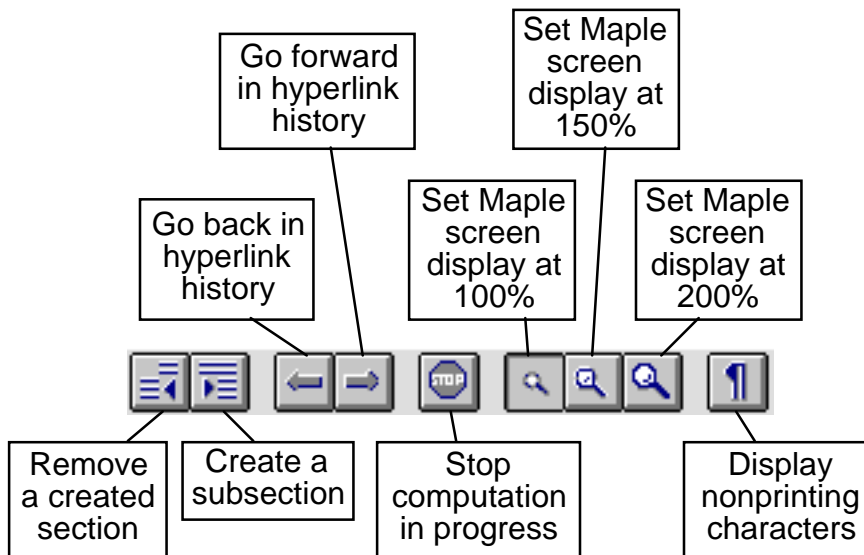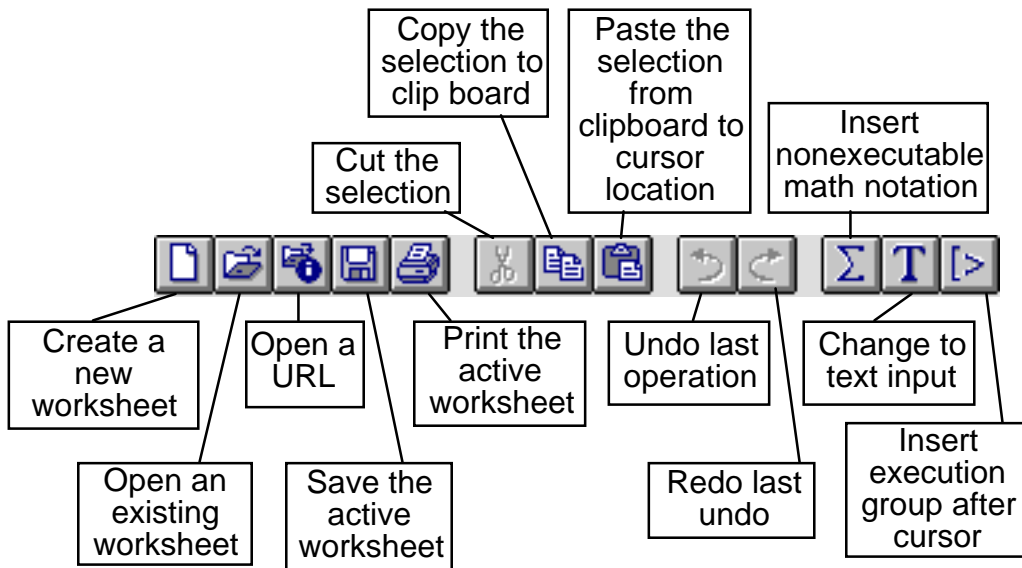# BASIC MAPLE COMMANDS AND MENUS

