Command Summary

The following is a list and brief description of the Maple commands used in the Maple labs. More detailed information on these commands (and others) can be obtained by typing ? followed by the name of the command while in the Maple program.

0. Introduction

`+,-,*,/,^` add, subtract, multiply, divide, exponentiate.
`sin(x), cos(x), tan(x), csc(x), sec(x), cot(x)` - trigonometric functions.
`sqrt(x);, root(x,n);` - the square root and the nth root of x.
`evalf(x);` - evaluate or convert `x` to a floating point number.
`?name` - gives Maple help on the command "name". (No semicolon required)

1. Algebra

`factor(p(x));` - factors the expression p(x).
`expand(f(x));` - expands f(x) by multiplying out or by applying expansion identities.
`simplify(f(x));` - algebraically simplifies f(x).
`solve(eq=0,x);` - solves the equation eq=0 in x for x.
`:=` - assigns the name on the left hand side to whatever the right hand side evaluates to.
example: `c:=5;` assigns the value `5` to the name `c`

2. Functions

`->` The "arrow" is used to define function rules.
examples:
`x->x^2;` is the squaring rule.
`(x->x^2)(3);` applies the squaring rule to `3`.
`f:=x->x^2;` assigns the name `f` to the squaring rule so `f(3);` applies the rule to `3`.
`piecewise(cond1, expr1, cond2, expr2);` - defines a piecewise function which is equal to `expr1` when `cond1` is true or is equal to `expr2` when `cond2` is true.
`plot(f(x),x=a..b);` - creates a plot of y=f(x) over the domain [a,b].
`plot([f(x),g(x)],x=a..b);` - creates a plot of both y=f(x) and y=g(x).

3. Limits

`limit(f(x),x=a);` - computes the limit of `f(x)` as `x` approaches to `a`.
`limit(f(x),x=a,left);` - computes the limit of `f(x)` as `x` approaches to `a` from the left.
`limit(f(x),x=a,right);` - computes the limit of `f(x)` as `x` approaches to `a` from the right.
`SHOWLIMIT(f(x),x=a);` - creates an animated plot showing the limit process.
`SHOWSECANTS(f(x),x=a);` - creates an animated plot showing how the slopes of the secant lines approach a limit.

4. Differentiation

`diff(f(x),x);` - computes the derivative of `f(x)` with respect to `x`.
`D(f);` - computes the derivative function (rule).
`D(f)(a);` - computes the derivative function and evaluates it at `a`.
`D(f)(x);` - computes the derivative function and evaluates it at `x` . (same as `diff(f(x),x);`)

`SHOWTANGENTS(f(x),x=a..b);` - creates an animated plot showing tangent lines on the curve.

5. Curve Plotting

`plot(f(x),x=a..b,options);` - creates a plot of the `f(x)` over the interval `[a,b]` . The following is a list of some plot options.
`y=c..d` - causes the vertical range to be from c to d.
`colour=blue` - causes the curve to be coloured blue. (see lab for all available colours)
`scaling=constrained` - causes the x and y scales to be the same.
`discont=true` - causes Maple to not plot vertical lines at discontinuities.
`style=point` - causes Maple to plot using points instead of line segments.
`symbol=cross` - puts a cross at the point. (also: point, diamond, circle,..)
`numpoints=200` - causes Maple to use at least 200 points to plot. (Default is 50)

`plot([[a,b],[c,d]],style=point);` - creates a plot of the points `(a,b)` and `(c,d)`.
`plot([[a,b],[c,d]]);` - creates a plot of the line segment connecting `(a,b)` to `(c,d)`.

6. Integration

`int(f(x),x);` - indefinite integral of `f(x)` with respect to x.
`int(f(x),x=a..b);` - definite integral of `f(x)` from `x = a` to `x = b`.
`Int(f(x),x);` - displays without trying to evaluate the integral.
`value(expr);` - causes Maple to evaluate `expr`.

Note: The commands which have all capital names are not standard Maple commands. These commands are names of programs (called procedures) written for these labs. These commands are only available in the labs in which they appear.