GNU TeXmacs
Towards a scientific office suite

Massimiliano Gubinelli
CEREMADE - UMR CNRS 7534, Université Paris Dauphine
Place du Maréchal De Lattre De Tassigny, 75775 Paris cedex 16, France

Joris van der Hoeven, François Poulain, Denis Raux
Laboratoire d'informatique, UMR 7161 CNRS, École polytechnique
91128 Palaiseau Cedex, France

September 30, 2014

1 Introduction

The GNU TeXmacs project aims to provide a free, polyvalent and user-friendly scientific office suite, which can easily be interfaced with a wide range of external mathematical software. The system can be downloaded from `www.texmacs.org`. It should be noticed that TeXmacs has been developed from scratch in C++ and Scheme. In particular, the software does not rely on TeX or LaTeX.

With respect to standard office suites such as Microsoft Word or Open Office, we offer better support for mathematical typesetting, formula editing, and other features useful for scientists. With respect to TeX/LaTeX [Knu84, Lam94] and its various front-ends, TeXmacs has the advantage of being completely *wysiwyg* (what you see is what you get). Indeed, the development of our system was initially motivated by the following reasons:

- An editor should allow the author to concentrate on *what* is written and not on *how* it is written. In particular, editors should be as *wysiwyg* as possible.

- With the advent of a wide variety of mathematical software, it should be possible to make documents more active. One might wish to incorporate computer algebra sessions and spreadsheets, for instance.

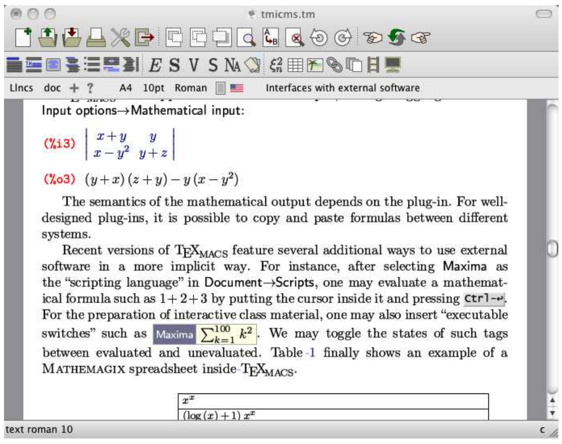
- Scientific editors should become more integrated, taking example on office suits for non-scientific users. For instance, they should provide tools for drawing technical pictures, making presentations from a laptop, annotating texts, etc.

In this paper, we will present a quick survey of the traditional features of TeXmacs and continue with more recent improvements and new features. The next major release of TeXmacs 2.1 is planned for later this year.

The core of TeXmacs consists of a free wysiwyg scientific text editor, which includes a mathematical formula editor, the possibility to write structured texts, and to extend the editor using personal style files or customizations of the user interface. Advanced typesetting algorithms are used, which allow for the creation of high quality documents.

Gradually, more and more features have been added to the software, thereby moving towards our goal to provide a fully fledged scientific office suite. TeXmacs currently offers an editor for graphical pictures, a presentation mode, a rudimentary spreadsheet facility, integrated version control, etc. In addition, TeXmacs has been interfaced to many external mathematical computation systems. These interfaces can be used either in shell like sessions, inside spreadsheets, or on the fly inside regular text.

Our main objectives for the next major version TeXmacs 2.1 are to increase the portability of the software and to further improve the user experience. For these reasons, we completely redesigned the graphical user interface (see Figure 1), which is now based on QT instead of XWINDOWS. Recent versions of TeXmacs are available under LINUX, MACOS and WINDOWS. We took special care at following standard user interface conventions for each of these operating systems (regarding keyboard shortcuts, for instance). The existing LaTeX converters were also greatly improved and we now provide a native converter to PDF. We finally improved the font support and the internationalization of TeXmacs.



Input options→Mathematical input:

(%i3) $\begin{vmatrix} x+y & y \\ x-y^2 & y+z \end{vmatrix}$

(%o3) $(y+x)(z+y)-y(x-y^2)$

The semantics of the mathematical output depends on the plug-in. For well-designed plug-ins, it is possible to copy and paste formulas between different systems.

