

## 19 What's New in 3.6?

New features have dribbled into *LiveMath* over the past decade(s). With this new *LiveMath* 3.6, we here highlight some of the new features to look out for in this minor update version.



---

### 19.1 *LiveMath* Plug-In is Retired

With some regrets, *LiveMath* Plug-In has been retired.

The overall problem with plug-ins is the new (and it is about time!) focus on browser security. Plug-ins, by their very nature, are programs in their own right that could be doing some very nasty things to your computer once installed, as can happen when installing any program.

The browsers have really clamped down on security, as you can see from the near weekly updates of Flash and Java, and their frequent disabling by Apple and Microsoft.

The “Netscape Plug-in API” has never been updated since 1998. It is horrible. The *LiveMath* Plug-In was continually in a “let’s try this example over here” to try to get it to work, and it never really did very well. With every new browser release of Firefox, Chrome, and Safari, the *LiveMath* Plug-In would break. We just couldn’t keep up with the maintenance required.

On the Windows side there are 2 plugins, of course: the Active-X control for Internet Explorer, now Edge on Windows 10, and the Netscape-based plugin for all of the other browsers (that support plugins).

On the Mac side, the browsers all made their move to 64-bit-based binaries, and *LiveMath* is still built upon the older 32-bit technologies (QuickDraw). So that just in and of itself sunk the Mac side.

The commerical side of *LiveMath* Plug-In never materialized, either. We made a lot of “textbook websites” for various publishers over the initial years of *LiveMath* Plug-In, but that development did not continue. We had “Viewer-type” restrictions in *LiveMath* Plug-In which allowed you to interact with preset equations and computations and graphs to achieve the “LiveExample” goal. Yet, aside from us and a few academics, few used the *LiveMath* Plug-In.

So, for a web presence for *LiveMath*, we will return to links from webpages into *LiveMath* Maker. You can “teach” the web browser to launch *LiveMath* when you try to open a “.the” file - and you may reteach the browsers to do the same for the formerly Plug-In files with “.thp” file extensions.

---

## 19.2 Larger Palette



### Screencast Video

<http://www.LiveMath.com/v/1902>

Way back in the year 2000, the *LiveMath* palette seemed enormous with a width of 500 pixels. “It’s too big” many screamed! The deluxe screensize of the day was a whopping 640x480 pixels.

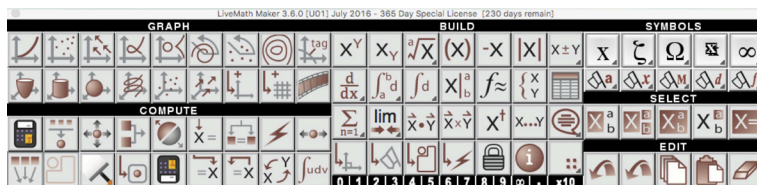
Fast-forward to 2017 and screens are commonly 1500-2000 pixels wide and beyond; the smaller icons just don’t fit the bill.

Some like the smaller button size, as they may only use the palette once in a while. But others, especially on Windows 10 and its crazy way it deals with screen resolution (dpi), a larger palette option is required.

While a stretchable palette is not possible given the older technology on both the Mac and Windows sides, we did a simple double enlargement to the new Large Palette options, with the width of the palette now at 1000 pixels:



*Small Full Palette: 500 pixels wide*





*Large Full Palette: 1000 pixels wide*

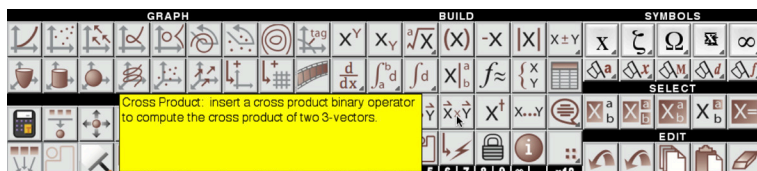
The Large palettes also use a new version of the artwork: vector-based artwork that was commissioned for *LiveMath* 4.0, but I stole it over for this intermediate *LiveMath* 3.6 release, mainly because the older artwork was not in vector format, so it could not be enlarged.

Both the Full and Skinny palette options remain, so now there are 4 palette options:

- (Small) Full Palette
- (Small) Skinny Palette
- Large Full Palette
- Large Skinny Palette

The small palette is (primarily) made of buttons that are 24x24 pixels. The new larger palette is (primarily) made of buttons that are 48x48 pixels.

Remember you may hold the mouse over a button, and then hold down the Mac:  or Win:  keys to have a text help bubble appear that describes the button.



## 19.3 New Menus



### Screencast Video

<http://www.LiveMath.com/v/1903>

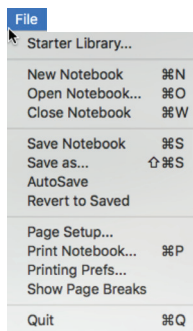
A new feature of *LiveMath* 3.6 is a redesigned menu structure.

The new menus are yet another feature stolen from the *LiveMath* 4.0 development, as I just got sick of using the old *LiveMath* 3.5 menus.

On the Mac side, we have new graphical feature: menu commands that correspond to palette buttons have those palette buttons in the menu. Maybe these menu icons will get into the Windows version in a future update, but for now this is a Mac-only version.

Let's take a look at the new menus.

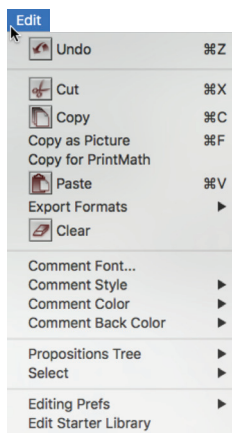
### File Menu



The File Menu is nearly unchanged, with the exception of **Show Page Breaks**, which is explained in §19.8.



## Edit Menu



In addition to the menu icons, the Edit menu has been cleaned up. Of significance are the new features:

- *Copy for PrintMath*

You may select an object in *LiveMath*, use **Copy for PrintMath**, and then go over to *PrintMath* and paste - if the object is something that *PrintMath* can handle (and work with), like integrals and math expressions, it will take the paste and you can edit. If you try this with a graph or something similar, then of course *PrintMath* does not know how to deal with this.

- *Export Formats*

This new feature is discussed at length in §19.4 below.

- *Comment Back Color*

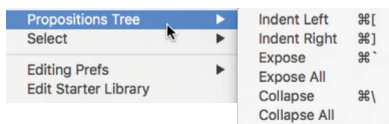
Each comment may have a background color. Due to limitations of original design (going back to *Expressionist* 30 years ago), we cannot back color individual subobjects (words or math expressions inside of comments).

☞ A nice comment

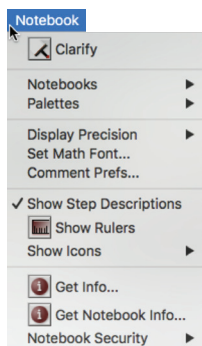
☞ A nicer comment with a bright background

- *Propositions Tree*

This was moved from the old Notebook menu over to here. Seems more natural here. This menu allows you to move propositions to different parts of the notebook tree. Usually you indent via dragging objects in the notebook tree.

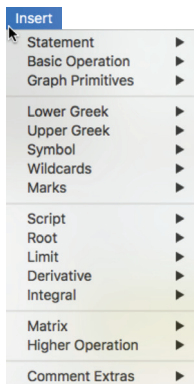


## Notebook Menu



The old **Formatting** menu has been removed, and the two main items in that former menu, **Set Font Math** and **Comment Prefs**, are now on the main level of the menu. Otherwise this menu has not changed.

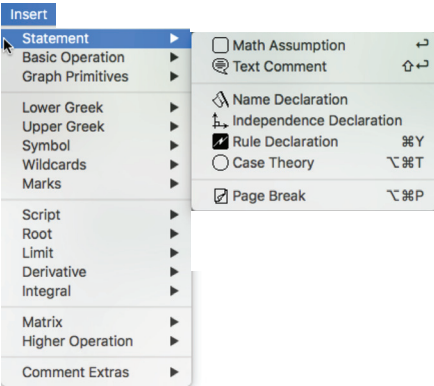
## Insert Menu



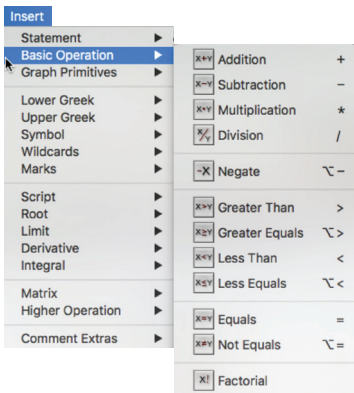
This menu is entirely new. The goal of this menu is to address the age-old question, “but how do I get that in there?” Many of these items are available in the Palette, some were inserted via special (and not well documented) keystrokes. Over the years in working with novice *LiveMath* users, it became clear that a central menu location was required that held all of the possible inputs into *LiveMath*.