This sheet is provided as a quick reference for some of the *Maple* commands and menus. For more detailed instructions, see the appropriate section in this workbook or "Using Help" under **Help** on the *Maple* menu bar.
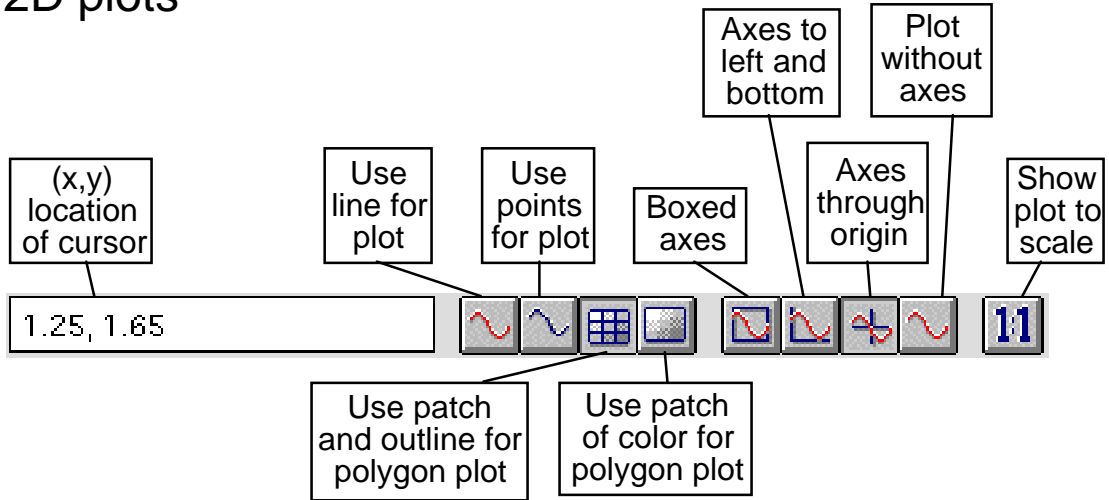
MAIN MENU BAR

Copy the selection to clip board

Paste the selection from clipboard to cursor location

Insert nonexecutable math notation

Cut the selection

Create a new worksheet

Open a URL

Print the active worksheet

Undo last operation

Change to text input

Open an existing worksheet

Save the active worksheet

Redo last undo

Insert execution group after cursor

Go forward in hyperlink history

Set Maple screen display at 150%

Go back in hyperlink history

Set Maple screen display at 100%

Set Maple screen display at 200%

Remove a created section

Create a subsection

Stop computation in progress
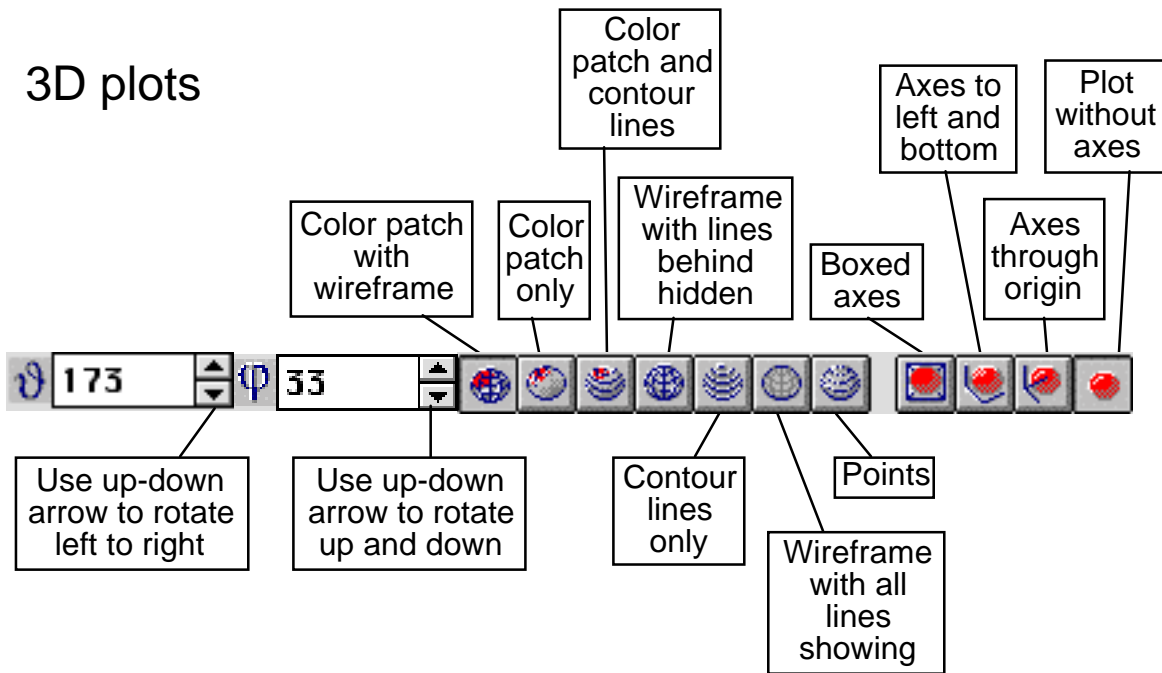
Display nonprinting characters

These commands can also be accessed under the menu headings **File**, **Edit**, **View**, and **Insert**. There are additional style commands under **Format** including justification for the worksheet.

