

Maple: A Quick Reference

Prepared by:

Douglas Meade

Department of Mathematics

University of South Carolina

January 1995 (February 1997, updated for Release 4)

Symbols and Abbreviations

| Symbol | Description | Example |
|-----------------------------------|---|--|
| <code>:=</code> | assignment | <code>f := x^2/y^3;</code> |
| <code>;</code> | terminate command; display result | <code>int(x^2, x);</code> |
| <code>:</code> | terminate command; hide result | <code>int(x^2, x):</code> |
| <code>..</code> | specify a range or interval | <code>plot(t*exp(-2*t), t=0..3);</code> |
| <code>{ }</code> | set delimiter | <code>{ y, x, y };</code> |
| <code>[]</code> | list delimiter | <code>[y, x, y];</code> |
| <code>"</code> | refers to previous result (double quote) | <code>Int(exp(x^2), x=0..1):</code> <code>" = value(");</code> |
| <code>`</code> (also ?strings) | string delimiter (back quote) NOTE: single quote (<code>'</code>) is different | <code>TITLE := `A Pretty Plot`;</code> |
| <code>'</code> (also ?uneval) | delayed evaluation (single quote) NOTE: back quote (<code>`</code>) is different | <code>x := 'x';</code> <i>not needed in normal usage</i> |
| <code>-></code> | mapping (procedure) definition | <code>f := (x,y) -> x^2*sin(x-y);</code> <code>f(Pi/2,0);</code> |

Mathematical Operations, Functions, and Constants

| Symbol | Description | Example |
|---|---|---|
| <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>^</code> | add, subtract, multiply, divide, power | <code>x^(-4)*3+x/Pi;</code> |
| <code>sin</code> , <code>cos</code> , <code>tan</code> , <code>cot</code> , <code>sec</code> , <code>csc</code> | trigonometric functions | <code>sin(x-Pi/5)-sec(x^2);</code> |
| <code>arcsin</code> , <code>arccos</code> , <code>arctan</code> , <code>arccot</code> , <code>arcsec</code> , <code>arccsc</code> | inverse trigonometric functions | <code>arctan(2*theta);</code> |
| <code>exp</code> | exponential function | <code>exp(2*x);</code> |
| <code>ln</code> | natural logarithm | <code>ln(x*y/2);</code> |
| <code>log10</code> | common logarithm (base 10) | <code>log10(1000);</code> |
| <code>abs</code> | absolute value | <code>abs((-3)^5);</code> |
| <code>sqrt</code> | square root | <code>sqrt(24);</code> |
| <code>!</code> | factorial | <code>k!;</code> |
| <code>=</code> , <code><></code> , <code><</code> , <code><=</code> , <code>></code> , <code>>=</code> | equations and inequalities <i>Note: E no longer exists; use exp(1)</i> | <code>diff(y(x), x) + x*y(x) = F(x);</code> <code>exp(Pi) > Pi^exp(1);</code> |
| <code>Pi</code> , <code>I</code> | π , i (mathematical constants) | <code>exp(Pi*I);</code> |
| <code>infinity</code> | infinity | <code>int(x^(-2), x=1..infinity);</code> |

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/>
- Please send comments, corrections, and suggestions for improvements to meade@math.sc.edu.

Commands

| Command | Description | Example |
|---------------|--|---|
| restart | clear all Maple definitions | restart; |
| with | load a Maple package | with(DEtools); with(plots); |
| help (also ?) | display Maple on-line help | ?DEplot |
| example | provide examples for a function | example(plot); |
| limit | calculate a limit | limit(sin(x)/x, x=0); |
| diff | compute the derivative of an expression | diff(a*x*exp(b*x)*cos(c*y), x) |
| int | definite or indefinite integration | int(sin(x), x=0..Pi); |
| value | evaluate an inert expression (typically used with Limit, Diff, or Int) | G := Int(exp(-x^2), x); value(G); |
| Limit | inert (unevaluated) form of limit | Limit(sin(x)/x, x=0); |
| Diff | inert (unevaluated) form of diff | Diff(exp(x^2), x); |
| Int | inert (unevaluated) form of int | Int(sqrt(x), x=0..1); |
| 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); |
| 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(x); DEplot(ODE, [y(x)], x=-2..2, y=-1..1, arrows=SMALL); |
| display | display a list of plot structures | 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'); |
| solve | solve equations | solve(x^4 - 5*x^2 + 6*x = 2, x); |
| fsolve | solve using floating-point arithmetic | fsolve(1/10*t+t*exp(-2*t)=1, t); |
| subs | substitute values into an expression | subs(x=r^(1/3), 3*x*ln(x^3)); |
| simplify | apply simplification rules to an expression | simplify(exp(a+ln(b*exp(c)))); |
| factor | factor a polynomial | factor((x^3-y^3)/(x^4-y^4)); |
| convert | convert an expression to a different form | convert(x^3/(x^2-1), parfrac, x); |
| collect | collect coefficients of like powers | collect((x+1)^3*(x+2)^2, x); |
| rhs | right-hand side of an equation | rhs(y = a*x^2 + b); |
| 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); |
| denom | extract the denominator of an expression | denom((x+1)^3/(x+2)^2); |
| evalf | evaluate using floating-point arithmetic | evalf(exp(Pi^2)); |
| evalc | evaluate a complex-valued expression (returns a value in the form a+I*b) | evalc(exp(alpha+I*omega)); |
| evalb | evaluate a Boolean expression (returns true or false) | evalb(exp(Pi) > Pi^E); |
| assign | perform assignments (often used after solve or dsolve) | s:=solve({x+y=1, 2*x+y=3}, {x,y}); assign(s); |
| seq | create a sequence | { seq([0,i], i=-3..3) }; |
| dsolve | solve ordinary differential equations; see ?dsolve for a list of available options | dsolve(diff(y(x),x)-y(x)=1, y(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)); |
| assume | inform Maple of additional properties of objects | assume(t>0); |
| about | check assumptions on Maple objects | about(t); |