Likely you will not use this Insert menu as the primary insertion method, but rather you will continue to use the Palette and/or keystrokes. But when you get to the need to insert something, and you forget where in the Palette it is located, or you are working with a novice user, or forget which keystroke does the insert, this Insert menu will be very helpful. The keystrokes are shown in the menu when there are keystrokes for a specific command.

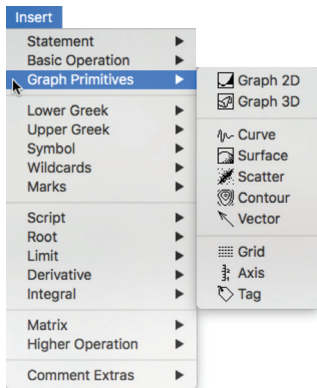
The **Insert ► Statement** submenu has the two most common entry options: New **Math Assumption** and New **Text Comment**. It is unlikely you would use these menu options to get either new objects into a notebook, but for novices just starting out who often get confused about the different between a Math Assumption and a Text Comment, well, there they are. With the keystroke reminders as well.



**Insert ► Basic Operations** submenu is rather obvious not needing explanation:



**Insert ► Graph Primitives** submenu is a menu version of the Graphics Primitives collapsar inside of every New Notebook, for the purpose of getting a blank graphics primitive into the notebook, for those (like me) that like to build up your graphs manually, without all of the entry processes of the Graph menu.

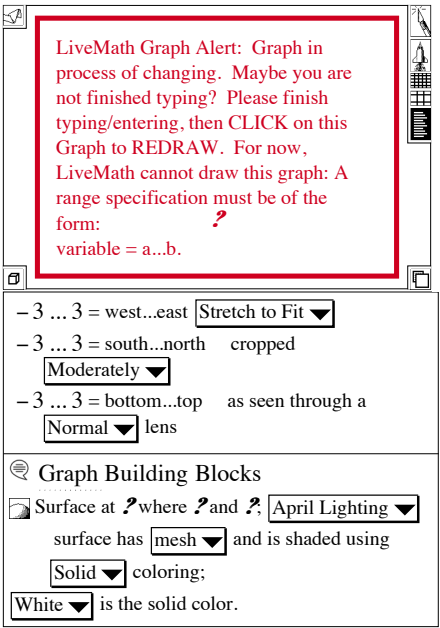


If the current selection is inside of a Graph Theory (2D or 3D), then any of these menu options will insert a new graphics primitive into that theory (even another graph, so beeeeeeee careful!).

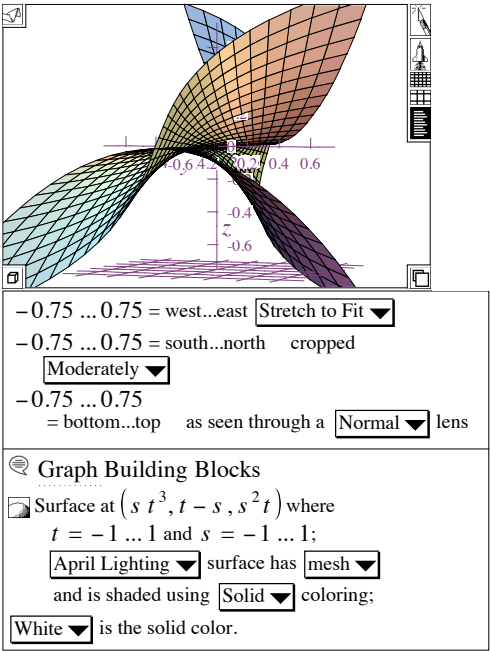
If the current selection is outside of a Graph Theory, these menus will insert the chosen graphics primitive, but outside of a graph theory, the SurfaceAt primitive will not do much (anything).

Surface at  $\mathcal{P}$  where  $\mathcal{P}$  and  $\mathcal{P}$ : April Lighting ▼ surface has mesh ▼ and is shaded using Solid ▼ coloring;  
White ▼ is the solid color.  
 No graph theory

The slots for any of these graphics primitives are blank, so you have to do all of the work to bring life to them inside of a graph theory:



Manually entering the (correctly formatted) data, you can then customize your surface:



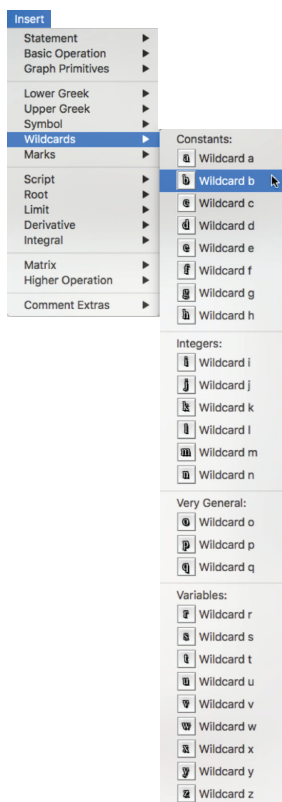
The **Insert ► Lower Greek** and **Insert ► Upper Greek** are self-explanatory, with the Mac: option or Win: ALT keystrokes. The easiest keystrokes for greeks are the apostrophe-based keystrokes:

' A =  $\alpha$

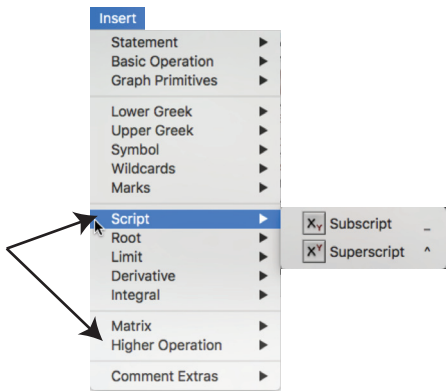
Likewise **Insert ► Symbol** is self-explanatory, giving the miscellaneous symbols available in *LiveMath*.

The **Insert ► Wildcards** gives a visual description of the different types of wildcard variables. These are for advanced users of *LiveMath* who are doing Transform rule programming. But, of course, wildcard variables are used for defining simple functions, too. The keystrokes

? X are not shown, as these are too specialized to get into the menus as we would want to show them. (Mac does not consider ? as a keystroke, although it is to *LiveMath*).

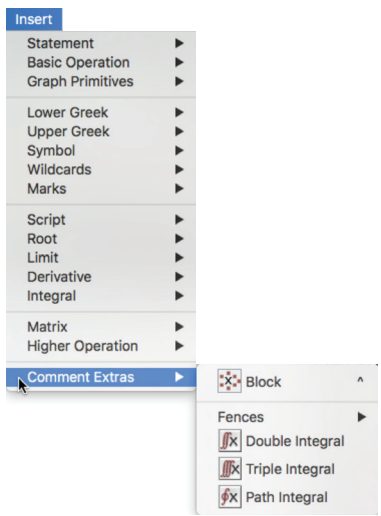


The next insert menus:

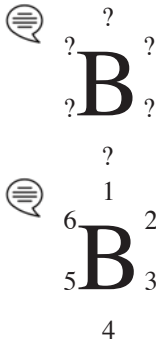


are all self-explanatory.

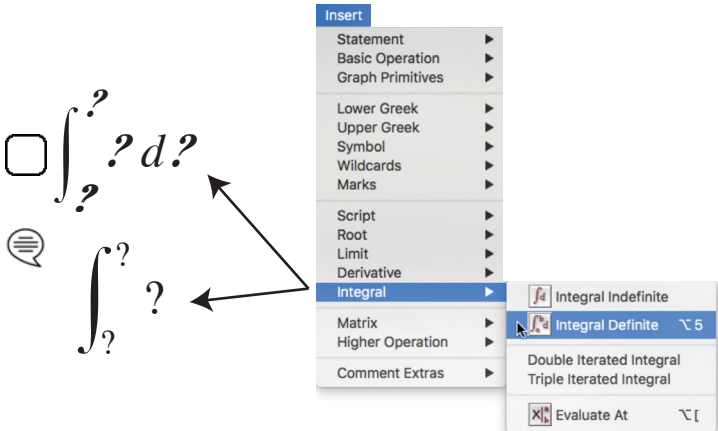
The last submenu, **Insert ► Comment Extras**, are inserts that only work in the comments:



These are constructions that only work inside of the Text Comments, and are borrowed from *PrintMath*. The Block is the most useful in the text comments for building up a complicated object (that may or may not have any mathematical meaning, but certainly does not have support inside of *LiveMath* for calculation purposes)



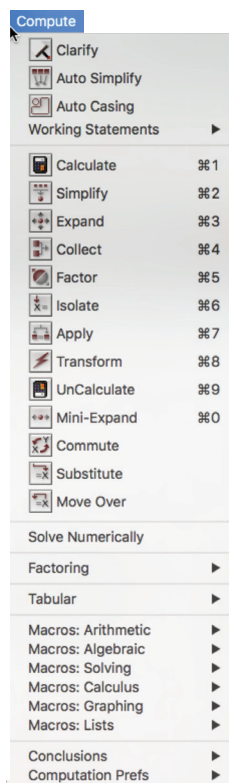
In general, the other mathematical object inserts work for both math computational areas, or in text comments, inserting the different objects appropriately:



### Compute Menu

The **Compute** menu has been reorganized, with the goal of greater simplicity. It is a big menu, and I almost split it into 2 menus, but there didn't seem to be a natural separation point to do that. So, 1 big menu it is.





