1 Basic Introduction

If you are new to LiveMath, you don't necessarily want to drudge through a big software manual to just get started doing some math. So this first chapter is just to get started.

1.1 Why is this program called "LiveMath" ?

LiveMath is a computer algebra and graphing program.

These types of programs were originally called "CAS" = computer algebra systems. The biggest such program is *Mathematica*, with others being *Maple*, *MatLab*, and other minor computational engines, both large and small, free and commercial.

Nearly all of these math software programs, with the exception of the *MathCad* program, are closely tied to a programming language - using command-line syntax to perform computations, make graphs, etc.: type in a command in some sort of text syntax, then "hit Enter" to send those commands to the computation engine, and wait for the result.

LiveMath was originally called *Theorist*, and was developed as a computational engine on top of the math equation editor *Expressionist*. This original *Theorist* was not created as an "input/output" engine, but rather as a *steady-state* machine.

This means that when you open up a *LiveMath* notebook, it will do computations, graphs, etc. to achieve a steady-state. If you *change* something inside of the notebook - a number, or a variable, or something - the *LiveMath* engine will bring the notebook back to a steady-state, and do the computations, graphs, etc necessary that have been affected by your *change*.

In this sense, the notebook is ready to do *live mathematics*. Imagine you had some equations written on a blackboard - a series of

computations. If you change one of the numbers in a starting equation, that would mean the computations will have to re-work themselves to the conclusions. On a blackboard, you would have to be the computational engine to re-perform these calculations. Inside of the *LiveMath* program, it is the *LiveMath* engine that does these recalculations for you.

The logo for *LiveMath*, with the lightening bolt, it meant to symbolize this property of the software: if you change something, get ready for the sparks to fly, as each and every computation inside of the notebook is *live*, as in electricity.

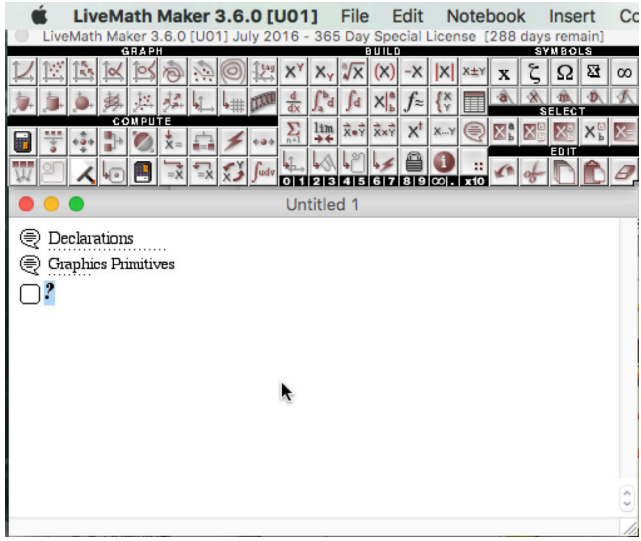
Pictured above is the old LiveMath retail box, back when software was sold in boxes in stores. Ah, those were the days, having to shrinkwrap boxes with floppy disks inside!

1.2 The Notebook

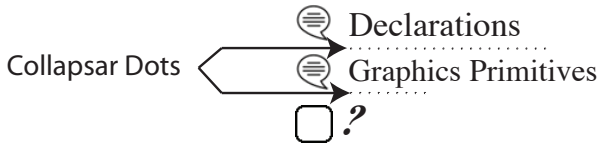
LiveMath files are called *notebooks*. Why? I'm not sure. Probably because the files in *Mathematica* were called "notebooks".

The idea is that the computations inside of a single LiveMath disc file are self-contained: you can't combine two different notebooks, or "include" one notebook inside of another notebook. You can copy and paste between two notebooks, or course, but the *knowledge* in one notebook is not automatically shareable nor known to another notebook. So in this sense, a notebook is a basic atom of *LiveMath* knowledge.

