Some Basic Maple for Math 163

Command Line Structure

Command lines in Maple must terminate with either a colon (`:`) or a semi-colon (`;`), and be followed by hitting the Enter key.

`?<Enter>`	provide some general help (this is one of the best features of Maple)
`?<command><Enter>`	provide help specifically about `<command>`
`<command>;<Enter>`	perform `<command>` and print the results to the screen
`<command>:<Enter>`	perform `<command>` but do not print the results to the screen
`#<junk><Enter>`	`<junk>` is a comment and will be ignored by Maple

Operators

`+`	means addition
`-`	means subtraction
`*`	means multiplication
`/`	means division
`^` and `**`	mean exponentiation
`@`	means function composition

Special Functions

`abs(x)` means $|x|$, and `sqrt(x)` means $\sqrt{x}$

The Trigonometric Functions

$\sin(x), \cos(x), \tan(x), \sec(x), \csc(x),$ and $\cot(x)$ follow the usual notation; the angle x is assumed to be measured in radians.

Constants

`Pi` means π, and `infinity` means ∞

Commands

`> %;`	the last computed value
`> %%;`	the second last computed value
`> %%%;`	the third last computed value
`> x := <value>;`	assigns `<value>` to x
`> x := <expression>;`	assigns `<expression>` to x
`> x := ´x´;`	reidentifies x as a variable
`> f := x -> <expression in x>;`	identifies the function $f(x)$ as `<expression in x>`
`> Digits := n;`	sets the number of digits in floating point notation to n
`> evalf(f);`	converts f to floating point form
`> expand(f);`	expands f
`> factor(f);`	factors f
`> fsolve(f = a, x);`	*numerically* solves the equation $f = a$ for x
`> quit;`	quits Maple
`> simplify(f);`	simplifies the expression f
`> solve(f = a, x);`	solves the equation $f = a$ for x *exactly*
`> subs(x = a, f);`	substitutes a for x in f
`> restart;`	clear all previously typed commands from memory
`> lhs(<equation>);`	the left-hand side of `<equation>`
`> rhs(<equation>);`	the right-hand side of `<equation>`
`> numer(<fraction>);`	the numerator of `<fraction>`
`> denom(<fraction>);`	the denominator of `<fraction>`
`> plot(f(x),x=a..b);`	plot the function f from $x = a$ to $x = b$
`> plot(f(x),x=a..b,y=c..d);`	plot the function f from $x = a$ to $x = b$, limiting output to points with y-coordinates between c and d
`> plot({f(x),g(x)},x=a..b,y=c..d);`	plot functions f and g from $x = a$ to $x = b$, restricting output to points with y-coordinates between c and d