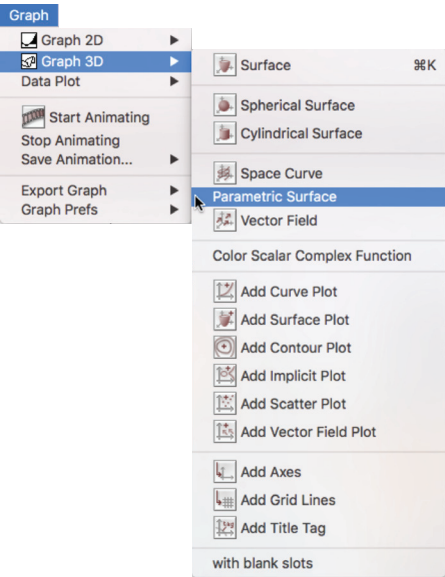
The Solve Numerically options is a menu command that is still under development, and will appear in a future minor release.

The Macros areas are the computations that are “multi-step”, as described in §19.11.

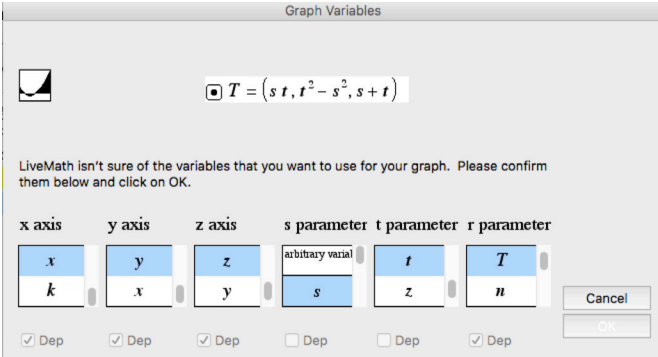
## Graph Menu

The Graph Menu is largely unchanged in LM 3.6, although there is one addition in the **Graph ► Graph 3D** menu with graphing a Parametric Surface via the menu command on a selected 3-vector with 2 independent variables, 1 dependent variable (Matrix variable):

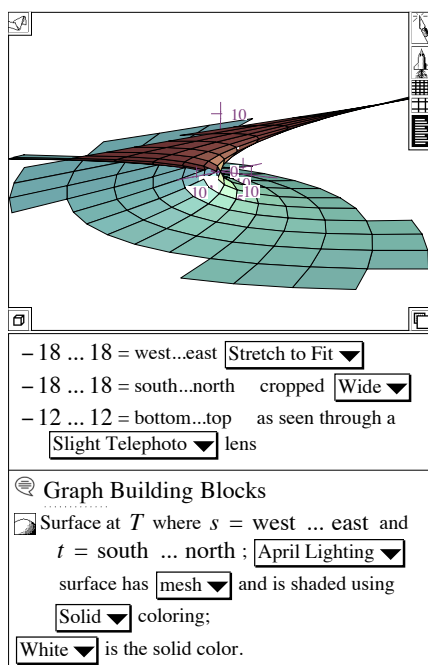
$$\square \quad T = (s \ t, t^2 - s^2, s + t)$$



This will trigger a variable match-up dialog:



and voilà:



## Examples and Help Menus

These menus have not changed, although some references to *LiveMath* Plug-In, which is now retired, have been removed.

## 19.4 Copy/Save as PDF & PostScript



Screencast Video

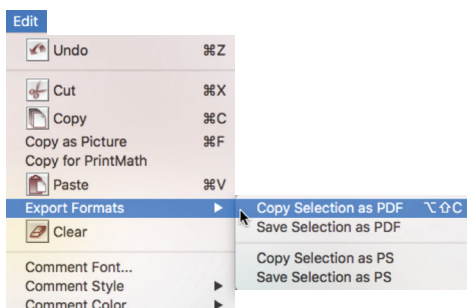
<http://www.LiveMath.com/v/1904>

Nearly 6 years ago, during a “hack attack”, I wrote a little converter to take our wonderful PostScript output of all things in *LiveMath* (and *PrintMath*), and convert it to PDF using a nice little Apple API function I found. (It is the same function that Preview uses when you double-click on a PostScript .ps file, converting it to PDF, which is the Mac’s “natural” graphics format since OSX).

This little feature lay dormant in the source codes until now. Better

late than never I suppose.

You may now highlight objects in a *LiveMath* notebook, and select:



and you get into the Clipboard the beautiful PostScript-quality PDF output for pasting into other programs like Microsoft Word, or Adobe InDesign, which is the layout program this *The LiveMath Book* is written in. So you can see the quality of this output method in this book.

Note the keystroke: option shift C It comes in very handy.

You may also Save selections out to PDF files, or copy/save the raw PostScript (PS) data.

The Copy As PDF options are only available on the Mac side. I have to write the cross-platform PDF writer, which is under development, and will appear on the Windows and Linux versions in a future minor release.

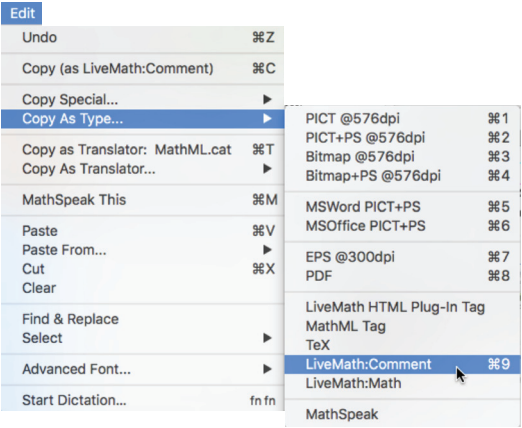
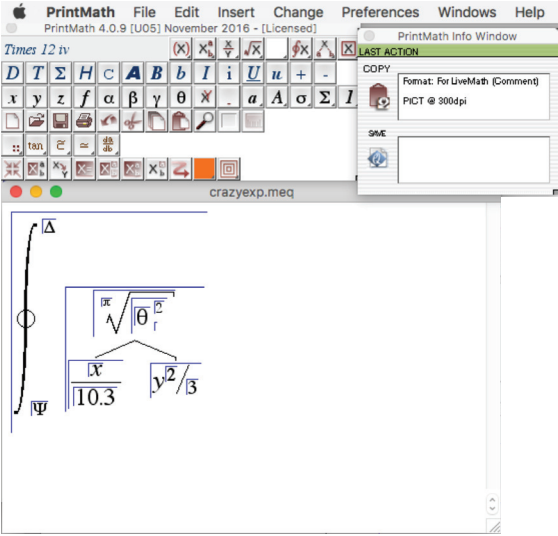
## 19.5 Copy/Save from *PrintMath*



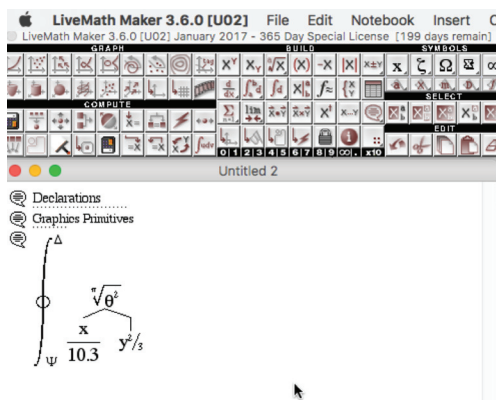
Screencast Video

<http://www.LiveMath.com/v/1905>

If you have more complicated math expressions you wish to have in Text Comments, you may use *PrintMath* to create these expressions, and copy/paste them into *LiveMath* Text Comments.



And then copy/paste into *LiveMath*:



You might want to get wild and create PDFs to copy/paste into *LiveMath*. That won't work with the current older-technology version of *LiveMath* on Mac (or Windows for that matter). On the Mac, you will need to use Preview to copy/paste your PDF to first, which will convert the PDF to an older PICT-image. On Windows you will do essentially the same, but using Paint.