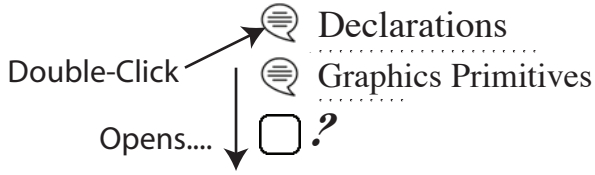
So when you open up a New Notebook in *LiveMath* via the menu command **File ▶New Notebook** you will see "blank" notebook:

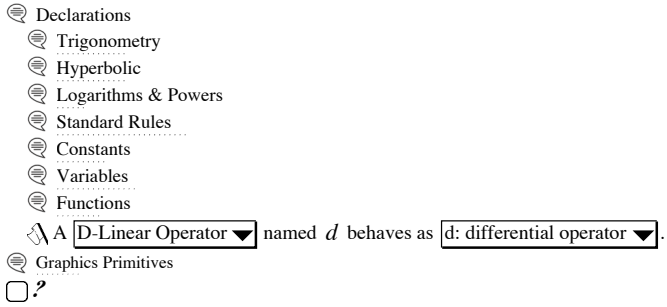


But it is actually not "blank" at all, but rather just minimally equipped. (You may customize your "blank notebook" to make it the way you want it - see Chapter XXX on Customization).



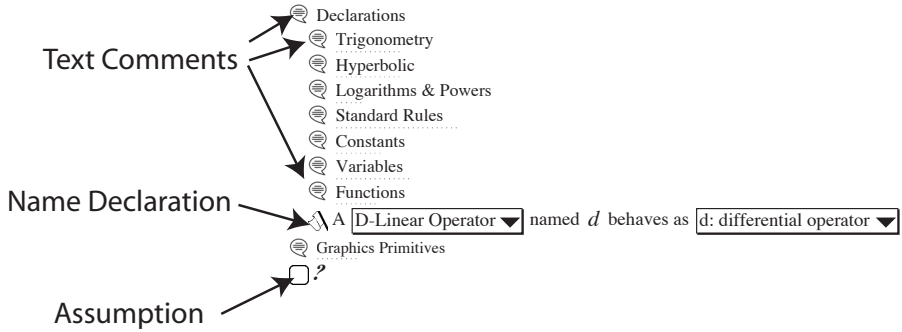
Any time you see these little dots below an object, it means there is "more stuff" underneath the *collapsar*. You may "open up" the collapsar by double-clicking on the object icons, which will reveal the additional hidden content.





This is a nice way of hiding *child objects* in *LiveMath*. So if you hunt around this blank new notebook, you will find out it is not so blank at all: lots of stuff in here, just hidden.

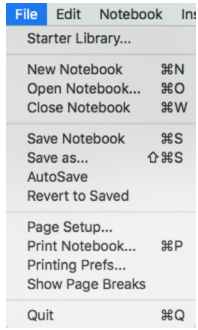
So what are all of these things? Each object in LiveMath has an icon to visually tell you what it is. The first few icons we see tell us about an initial set of objects:



We will explain more about the icons in the next section. The big thing to understand at this point is that each object has an icon, and to select that object, you click on its icon.

Each object in LiveMath has a certain place in the notebook *tree*. Objects cannot just be anywhere inside of the notebook in freeform, but they must be in some "slot" or tree position. These slots have (nearly) nothing to do with computations, so you can have computations anywhere in the notebook. This is *LiveMath*'s way of just keeping everything nice and tidy.[1]

Just like any program, you can Open, Close, Save, and Print a notebook, along with other file-based commands we will explain in more detail later. These are contained in the **File** menu.

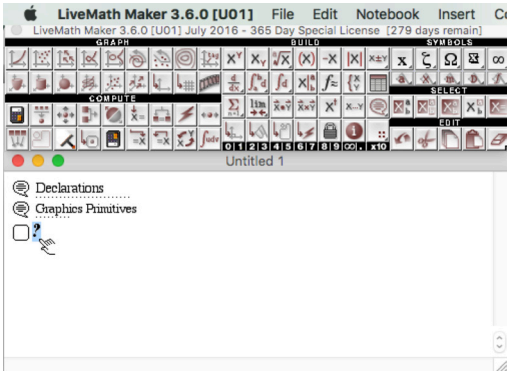


1 This structure may have originally been to avoid a patent conflict with MathCad, which apparently had a patent on "math anywhere on the page".

