE

Principales macro-commandes MAPLE

Symboles	Description	Exemples
Im	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>
implicitdiff	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>
implicitplot	Pour tracer le lieu géométrique définie implicitement par une équation. Cette macro-commande appartient à l'extension plots.	<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>
int	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>
lhs	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>
limit	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>
ln	Fonction logarithme naturel	<code>ln(x);</code> <code>ln(-1);</code>
log	Fonction logarithme naturel	<code>log(x);</code> <code>log(-1)</code>
log[b]	Fonction logarithme de base b	<code>log[2](5);</code>
log10	Fonction logarithme de base 10.	<code>log10(10000);</code>
map	Pour appliquer une procédure à chaque opérande d'une expression.	<code>Éq:=sqrt(1-x)=sqrt(7*x-11);</code> <code>map(x->x^2,Éq);</code>
Matrix	Pour créer des matrices compatibles avec les macro-commandes de l'extension LinearAlgebra.	<code>Matrix([[1,2,3],[4,5,6]]);</code> <code><<1,4> <2,5> <3,6>>; # Écriture abrégée</code>
max	Pour obtenir le maximum d'une séquence de nombres réels.	<code>max(3/5,ln(2),9/13,-infinity);</code> <code>max();</code>
min	Pour obtenir le minimum d'une séquence de nombres réels.	<code>min(3/5,ln(2),9/13,-infinity);</code> <code>min();</code>
normal	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>
numer	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>
odeadvisor	Pour déterminer la classe d'une équation différentielle. Cette macro-commande appartient à l'extension DEtools.	<code>Équation := diff(y(t),t)+sin(t)*y(t)=0;</code> <code>with(DEtools,odeadvisor):</code> <code>odeadvisor(Équation,y(t));</code>
piecewise	Pour créer une expression définie par morceaux.	<code>piecewise(x<2,x^2,x>=3,x,non_définie)</code>
polar	Pour saisir un nombre complexe sous la forme géométrique.	<code>polar(1,Pi);</code> <code>polar(-sqrt(2),5*Pi/6)</code>
polarplot	Pour créer un tracé en coordonnées polaires. Cette macro- commande appartient à l'extension plots.	<code>with(plots,polarplot);</code> <code>polarplot(2*cos(3*theta),theta=-Pi..Pi);</code>

TABLEAU 1: Macro-commandes MAPLE

Principales macro-commandes MAPLE

Symboles	Description	Exemples
plot	Pour créer un tracé (structure graphique) à deux dimensions .	<code>plot([sin(t),cos(t),t=-Pi..Pi]);</code>
plot3d	Pour créer un tracé (structure graphique) à trois dimensions.	<code>plot3d([x,y,exp(x^3*y)],x=-1..1,y=-1..1);</code>
plot_real_curve	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>algcurses</code> et est beaucoup plus efficace que <code>impliciplot</code> de l'extension <code>plots</code> .	<code>with(algcurses,plot_real_curve);</code> Ex: <code>=x^2+2*x*y+y^2-75/7*x-55/7*y+122/7;</code> <code>plot_real_curve(Ex,x,y);</code>
print	Pour forcer un affichage dans la zone des résultats.	<code>print(`2+1`=2+1);</code> <code>interface(verbosespec=2);</code> <code>print(abs);</code> <code>print(surd);</code>
quo	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;</code> <code>Divis:=x^2+x+1;</code> <code>quo(Divid, Divis, x, reste);</code> <code>reste;</code>
radnormal	Pour obtenir la réduction des radicaux dans un nombre et, sur demande, rationaliser le dénominateur.	<code>N:=(2+sqrt(6))/(sqrt(3)+sqrt(2));</code> <code>simplify(N);</code> <code>radnormal(N);</code> <code>radnormal(N,rationalized);</code>
Re	Pour obtenir la partie réelle d'un nombre complexe.	<code>Re(3-3*I);</code> <code>Re(polar(-5,Pi/4));</code> <code>Re(exp(-7*i*Pi/6));</code>
rem	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;</code> <code>Divis:=x^2+x+1;</code> <code>rem(Divid, Divis, x, quotient);</code> <code>reste;</code>
restart	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>
rhs	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>
root	Pour obtenir la racine principale n^e d'un nombre complexe.	<code>root(81,3);</code> <code>root(-8,3);</code> <code>evalc(%);</code>
RootOf	Représente les racines d'une équation.	<code>RootOf(x^3-5*x^2+x-1);</code> <code>convert(%,radical,1);</code>
sec	Fonction trigonométrique sécante	<code>sec(Pi/24);</code> <code>convert(sec(Pi/24),radical);</code>

TABLEAU 1: Macro-commandes MAPLE

Principales macro-commandes MAPLE

Symboles	Description	Exemples
seq	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->convert(x,radical),L);</code>
simplify	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>
sin	Fonction trigonométrique sinus	<code>sin(Pi/5);</code> <code>convert(%,radical);</code>
smartplot	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>
sort	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)->is(x<y);</code> <code>sort([3,Pi,1,sqrt(2)],Critere);</code>
solve	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>
spacecurve	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)->-sqrt(x^2+y^2);</code> <code>spacecurve([x,-x,f(x,-x)],x=-3..3);</code>
sqrt	Pour obtenir la racine carrée principale.	<code>sqrt((x+1)^2);</code>
sum	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>
subs	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>
surd	Pour obtenir la racine n ^e impaire réelle d'un nombre négatif.	<code>surd(81,3);</code> <code>surd(-8,3);</code>
tan	Fonction trigonométrique tangente	<code>tan(Pi/5);</code> <code>convert(tan(Pi/5),radical);</code> <code>convert(tan(Pi/7),radical);</code>

TABLEAU 1: Macro-commandes MAPLE

Principales macro-commandes MAPLE

Symboles	Description	Exemples
testeq	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.	Éq:=1+(tan^2)(x)=(sec^2)(x); evalb(Éq); testeq(Éq);
unapply	Pour créer une fonction (procédure) à partir d'une expression.	E:=x^2+3*x+1; f:=unapply(E,x); f(infinity); f(-infinity);
unassign	Pour rendre libres une ou plusieurs variables simultanément.	a:=sin(Pi/4); b:=-5; a+b; unassign(a,b); a+b:
Vector	Pour créer des vecteurs colonnes compatibles avec les macro-commandes de l'extension LinearAlgebra.	Vector([1,-2,3,-3]); <1,-2,3,-3>; # Écriture abrégée
Vector[row]	Pour créer des vecteurs lignes compatibles avec les macro-commandes de l'extension LinearAlgebra.	Vector[row]([1,-2,3,-3]); <1 -2 3 -3>; # Écriture abrégée
type	Pour vérifier le type d'une expression.	type(5/2,rational);
whattype	Pour obtenir le type d'une expression.	whattype(x-y); whattype(x/y);
with	Pour mémoriser le chemin de certaines ou de toutes les macro-commandes d'une extension au cours d'une session Maple.	with(plots,[display,implicitplot]); with(LinearAlgebra);

TABLEAU 1: Macro-commandes MAPLE