## 19.6 Drag & Drop Add To Graph

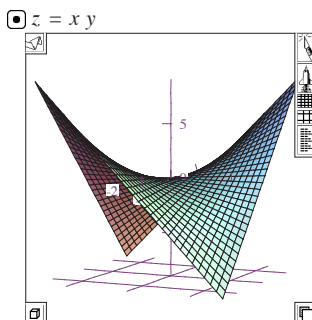


Screencast Video

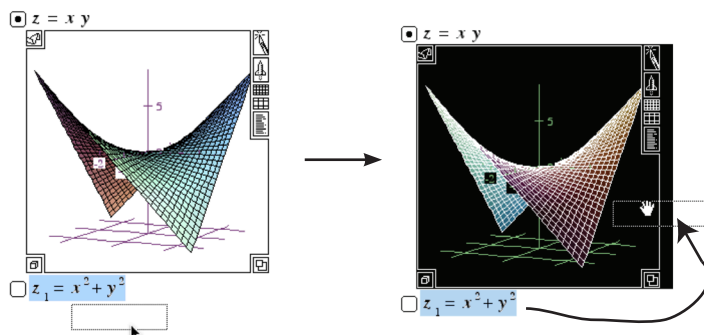
<http://www.LiveMath.com/v/1906>

### Drag & Drop Equations Onto Graph Theories

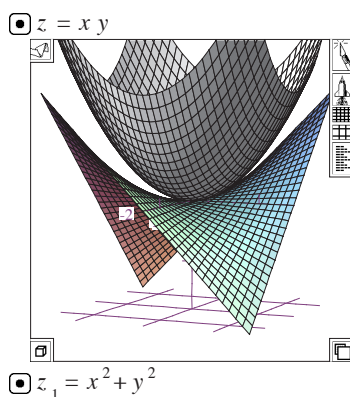
When you have a graph object, like this:



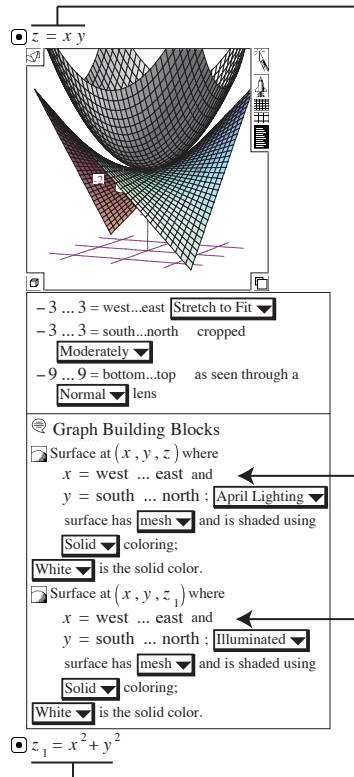
you may wish to add another graphical object to this 3D graph: let's say another surface plot. The (new to LM 3.6) method is to simply drag your new equation onto the 3D graph:



And the result (after approving the variable-to-space coordinates dialog)



Let's look inside of the Graph Details box to see what that Drag & Drop action did for us:



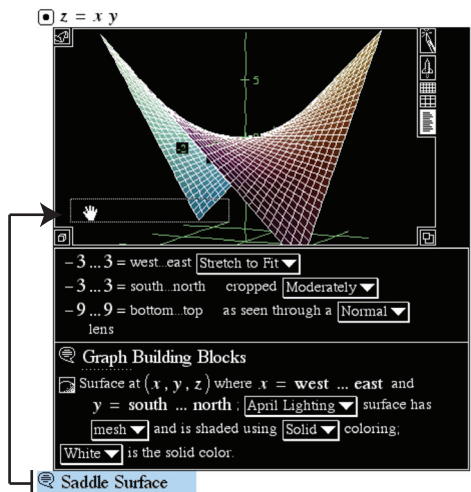
We can see that a second **SurfaceAt** graphic primitive was placed inside of this 3D graph object, with the  $z_1$  dependent variable in the  $z$ -coordinate slot. You could now fine tune this new **SurfaceAt** object to a different color scheme, or different mesh setting, etc.

## Drag & Drop Comments Onto Graph Theories

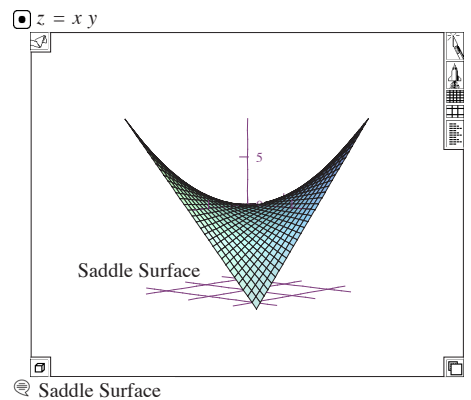
The Drag & Drop feature attempts to figure out what you want to do with the drag, and tries to be as forgiving as it can be for novice mistakes.

If you drag a Text Comment into Graph object, a *copy* of the Text Comment will be used to create a **TagAt** graphic primitive inside of the Graph object:

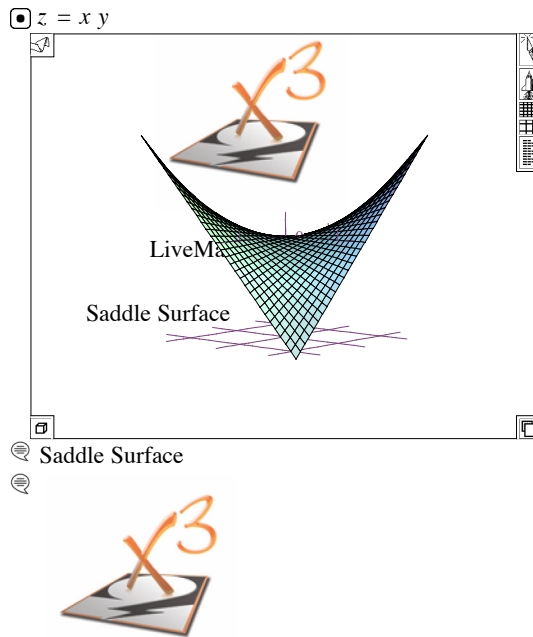




The drag results in:



The drop point inside the graph is used to place the Comment, so if you drop the comment on the bottom left of the graph window, the **TagAt** is placed there (approximately, of course, in the 3D graph window case). You may, of course, adjust the new **TagAt** graphic primitive to customize it as you like. You may also have a text comment with images inside of it - those, too, will get inserted into the **TagAt** text comment:



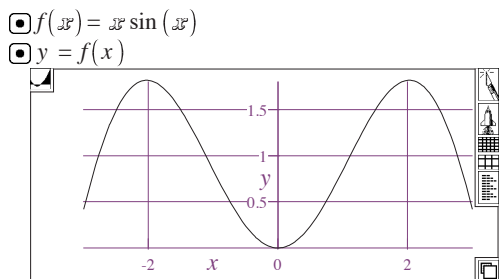
LiveMath rocks!

Anything you can put inside of a Text Comment: text, typeset expressions, images - all of these can go inside of a live Graph object.

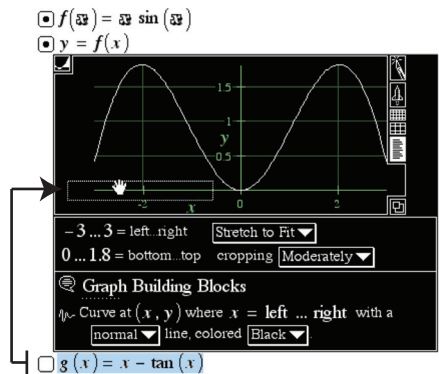
Remember that the text comment is copied into the Graph object at the time you make the Drag & Drop action. If you change the Text Comment later, the copy inside of the Graph object does not change with it.