2D plots

Axes to left and bottom

Plot without axes

(x,y) location of cursor

Use line for plot

Use points for plot

Boxed axes

Axes through origin

Show plot to scale

`1.25, 1.65`

Use patch and outline for polygon plot

Use patch of color for polygon plot

3D plots

Color patch and contour lines

Axes to left and bottom

Plot without axes

Color patch with wireframe

Color patch only

Wireframe with lines behind hidden

Boxed axes

Axes through origin

`173` `33`

Use up-down arrow to rotate left to right

Use up-down arrow to rotate up and down

Contour lines only

Points

Wireframe with all lines showing

These commands can also be accessed under the menu headings **Axes**, **Color**, **Style**, and **Projection**. There are additional style commands under these menu headings for various views and coloring schemes.

COMMANDS FOR 2D PLOTS

Common Name	Maple Command and Options

Cartesian Plot
(See pages 3, 15, 21, 22, and 43.)
	`plot([f,g],x=a..b,c..d,discont=true);`
	  `[f,g]` — function(s) to be plotted (omit [ ] to plot one function)
	  `x=a..b` — horizontal domain
	  `c..d` — (optional) vertical range
	  `discont=true` — (optional) remove lines connecting plus and
	    minus infinity

Contour Plot
(See page 53.)
	execute `with(plots);`
	`contourplot(f,x=a..b,y=c..d);`
	  `f` — function to be plotted
	  `x=a..b` — horizontal domain
	  `y=c..d` — vertical range

Density Plot
(See page 53.)
	execute `with(plots);`
	`densityplot({f},x=a..b,y=c..d);`
	  `f` — function to be plotted
	  `x=a..b` — horizontal domain
	  `y=c..d` — vertical range

Direction Field
(See page 33.)
	execute `with(plots);`
	`fieldplot([dx,dy],x=a..b,y=c..d);`
	  `dx` — change in the x value
	  `dy` — change in the y value
	  `x=a..b` — horizontal domain
	  `y=c..d` — vertical range

Histogram
(See page 30.)
	execute `with(stats);` and `with(statplots);`
	`histogram(L,area=a,numbars=b);`
	  `L` — list of data to be plotted
	  `area=a` — total area of all bars
	  `numbars=b` — number of bars to be plotted

Implicit Plot
(See pages 17 and 32.)
	execute `with(plots);`
	`implicitplot({f,g},x=a..b,y=c..d);`
	  `{f,g}` — equation(s) to be plotted (omit {} to plot one function)
	  `x=a..b` — horizontal domain
	  `y=c..d` — vertical range

Parametric Plot
(See pages 42 and 49.)
	`plot([x,y,t=a..b]);`
	  `x` — x-coordinate as a function of t
	  `y` — y-coordinate as a function of t
	  `t=a..b` — domain for t

Polar Plot
(See page 45.)
	execute `with(plots);`
	`polarplot({f,g},q=a..b);`
	  `{f,g}` — function(s) to be plotted (omit {} to plot one function)
	  `q=a..b` — angle domain to be plotted