1.4 First Computation

Just to get started, let's just add some numbers.

In a new notebook, click on the a Math Assumption ? to place where input from the keyboard will go,



and type in:

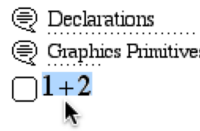
Declarations
Graphics Primitives
1+2

Now you might be inclined to hit the RETURN key and hope that something happens, but that will only create a new Assumption square to type in more input.

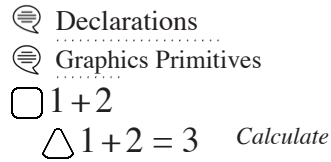
In *LiveMath*, you select an object or expression to tell *LiveMath* **WHAT** you want to act on, and then do a menu or palette (or keystroke) command **perform the action** to the object or expression.

In English grammar, you *select the noun*, and then *apply the verb* with a menu/palette/keystroke command.
So, to compute 1+2, we need to select the expression 1+2 with the

mouse. We can do the selection by double-clicking on the + sign:

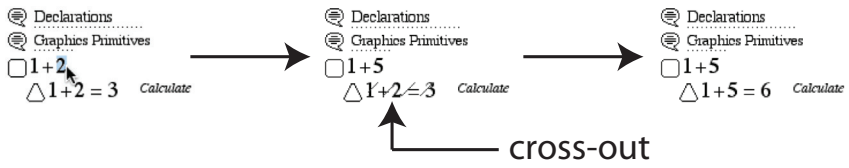


Now we have told LiveMath WHAT we want to act on, next we apply the verb, which, in this case, is the Calculate command. Click on this button in the palette:  The result is:



The triangle icon  is a Conclusion, indicating an *output* from *LiveMath*.

Now we can demonstrate the reason this software is called *LiveMath*: the "Live" in the name is the essence of the following action: select the "2" in the original equation (by double-clicking on the "2") and then type over it with a [5] :



Notice that just after the "2" was changed to a "5", the output Conclusion line from *LiveMath* was crossed-out, and it changed to the updated computation of "1+5" from the original "1+2". This is a *live recomputation* in *LiveMath*.

This live computation feature is the cornerstone of the *LiveMath* sys-

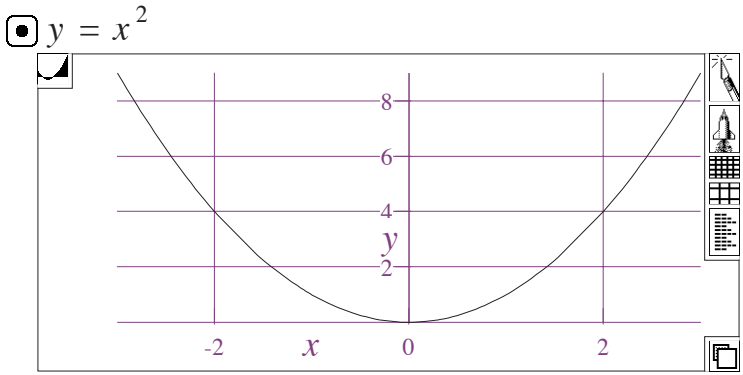
tem, and part of the reason why the software's name was changed from *Theorist* to *LiveMath*.