## Drag & Drop Functions Onto Graph Theories

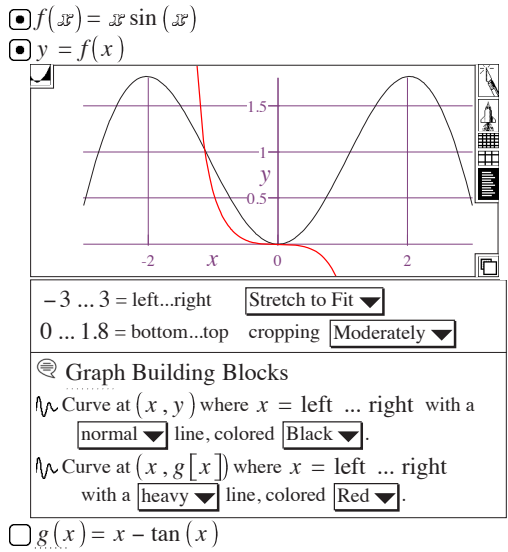
*LiveMath* is notoriously persnickity about how you graph a function  $f(x)$  or  $f(x,y)$ . Technically you should set up a function defined with a wildcard variable, and then create a dependent variable relation with the function, like this:



With Drag & Drop, we loosened the rules a little bit. So you could define a function  $g(x)$  and drag & drop that onto the 2D Graph object, and *LiveMath* will create the dependent variable relation for you:

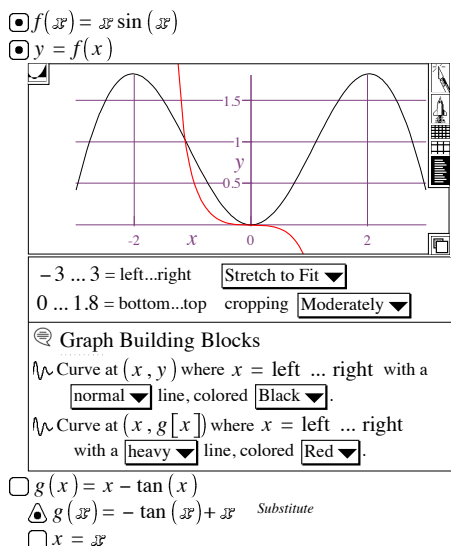


Results in:



Notice that *LiveMath* put the function actually in the y-coordinate spot in the CurveAt graphic primitive. You may be wondering: how did *LiveMath* do that function call without wildcard variables?

Ahhhhh. *LiveMath* cheated for you a little. Notice there is a little collapsar below the  $g(x)$  definition. Let's have a look:



*LiveMath* did the standard trick of substituting

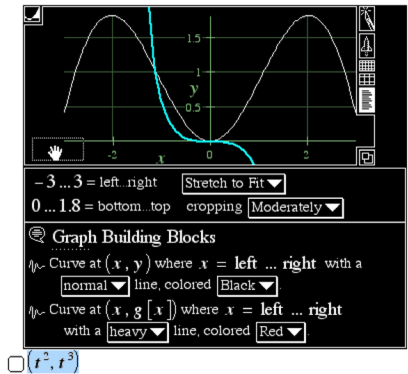
$$x = \mathcal{X}$$

into the regular variable definition of  $g(x)$  to create the wildcard definition of  $g(x)$ , and made a Hot Dot on that substitution Conclusion statement to make it the Working Definition of  $g(x)$ .

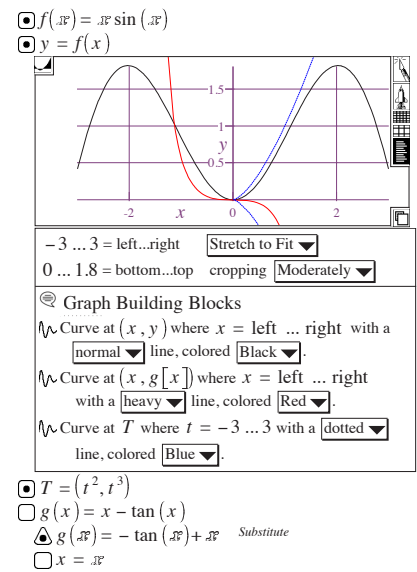
We do this trick manually all the time. The Drag & Drop mechanism does it for you, trying to save yourself from yourself.

## Drag & Drop Parametrics Onto Graph Theories

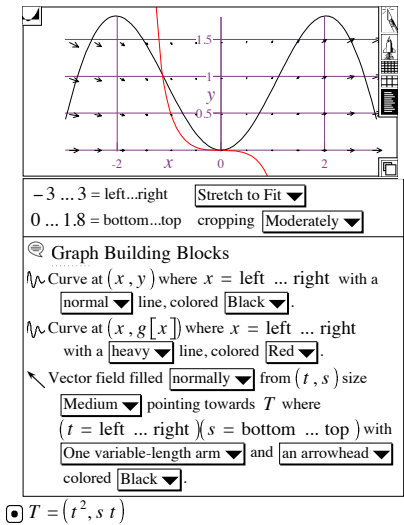
If you try to drag a 2-vector like this:



You won't get very far. *LiveMath* will just go thud. But put a nice dependent Matrix in a relation with that 2-vector, and you'll be good to graph with Drag & Drop:



There are limitations for what *LiveMath* will try to compensate for. If that T vector has what looks to be 2 independent variables, then *LiveMath* will interpret that as a vector field plot request:



### Drag & Drop With Dependent Variables Onto Graph Theories

Usually *LiveMath* is very unforgiving with multiple dependent variables that are the same symbol - you can only Hot Dot one of the equations at a time - that is a fundamental construction (and power) of *LiveMath*:

Same Dependent  
Variable Symbols:  $y$

$y = x \sin(x)$   
 $y = -x^2 + 1$

But when you Drag & Drop an equation that may have a dependent variable conflict with a previously graphed equation, *LiveMath* will compensate with the some trickery to try to save yourself from yourself: subscripts are introduced, and the subscripted equation is hidden under a collapsar of the original equation:

Drag & Drop

New Equations with subscripts and substitution automatically inserted under collapsar

☐  $y = -x^2 + 1$

☐  $y_1 = y$

☒  $y_1 = -x^2 + 1$  *Substitute*

☒  $y = f(x)$

$-3 \dots 3 = \text{left} \dots \text{right}$  **Stretch to Fit**

$0 \dots 1.8 = \text{bottom} \dots \text{top}$  **cropping** **Moderately**

**Graph Building Blocks**

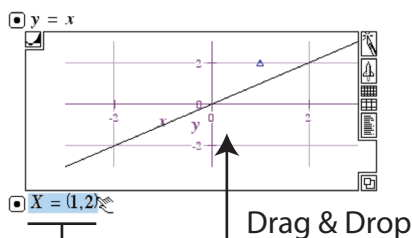
Curve at  $(x, y)$  where  $x = \text{left} \dots \text{right}$  with a **normal** line, colored **Black**.

Curve at  $(x, y_1)$  where  $x = \text{left} \dots \text{right}$  with a **normal** line, colored **Orange**.

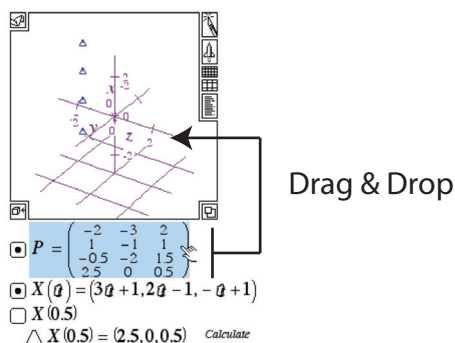
The reason: this is what you would do yourself, manually, to get past the multiple dependent variable problem. But for many of us, we just like to see graphing equations with a pure “y” dependent variable, without always having to go to subscripts or different letters.

## Drag & Drop Points and Matrices Onto Graph Theories

You may also drag points and/or matrices to a graph window:



of a matrix of points:

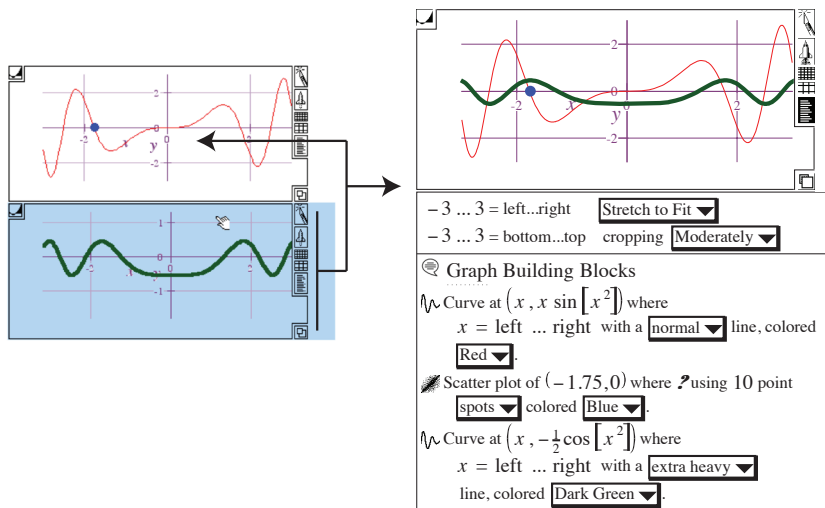


Make sure you have the dimensions correct! Otherwise, *LiveMath* is not going to fix things up for you!

## Drag & Drop One Graph Theory Onto Another Graph Theory

Sometimes you are building up a graph, and you want to get all of the (major) graphics primitives copied over to a new graph. You may do this easily by Drag & Drop one Graph Theory onto another Graph Theory.