COMMANDS FOR 3D PLOTS

Common Name	Maple Command and Options
Cartesian Plot (See pages 53 and 57.)	`plot3d({f},x=a..b,y=c..d,orientation=[p,t]);` `{f}` — function of (x,y) to be plotted `x=a..b` — horizontal domain `y=c..d` — vertical domain `orientation=[p,t]` — (optional) angles of view for graph
Cylindrical Plot (See page 59.)	execute `with(plots);` `cylinderplot([r,t,f,g],t=a..b,r=c..d);` `r,t` — letters used for radius and angle `f,g` — functions used for radius, r, and angle, t `t=a..b` — domain for angle `r=c..d` — domain for radius
Direction Field (See page 63.)	execute `with(plots);` `fieldplot3d(F,x=a..b,y=c..d,z=e..f);` `F` — function of (x, y, z) to be plotted `x=a..b` — domain to plot in x direction `y=c..d` — domain to plot in y direction `z=e..f` — range to plot in z direction
Solid of Revolution (See page 27.)	execute `with(plots);` `tubeplot([x,0,0],x=a..b,radius=f);` `[x,0,0]` — axis to revolve about `x=a..b` — domain for t `radius=f` — function to be revolved
Spherical Plot (See page 60.)	execute `with(plots);` `sphereplot(f,t=a..b,p=c..d);` `f` — function of theta, t, and phi, p, to be plotted `t=a..b` — domain for theta `p=c..d` — domain for phi
Vectors (See page 51.)	execute `with(plots);` `spacecurve([v],t=a..b,axes=boxed,numpoints=n);` `[v]` — vector in (x, y, z) coordinates as functions of t `t=a..b` — domain for t `axes=boxed` — (optional) label axes on outside edges `numpoints=n` — number of points to compute for graph

Maple has more than one command for some operations. This summary gives guidelines for the multiple commands.

Category	Commands	Guidelines
Declare equation	`f:=x->`	The created equation is a function of x. Substitution is done by entering f(a). This can be cumbersome when plotting or doing other operations with a function. (See pages 3, 4, and 5.)
	`y:=`	Names an equation, plot, or anything. The equation may include an equal sign. Using the named expression in plots or other equations is done by just using the given letter. Substitution must be done using the command `subs(x=a,y);`. (See pages 17 and 18.)
Derivative	`D(f)(x)`	*Maple* uses this command if f is declared f(x). This must be used to declare a new function as a derivative of a given function. (See pages 19 and 46.)
	`diff(y,x);` `diff(f(x),x);`	Takes the derivative of any expression. The expression may be a named equation or an f(x). (See page 22.)
	`Diff(y,x);`	Displays, but does not compute the derivative.
Integral	`int(y,x);` `int(f(x),x);`	Gives the most general antiderivative of any expression. The expression may be a named equation or an f(x). (See page 23.)
	`int(y,x=a..b);`	Computes the definite integral on the range x=a to x=b. (See page 26.)
	`Int(y,x);` `Int(y,x=a..b);`	Displays, but does not compute the integral.
Solve equation	`solve(f=g,x);`	*Maple* gives the exact roots for the equality. The answer may be in rational numbers, radical form, complex numbers, or as "ROOTS" of a given equation in Z. (See page 5.)
	`fsolve(f=g,x);`	*Maple* gives a decimal approximation for the real roots of the equality. If no real roots exist, no answer is given. (See page 5.)
	`fsolve(f=g,x=a);`	*Maple* gives a decimal approximation for the real root nearest x=a. (See page 6.)
	`fsolve(f=g,x=a..b);`	*Maple* gives a decimal approximation for the real root between x=a and x=b. (See page 50.)

QUICK REFERENCE OF COMMON COMMANDS

To Do	Command	Comments
Arithmetic	**+**	Add
	–	Subtract
	*	Multiply
	/	Divide
	sqrt(a);	Take square root of a.
	^	Raise to a power
	abs(a);	Take absolute value of a.
Clear *Maple*'s memory	**restart;**	This does not delete the worksheet. Only *Maple*'s memory is cleared. (See page 39.)
Decimal form of number	**evalf(n);**	Changes the given number, n, to a decimal. (See pages 1 and 26.)
Decimal places	**Digits:=n;**	Sets the *Maple* display to n decimal places. (See page 39.)
Derivative	**diff(y,x);**	Computes derivative of y with respect to x. (See page x.)
Graph	**plot(y,x=a..b);**	Graphs the equation, y, on the domain x=a to x=b. (See page viii.)
Integrate (indefinite)	**int(y,x);**	Integrates y with respect to x. (See page x.)
Integrate (definite)	**int(y,x=a..b);**	Computes integral of y from x=a to x=b. (See page x.)
Substitute	**f(a);** **subs(x=a,f);**	Substitutes the value a for x. Use **f(a);** if the expression was declared using **f:=x->**. Use **subs(x=a,f);** if the expression was declared using **f:=**.
Un-name	**unassign('f');**	Removes the expression assigned to f from *Maple*'s memory. (See page 2.)