1.4 First 2D Graph

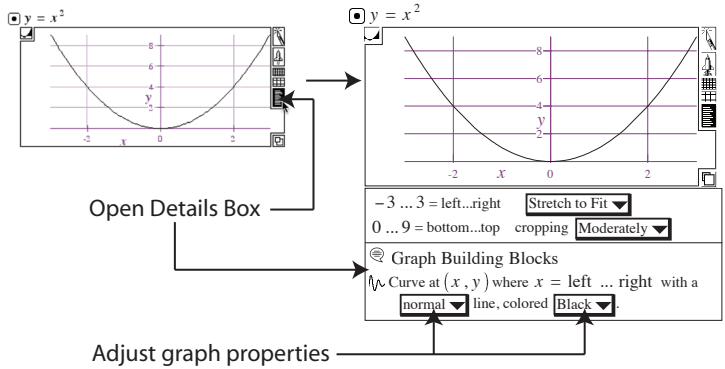
To make your first 2D graph, type in an Assumption blank ? :

Highlight this equation of 2 variables and click on this Palette button:

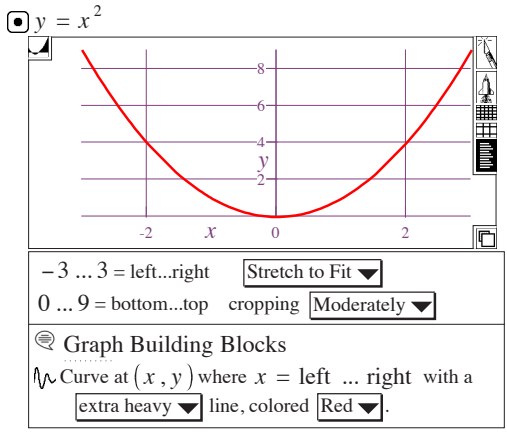
And voilà:



You may change that graphed curve to being drawn in *red*, with a **extra heavy** line, by clicking on the Details box of the graph, and editing the LineAt graphics primitive as shown:

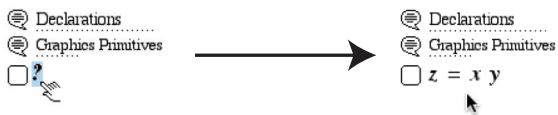


to create



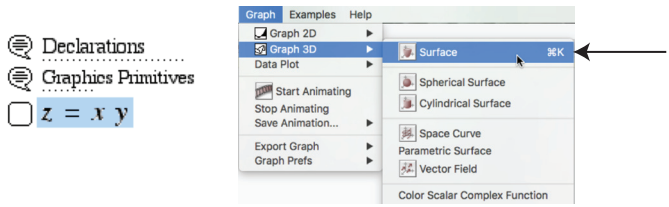
1.5 First 3D Graph

To make your first 3D graph, type in an Assumption blank ? :

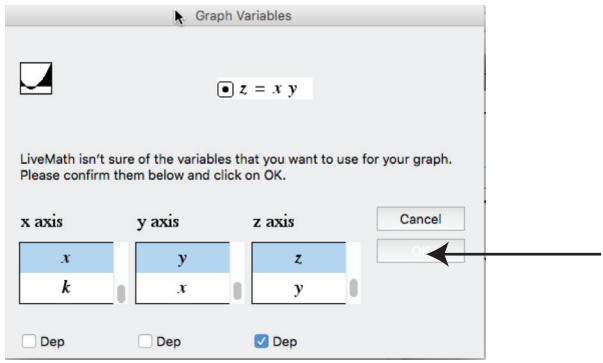


The * in between the "x" and the "y" is very important: it signifies multiplication. LiveMath does not show the multiplication, but gives a slight spacing between the "x" and "y", and keeps both variables italicized. The most common mistake it is not include the * between variables.