All of the major Graphics Primitives will be copied over to the new graph.





If you drag a 2D Graph onto a 3D graph, the 2D graph objects objects are copied over to the x-y plane (or the equivalent in that graph theory).


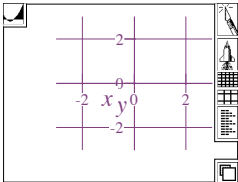
If you drag a 3D Graph onto a 2D Graph, all of the 3D graphics primi- tives are squished to the x-y plane (simple coordinate projection).

The minor graphics primitives are not copied over with this operation: Grid, Axis, nor any of the Name Declarations nor functional defini- tions. This would cause havoc is everything were copied over. The goal of this operation is to quickly combine simple graphs together, not address every pathological example of specialized *LiveMath* program- ming.

Drag & Drop Catalog

Here is a complete listing of all Drag & Drop Add Graph actions

All Graphs

Drag Object	Example	Result
Point/Matrix	$X = (1,2)$	<b>Scatter</b>
Text Comment	 Text!	<b>Tag</b>
Graph Theory		Major Primitives Adjusted for Dimension

2D Graphs

Drag Object	Example	Result
Function $f: \mathbb{R}^1 \rightarrow \mathbb{R}^1$	$f(x) = x^2$	<b>CurveAt</b>

Function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^1$	$f(x, y) = x^2 y^3$	<b>Contours</b>
Function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$	$f(x, y) = (x y^2, x^2)$	<b>VectorField</b>
Equation 2 Variables	$y = x^2$	<b>CurveAt</b>
Equation 3 Variables	$z = x^2 + y^2$	<b>Contours</b>
Equation 2-Vector	$T = (t^2, t^3)$	Parametric Curve
Equation 3-Vector	$T = (t, t^2, t^3)$	Parametric Curve Projected to x-y plane
Equation 2-Vector	$F = (s t, t^2)$	<b>VectorField</b>
Equation 3-Vector	$F = (s t, s^2, t^2)$	<b>Contours of</b> Parametric Surface

3D Graphs

<u>Drag Object</u>	<u>Example</u>	<u>Result</u>
Function $f: \mathbb{R}^1 \rightarrow \mathbb{R}^1$	$f(x) = x^2$	<b>CurveAt</b> in x-y plane
Function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^1$	$f(x, y) = x^2 y^3$	<b>SurfaceAt</b>
Function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$	$f(x, y) = (y, x y, x)$	Parametric Surface
Function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$	$f(x, y) = (x y^2, x^2)$	<b>VectorField</b> in x-y plane
Function $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$	$f(x, y, z) = (y^2, z^3, x)$	<b>VectorField</b>
Equation 2 Variables	$y = x^2$	<b>CurveAt</b> in x-y plane
Equation 3 Variables	$z = x^2 + y^2$	<b>SurfaceAt</b>

Equation 2-Vector	$T = (t^2, t^3)$	Parametric Curve in x-y plane
Equation 3-Vector	$T = (t, t^2, t^3)$	Parametric Curve
Equation 2-Vector	$F = (s\ t, t^2)$	<b>VectorField</b> in x-y plane
Equation 3-Vector	$F = (s\ t, s^2, t^2)$	Parametric Surface

---

## 19.7 Marks



Screencast Video

<http://www.LiveMath.com/v/1907>

This little feature has been in place for many years, but hidden and undocumented.

If you type into a math assumption:  $xVEC$

You will magically get this:  $\vec{x}$

The suffix VEC on the end of any Name Declaration gets magically transformed into a Vector mark.

Historically *LiveMath* “names” in Name Declarations (variables, functions, et al.) are strings of plain characters. In order to preserve backwards compatibility with earlier minor releases of *LiveMath* 3.5, I adopted this suffix VEC trick to get marks onto the variable or function names; if an older version of *LiveMath* opened a notebook with a  $\vec{x}$  that earlier version would simply see a  $xVEC$  as a variable name - a bit odd, but it would not crash nor cause a variable conflict.

You may put a VEC suffix on any Name in *LiveMath*, so here are some other examples:

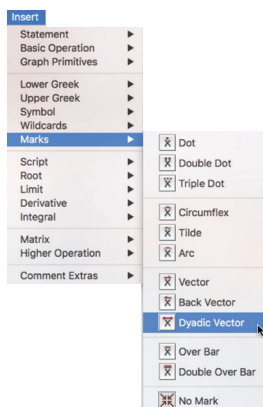
$abc\text{VEC} \longrightarrow \overrightarrow{abc}$

$LiveMathIsCool\text{VEC} \longrightarrow \overrightarrow{\text{LiveMathIsCool}}$

The following suffices will give the corresponding marks:

xDOT	$\dot{x}$
xDDT	$\ddot{x}$
xDDD	$\dddot{x}$
xTIL	$\tilde{x}$
xVEC	$\overrightarrow{x}$
xDVE	$\overleftrightarrow{x}$
xBAR	$\overline{x}$
xHAT	$\hat{x}$
xARC	$\frown x$
xBVE	$\overleftarrow{x}$
xDBA	$\overline{\overline{x}}$

You may also make Marks using the menu command in the Insert menu:



The No Mark menu item in this submenu will undo a mark.

**Warning:** when you select and “ $x$ ” and convert it into a  $x\text{VEC} = \vec{x}$  using the menu command, you are converting EVERY  $x$  in the current notebook into an  $x\text{VEC} = \vec{x}$ . You are actually changing the Name Declaration that defines the name “ $x$ ” into “ $x\text{VEC} = \vec{x}$ ”. So be careful! If you change such a variable globally, and you didn’t really want to, just go find the Name Declaration and overtype over the  $x\text{VEC} = \vec{x}$  just  $x$  - this will convert it back.

You may also do marks in Text Comments areas as well.

## 19.8 Print Page Preview and Improved Pagebreak Printing



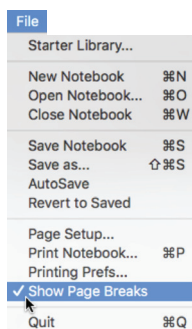
Screencast Video

<http://www.LiveMath.com/v/1908>

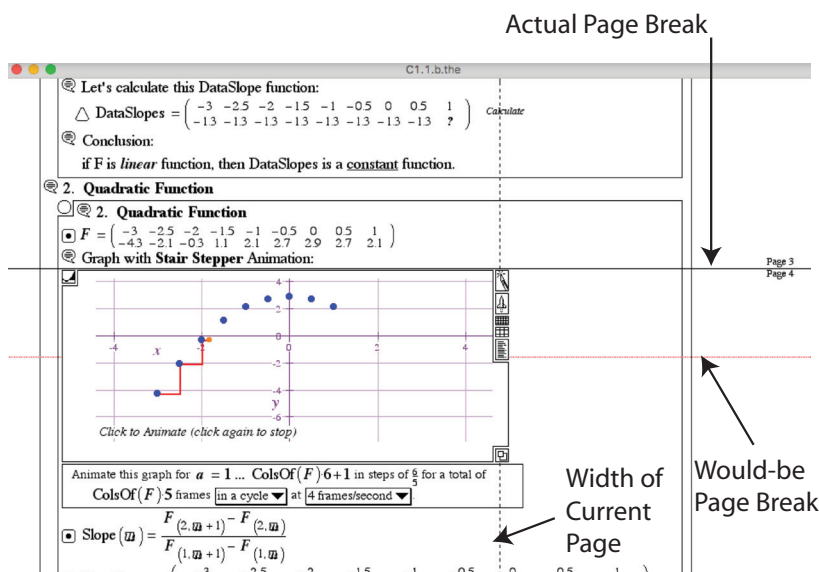
Suppose you have a big *LiveMath* notebook, and you want to print it out (or output to PDF via the Print function, on the Mac with Save As PDF, and on Windows with PrimoPDF or some other tool).

You may wish to fine-tune Page Break locations and other adjustments so that the final output is more appealing.

