1 Maple Quick Reference

1.1 On-Line Help

Command	Description	Example
help (also ?)	display on-line help	?DEplot
usage (also ??)	display USAGE section of an on-line help document	??dsolve
example (also ???)	display EXAMPLES section of an on-line help document	???plot

1.2 Mathematical Operations, Functions, and Constants

Symbol	Description	Example
+, -, *, /, ^	add, subtract, multiply, divide, power	x^(-4)*3+x/Pi;
sin, cos, tan, cot, sec, csc	trigonometric functions	sin(x-Pi/5)-sec(x^2);
arcsin, arccos, arctan, arccot, arcsec, arccsc	inverse trigonometric functions	arctan(2*theta);
exp	exponential function	exp(2*x);
ln	natural logarithm	ln(x*y/2);
log10	common logarithm (base 10)	log10(1000);
abs	absolute value	abs((-3)^5);
sqrt	square root	sqrt(24);
!	factorial	k!;
=, <>, <, <=, >, >=	equations and inequalities	a*x^2 + b*x + c = 0; E^Pi > Pi^E;
Pi, E, I	π, e, i (mathematical constants)	exp(Pi*I);
infinity	infinity (∞)	int(1/x^2, x=1..infinity);

1.3 Maple Symbols and Abbreviations

Symbol	Description	Example
:=	assignment	f := x^2/y^3;
;	terminate command; display result	int(x^2, x);
:	terminate command; hide result	int(x^2, x):
..	specify a range or interval	plot(t*exp(-2*t), t=0..3);
{ }	set delimiter	{ y, x, y };
[ ]	list delimiter	[y, x, y];
", "", """ (also ?ditto)	references prior result (double quote)	Int(exp(x^2), x=0..1): " = value(");
` ` (also ?strings)	string delimiter (back quote) Note: single quote (´) is different	plot(sin, title=`y=sin(x)`);
´ ´ (also ?uneval)	delayed evaluation (single quote) Note: back quote (`) is different	f := x->piecewise(x<0,1-x,1); plot('f(x)', x=-1..1);
->	mapping (procedure) definition	f := (x,y) -> x^2*sin(x-y); f(Pi/2,0);

1.4 General Display, Evaluation, and Manipulation

Command	Description	Example
about	check assumptions on objects	about(t);
assign	perform assignments(often used after solve or dsolve)	s:=solve({x+y=1, 2*x+y=3}, {x,y}); assign(s);
assume	impose additional properties on objects	assume(t>0);
convert	convert an expression to a different form	convert(x^3/(x^2-1), parfrac, x);
denom	extract the denominator of an expression	denom((x+1)^3/(x+2)^2);
display	display a list of plot structures (in the plots package)	with(plots): F:=plot(exp(x), x=0..1, style=line); G:=plot(ln(x), x=0..1, style=point); display({F,G}, title=‘2 curves‘);
evalb	evaluate a Boolean expression (returns true or false)	evalb(exp(Pi) > Pi^E);
evalc	evaluate a complex-valued expression(returns a value in the form a+I*b)	evalc(exp(alpha+I*omega));
evalf	evaluate using floating-point arithmetic	evalf(exp(Pi^2));
lhs	left-hand side of an equation	lhs(y = a*x^2 + b);
numer	extract the numerator of an expression	numer((x+1)^3/(x+2)^2);
plot	create a 2-dimensional plot of functions	plot(u^3, u=0..1, title=`cubic`); plot({sin(x), cos(x)}, x=0..Pi);
plot3d	create a 3-dimensional plot of functions	plot3d(sin(x+y), x=0..2, y=0..1);
restart	clear all definitions	restart:
rhs	right-hand side of an equation	rhs(y = a*x^2 + b);
seq	create a sequence	{ seq([0,i], i=-3..3) };
subs	substitute values into an expression	subs(x=r^(1/3), 3*x*ln(x^3));
value	evaluate an inert expression (typically used with Limit, Diff, or Int)	G := Int(exp(-x^2), x); value(G);
with	load a package	with(DEtools); with(plots):

1.5 Algebra, Calculus and Differential Equations

Command	Description	Example
`collect`	collect coefficients of like powers	`collect( (x+1)^3*(x+2)^2, x );`

`D`	differential operator(often used when specifying derivative initial conditions for `dsolve`)	`ODE := diff(y(x),x$2) +y(x) = 1;`
		`IC := y(0)=1, D(y)(0)=1;`
		`dsolve( { ODE, IC }, y(x) );`

`DEplot`	create a plot associated with an ODE or system of ODEs; see `?DEplot` for more information	`ODE := diff( y(x),x ) = 2*x*y;`
		`DEplot( ODE, [x,y], x=-2..2,`
		`y=-1..1, arrows=THIN );`

`diff`	compute the derivative of an expression	`diff( a*x*exp(b*x)*cos(c*y), x )`
`Diff`	inert (unevaluated) form of `diff`	`Diff( exp(x^2), x );`
`dsolve`	solve ordinary differential equations; see `?dsolve` for a list of available options	`ODE := diff(y(x),x)-y(x)=1;` `dsolve( ODE, y(x) );`
`factor`	factor a polynomial	`factor( (x^3-y^3)/(x^4-y^4) );`
`fsolve`	solve using floating-point arithmetic	`fsolve( 1/10*t+t*exp(-2*t)=1, t);`
`int`	definite or indefinite integration	`int( sin(x), x=0..Pi );`
`Int`	inert (unevaluated) form of `int`	`Int( sqrt(x), x=0..1 );`
`limit`	calculate a limit	`limit( sin(x)/x, x=0 );`
`Limit`	inert (unevaluated) form of `limit`	`Limit( sin(x)/x, x=0 );`

`odeplot`	plot the solution of an initial value problem (in the `plots` package)	`ode:=diff(y(x),x) = sin(x*y(x));`
		`ic:=y(0)=2;`
		`dsolve({ode,ic},y(x),type=numeric);`
		`odeplot(",[x,y(x)],0..6;`

`simplify`	apply simplification rules to an expression	`simplify( exp( a+ln(b*exp(c)) ) );`
`solve`	solve equations	`solve( x^4 - 5*x^2 + 6*x = 2, x );`
`piecewise`	piecewise-defined functions	`piecewise(x<Pi,cos(2*x),1);`

Miscellaneous Notes:

- The document is also available on the World Wide Web; the Universal Resource Locator is `http://www.math.sc.edu/~meade/maple/maple-ref/maple-ref.html`
- Comments, corrections, and suggestions for improvements may be sent to the author by e-mail (`meade@math.sc.edu`).
-