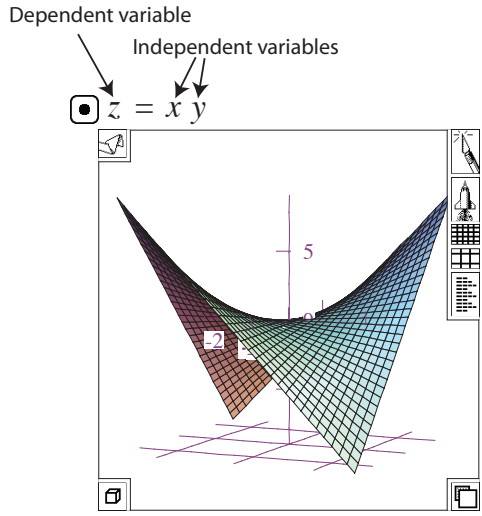
Highlight this equation of 3 variables and select this menu:



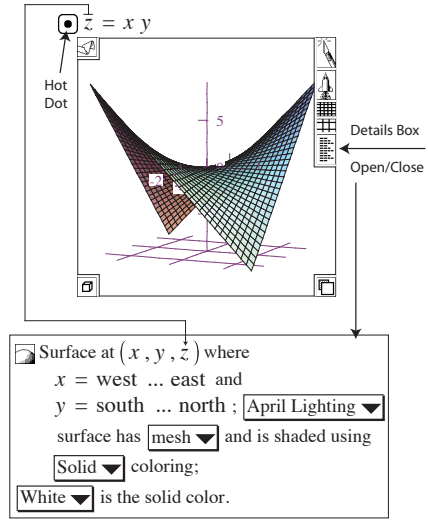
LiveMath will ask you to confirm the variable-to-coordinate axes match ups, which are easy since we used the standard *x,y,z* as variables for our equation:



Upon clicking OK, you will see your first 3D Graph created:

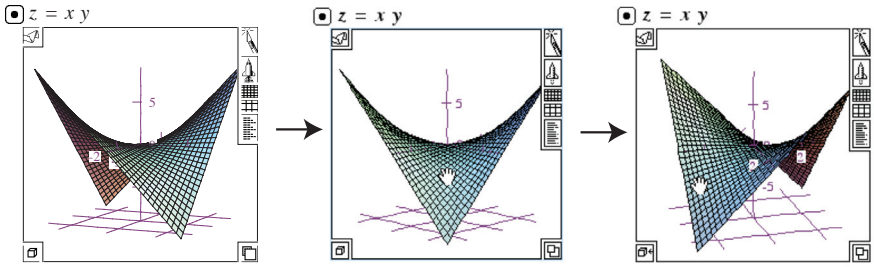


If you click on the Details Box to open up the graph, you will see that *LiveMath* created a **SurfaceAt** *graphics primitive*, and is using the $z = x\ y$ equation, and its dependent variable z - as signified by the **Hot Dot** declaring this equation as the working definition for defining the values of the dependent variable z.



Now do something we can't very describe very well in a printed book: put the mouse cursor over the surface, click down and drag to turn the

3D surface to look at it from a different point of view.



Lastly, do something we can't very describe very well in a printed book: put the mouse cursor over the surface, click down and drag and throw to start the 3D surface spinning!

1.6 Save That Notebook!

LiveMath is rather old software. It was originally created back in 1990, based upon code from earlier in 1985. That's quite an old codebase.

One of the very negative design decisions was back when was: if there is something wrong in the program - some data is not as expected, or a click in wrong place was not anticipated, or any number of such issues - the program was *designed* to crash.

Personally I disagree with this design decision. I have spent considerable time and resources over the years trying to undo this decision, and let the program *fail gracefully* rather than crashing.

However, facts are facts, and *LiveMath* crashes. So do many programs, including Microsoft Word and Excel, the difference being those programs (that have hundreds or thousands of programmers working on them) have backup plans - saving their files incrementally along the way (which actually is one of the reasons they are rather sluggish).

Our motto with *LiveMath* is: after you do something you like, SAVE.

You can try turning on **AutoSave**, which is a rudimentary "save while the program is inactive" feature, but if you are actively working on a notebook, it won't save until the program is idle for 20 seconds. So it is best to do *manual saves, frequently.*

The quick key ⌘ S or ctrl S should be your friend. Save save save save save .....

LiveMath will crash. It is only a matter of time. It is sad when a user says "I was working on that notebook for an hour, and *LiveMath* crashed, and I lost everything!" Yes, that will happen, if you don't take this advice!