To turn on Show Pagebreaks, select the menu item in the **File** menu:

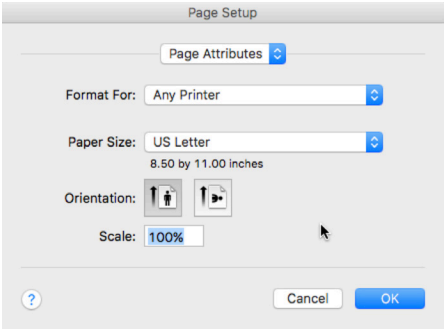


Next, hold down the **option** on Mac or **ALT** on Windows, and you will see the page delimiters show up:

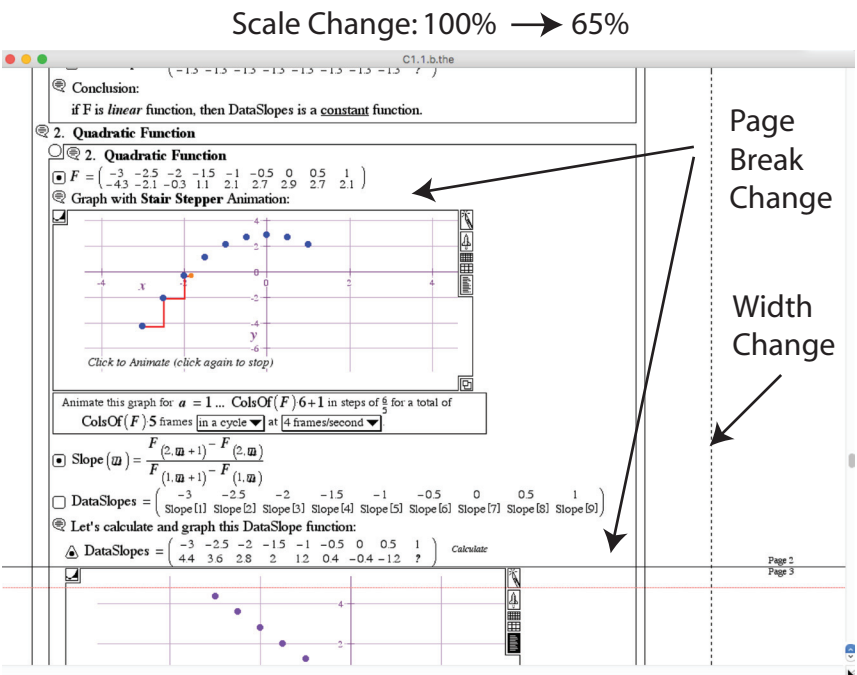


The red line shows where the page break would be, but in this example, the graph theory is in the way, so the page break needs to come before the graph theory.

All of the calculations of page breaks are dependent upon the chosen printer and Page Setup dimensions. In the example above, the Page Setup shows this:



If we change the Scale to 65% instead, we will see the change in the Page Breaks:



I have this two part approach to showing these page delimiters because the calculations of these page cut-offs is very hard on the system: there are a lot of calculations going on, and you don't want that heavy burden of calculations happening all the time - especially while you might be doing some drag calculations or something. This is a limitation of the older QuickDraw/Mac and GDI/Windows technologies.

## 19.9 Inequality Solving



Screencast Video

<http://www.LiveMath.com/v/1909>

A feature that was snuck in a few years ago as I wrote a Precalculus e-text was support for solving inequalities. As a first example, start with:

$$\square -5x + 3 > 7$$

With AutoSimplify OFF, you may complete this series of computations:

$$\square -5x + 3 > 7$$

$$\triangle -5x > 7 - 3 \quad \text{Move Over}$$

$$\triangle x < \frac{1}{5}(-[7 - 3]) \quad \text{Isolate}$$

$$\triangle x < -\frac{4}{5} \quad \text{Simplify}$$

You can do computations involving cases, similar to solving polynomial equations:

$$\square x^2 - 2x + 6 > 9$$

$$\triangle (x^2 - 2x + 6) - 9 > 0 \quad \text{Move Over}$$

$$\triangle x^2 - 2x - 3 > 0 \quad \text{Simplify}$$

$$\triangle (x - 3)(x + 1) > 0 \quad \text{Factor}$$

$$\square \square x + 1 > 0$$

$$\triangle x > 0 - 1 \quad \text{Isolate}$$

$$\triangle x > -1 \quad \text{Simplify}$$

$$\triangle x - 3 > 0 \frac{1}{x + 1} \quad \text{Isolate}$$

$$\triangle x - 3 > 0 \quad \text{Simplify}$$

$$\triangle x > 0 + 3 \quad \text{Isolate}$$

$$\triangle x > 3 \quad \text{Simplify}$$

$$\square \square x + 1 < 0$$

$$\triangle x < 0 - 1 \quad \text{Isolate}$$

$$\triangle x < -1 \quad \text{Simplify}$$

$$\triangle x - 3 < 0 \frac{1}{x + 1} \quad \text{Isolate}$$

$$\triangle x - 3 < 0 \quad \text{Simplify}$$

$$\triangle x < 0 + 3 \quad \text{Isolate}$$

$$\triangle x < 3 \quad \text{Simplify}$$

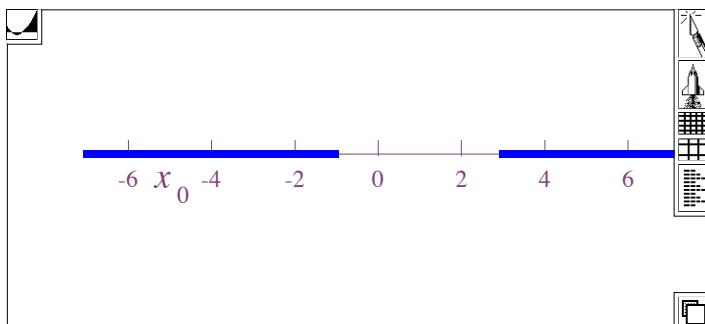


Note that each Case Theory here has the assumption conditions at the top line, so the solutions are read off as:

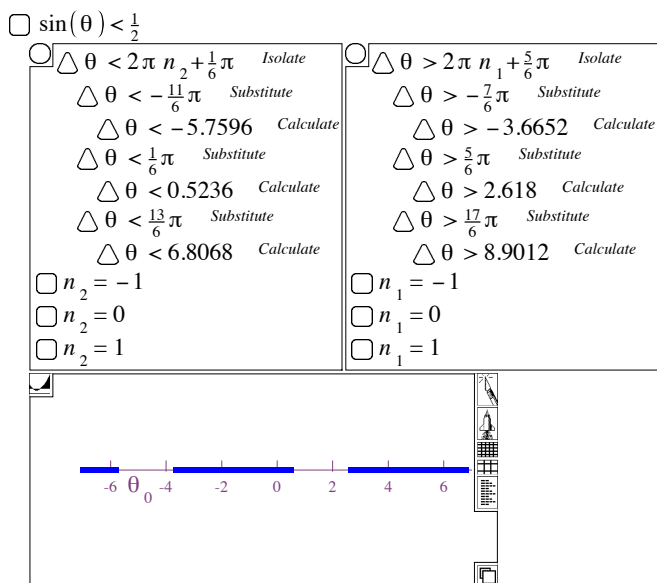
$$x > -1 \text{ and } x > 3 \longrightarrow x > 3$$

$$x < -1 \text{ and } x < 3 \longrightarrow x < -1$$

We can see this solution also with one of the (graphing) Macros, in menu **Compute ► Macros: Graphing ► Graph 1D Inequality**



You may also try inequalities with functions as well:



There are some other graphical solution Macros in the **Compute ► Macros** menu.


## 19.10 Contour Plot Changes



Screencast Video

<http://www.LiveMath.com/v/1910>

Sometime since the last major update, a few small changes were inserted into the Contour Plot primitive:

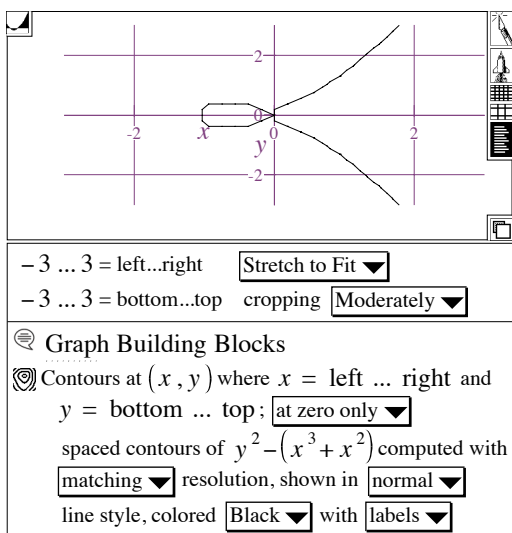
 Contours at  $\mathcal{P}$  where  $\mathcal{P}$  and  $\mathcal{P}$ ,  spaced contours of  $\mathcal{P}$  computed with  resolution, shown in  line style, colored  with

Often Contour Plots need a higher amount of resolution (more points) to make their graphs smoother. But if you have other graphic primitives in the same Graph Theory, you don't necessarily want those other primitives to have their resolution cranked up. So Contour Plots may have matching resolutions to the Graph Theory, or you may change that popup menu to a different, independent setting.

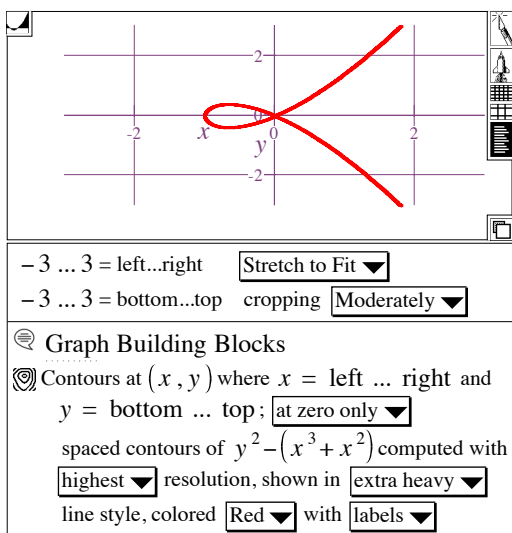
It used to be that Contour Plots had only a single line style - now Contour Plots may have many different line styles, with Extra Heavy the most useful, especially in a complicated Graph Theory.

It used to be that Contour Plots had only a fixed color: Black. Now Contour Plots may have multiple colors like any other CurveAt plot.

Here is what Contour Plots could look like previously:



And now with these new customizations:



## 19.11 Macros

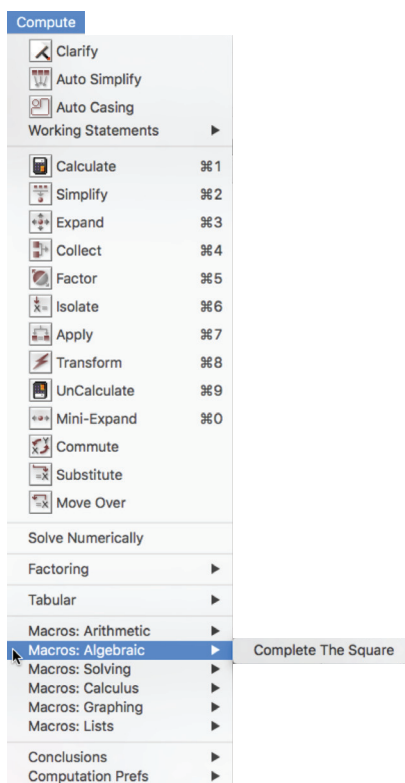


Screencast Video

<http://www.LiveMath.com/v/1911>

The Macros area of the Compute Menu were designed to be semi-automated constructions in *LiveMath* that you could do manually, but it would be a lot of work to do, and why not have a nice one-shot way of doing these constructions.

An excellent example is the Macro for Completing the Square, a standard algebraic computation in high school algebra. For years, high school math teachers and students would write to me to ask, “Can *LiveMath* Complete The Square?” The philosophy of *LiveMath* dictates that the answer should be, “Yes, of course, but you need to do it yourself!”



But it seemed that we needed to step back from that philosophy a little bit, and help out with some calculations like this, rather than just telling wayward students (and teachers) “yes, but do it yourself”. These kinds of requests collected together to create the Macros menu

area and functionality.

A typical problem in high school algebra of Completing the Square might start with this equation:

$$y = x^2 + 3x$$

Selecting this equation and then the menu **Compute ►**  
**Macros: Algebraic ► Completing the Square** gives us:

$$\begin{aligned} \square y &= x^2 + 3x \\ \triangle (y - 0) + \left(\frac{3 \cdot 1}{2 \cdot 1}\right)^2 \cdot 1 &= (x^2 + 3x) + \left(\frac{3 \cdot 1}{2 \cdot 1}\right)^2 \cdot 1 && \text{Complete Square} \\ \triangle y + \frac{9}{4} &= x^2 + 3x + \frac{9}{4} && \text{Simplify Simplify} \\ \triangle y + \frac{9}{4} &= \left(x + \frac{3}{2}\right)^2 && \text{Factor} \\ \triangle y &= \left(x + \frac{3}{2}\right)^2 - \frac{9}{4} && \text{Move Over} \end{aligned}$$

As you can see, all of the functionality shown in this computation exists in *LiveMath* already, so it is not like there is a new “function” called “Completing the Square” where the beginning equation is simply transformed to the final answer. The idea of Macros was to show students (and others) the steps involved in doing this algebraic computation in *LiveMath*, that they could do on their own manually, or use the Macro to get a little help from *LiveMath* on the steps to complete.

Let’s look at each of the Macros menus a bit more in detail.

## Macros: Arithmetic

Macros: Arithmetic	▶	Round to Nearest Integer
Macros: Algebraic	▶	Round Down to Integer
Macros: Solving	▶	Round Up to Integer
Macros: Calculus	▶	
Macros: Graphing	▶	Factor Pairs
Macros: Lists	▶	
Conclusions	▶	Reduce Root
Computation Prefs	▶	Rationalize Denominator
		Real Part
		Imaginary Part
		Complex Conjugate

The Rounding macros are rather obvious in their goal. In the style of Macros, these options actually try to teach you how you would set up these functions in *LiveMath* manually: a new function is defined, named “round”, that uses one of the predefined functions already built into *LiveMath*.

⚙️ A Function ▼ named round behaves as Round to nearest integer ▼.

☐ 13.5

☐ round(13.5)

△ round(13.5) = 14     *Calculate*

The Factor Pairs menu was an interesting request from a teacher years ago: to show all factorizations of a number, in factor pairs.

☐ 25·49

△ 25·49 = 1225     *Simplify*

△ 25·49 = 1225·1     *Factor Pair*

△ 25·49 = 245·5     *Factor Pair*

△ 25·49 = 175·7     *Factor Pair*

△ 25·49 = 49·25     *Factor Pair*

△ 25·49 = 35·35     *Factor Pair*

The Reduce Root helps a beginning in *LiveMath* deal with root simplification. If you try to Simplify this square root:

☐  $\sqrt{175}$

Nothing will happen. Simplify will not first factor the interior of

the square root: you should do that manually yourself, being a good mathematician. But in case you just learning how *LiveMath* works, this Reduce Root macro will help you, and show you the process you should follow:

$$\begin{aligned} &\square \sqrt{175} \\ &\triangle \sqrt{175} = \sqrt{5^2 \cdot 7} \quad \text{Factor Root Base} \\ &\triangle \sqrt{175} = 5\sqrt{7} \quad \text{Expand} \end{aligned}$$

Rationalize Denominator is another computational process that users of *LiveMath* should do on their own, but the Macro helps you a little bit to get there:

$$\begin{aligned} &\square \frac{17}{\sqrt{11}} \\ &\triangle \frac{17}{\sqrt{11}} = \frac{17\sqrt{11}}{\sqrt{11}\sqrt{11}} \quad \text{Multiply By One} \\ &\triangle \frac{17}{\sqrt{11}} = \frac{17\sqrt{11}}{11} \quad \text{Expand Expand} \end{aligned}$$

The Real and Imaginary Parts macros are similar to the Rounding macros: in *LiveMath* you should define a function and apply that function to the object, using the built-in functions for Real and Imaginary parts. Here the “Re” function is created and defined:

✎ A Function ▼ named Re behaves as Real part ▼.

$$\begin{aligned} &\square 19 - 7i \\ &\square \text{Re}(19 - 7i) \\ &\triangle \text{Re}(19 - 7i) = 19 \quad \text{Calculate} \end{aligned}$$

The Complex Conjugate computation should be something that *LiveMath* users do manually, but the Macro helps you this conclusion a little bit:

✎ A Function ▼ named Conjugate behaves as Complex Conjugate ▼.

$$\square 13 - 4i$$

$$\square \text{Conjugate}(13 - 4i)$$

$$\triangle \text{Conjugate}(13 - 4i) = 13 + 4i \quad \text{Calculate}$$

## Macros: Solving

---

## 19.12 Miscellaneous



Screencast Video

<http://www.LiveMath.com/v/1912>

Arrows for scrolling up and down the notebook.

- Scroll arrows at bottom right of window for fine-tuned scrolling. No, we can't do gestures using older QuickDraw technology.
- f' made nicer: extra space for prime on f', but not for other letters. Just a cheap hack for dealing with f ligature.
- Big Palettes: twice a big as small palette, with new artwork (stolen from LM4)
- Printing: case theories will now print nicely, more compact, more intelligent split up of pages.
- Printing to PS/PDF: now working correctly, using Postscript to create the PDF and Preview
- New Menu Layout
- Insert Menu
- Name Declaration (variable, function) marks. For backward compatibility, xVEC gives a x with a vector hat on top. The insert->marks menu now changes that 3 letter magic suffix. when you use this you are actually re-defining the variable, so be careful! You can type these in manually as well. Give all 11 magic 3-letter suffices.



- Expanded math (from *PrintMath*) in comments
- Multiselect: Option key down, single-click on name/variable will highlight all instances of this variable in the current object
- Drag Equation to Add to Graph:
- Drag comment to graph: make Tag out of it. Gets placed where you dropped it.