1.7 Explore The Palettes

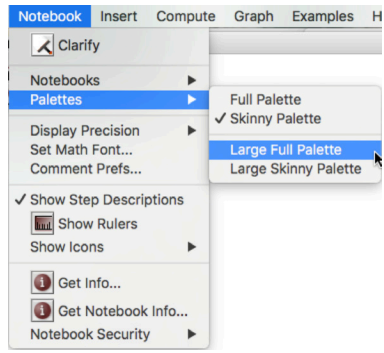
When LiveMath started as the original program Theorist, the Palette was the primary innovation of the software, in addition to its natural ease-of-use.

The palette has changed in many iterations over the years. For the lastest version, we have 2 palettes in 2 different sizes (making 4 palettes total):

Main Palette:

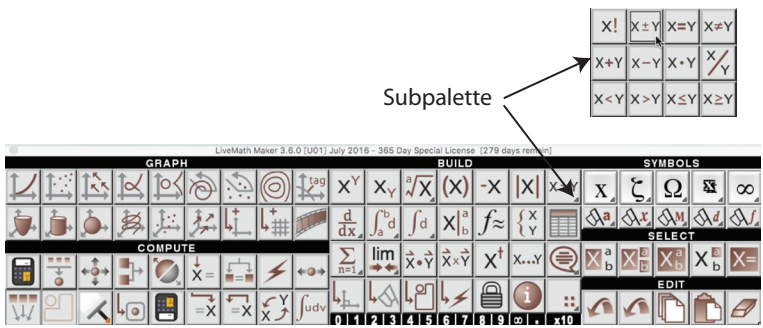
Skinny Palette:

These palettes also come in Large size: they look the same, but the larger size uses newer artwork (stolen from the next version of the LiveMath software). You may change the palettes via this menu:



If you open the Large Full Palette in a notebook, and then **Save** that notebook, the choice of palette will be remembered when you open up the notebook again.

Some of the buttons on the palette have subpalettes, which you access by click-down-and-hold over the palette button, then drag to make a selection from the subpalette. Palette buttons that have a subpalette have a wedge in the lower right hand corner of button to indicate that a subpalette is available:



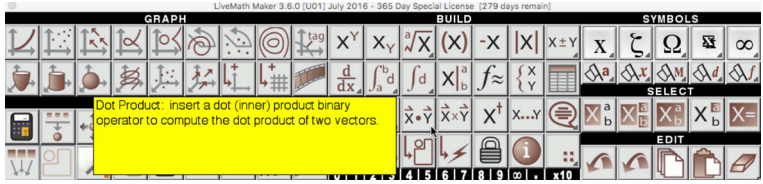
Some of the buttons are rather obvious as to what they do:

X^Y	**Superscript**
X_Y	**Subscript**

$\frac{\delta}{\delta x}$	**Partial Derivative**
$\int_a^b d$	**Definite Integral**

But other buttons will be quite mysterious indeed! What the heck is  ?? How does a window wiper fit into math software?

You will learn all of the buttons in good time. You can also hold down the ⌘ or *ctrl* and put the mouse pointer over the buttons to see an explanation of each palette button.



Some of the explanations will make sense, others will not until you learn more about the program.

1.8 Concluding Remarks

In this first chapter we have shown very very very basic examples of the main functions in *LiveMath*: computations and graphs.

LiveMath is infinitely more powerful than these initial examples, and yet *LiveMath* does not pretend nor attempt to be as powerful as some other computer algebra systems such as *Mathematica, Maple,* or *Matlab*. Each tool has its place, and if you are trying to solve system of second order partial differential equations with boundary value conditions, then these higher computer algebra systems are going to be your tool.

But if, in contrast, you want to *think* about the mathematics of your

equations and graphs, and do various computations of substitutions, factorizations, expansions, and numerical computations, then *LiveMath* might be a good complimentary tool to these higher and more powerful computer algebra systems, which often lack an ease of use or simplicity that is at *LiveMath*'s core strength.