Recent versions of TeXmacs feature several additional ways to use external software in a more implicit way. For instance, after selecting Maxima as the "scripting language" in Document→Scripts, one may evaluate a mathematical formula such as $1+2+3$ by putting the cursor inside it and pressing `Ctrl-↵`. For the preparation of interactive class material, one may also insert "executable switches" such as Maxima $\sum_{k=1}^{100} k^2$. We may toggle the states of such tags between evaluated and unevaluated. Table 1 finally shows an example of a MATHEMAGIX spreadsheet inside TeXmacs.

x^x
$(\log(x)+1)\,x^x$

Figure 1. The new graphical user interface of TeXmacs under MacOS.

2 TeXmacs as a structured text editor

The backbone of TeXmacs is a wysiwyg structured text editor. The user interface is redundant by design, so as to make the software suitable for users with diverse backgrounds. For instance, in order to create a new section, the user has the following options:

1. Use the Insert→Section→Section menu item.

2. In the second icon bar, click on the ☰ icon, followed by a click on Section.
3. As in LaTeX, type `\section` followed by ↩.
4. Use the keyboard shortcut `Alter-1`.

In a similar way as in LaTeX, authors are invited to concentrate on intent rather than presentation. Nevertheless, most tags have a sufficiently distinctive presentation for making the structure apparent from the mere rendering of the document.

All documents are internally represented and manipulated as trees. The structure of the documentation is made more visible to the user by putting non intrusive boxes around all tags which contain the cursor. The innermost tag is called the current focus and is highlighted using a special color. Various editing operations allow the user to directly operate on the structure of the document. For instance, if the current focus is a section title, then there are actions for changing it into a subsection title, to jump to the next and the previous section, to toggle the numbering, or to get contextual help on the section tag.

An analogue of the “LaTeX source code” is available in TeXmacs using Document→Source→Edit source tree. However, from our standpoint, there is no real concept of *the* “source code”. In reality, documents are trees, which can be *rendered* in different ways so as to make certain tags more or less explicit. In particular, the presentation of the “source code” can be customized using Document→Source→Preferences. Furthermore, we consider $\sqrt{x}$ to be just as good (and arguably even better) a “source code” as `\sqrt{x}`.

TeXmacs also allows the user to create new style sheets or to modify the definitions of existing presentation macros. In recent versions, this kind of customizations have been made even easier: we both provide simplified widgets for editing macros and the possibility to jump directly to the definition of the macro corresponding to the current focus.

3 Mathematical formulas

Special care has been taken so as to make the input of mathematical formulas particularly efficient. First of all, we designed a special input method which allows users to enter most mathematical symbols are obtained using a small set of basic rules:

- Characters which are naturally obtained as “superpositions” or “concatenations” of symbols on your keyboard are entered in a straightforward way. For instance, `->` yields $\rightarrow$, `<=` yields $\leqslant$, `+-` yields $\pm$ and `<<` yields $\ll$.
- The “variant” key `⇥` may be used in order to obtain variants of a given symbol or keyboard shortcut. For instance, `<=⇥` yields $\leq$, `<=⇥⇥⇥` yields $\Leftarrow$, `<⇥` yields $\prec$ and `<|⇥` yields $\lhd$. All Greek letters can be obtained as variants of the Roman ones: `A⇥` yields α, `L⇥` yields λ and so on. Sometimes, additional variants are available: `B⇥⇥` yields $\flat$ and `E⇥⇥` yields the mathematical constant e.

- The `/` and `@` keys are used for obtaining negations and symbols inside other symbols. For instance, `<=/` yields $\nleqslant$ and `@+` yields $\oplus$. More elaborated examples are `@⇥+` and `<⇥=⇥/⇥`, which yield $\boxplus$ resp. $\lneqq$.

Efficient shortcuts are also available for most mathematical constructs: `⌥F` starts a fraction, `Alter-S` a square root, `_` a subscript, `^` a superscript, etc. Being faithful to the principle of redundancy, these actions can also be performed through the menus, the icons, or *via* LaTeX equivalents.

Inside a mathematical formula, the cursor keys allow you to move around in a graphically intuitive way. In particular, when done with a particular subformula, it usually suffices to press the right arrow key `→` in order to return to the main formula.

Wysiwyg editors are especially interesting for more complex formulas. For instance, it is easy to insert new rows and columns inside a matrix, or to copy and paste submatrices (and not only rows).

Another particular feature of TeXmacs is that formulas carry more semantics than in LaTeX, when entered appropriately. For instance, "invisible" multiplication (as in $x\,y$) should be entered explicitly using `*`, whereas function application (as in $\sin x$) should be entered using `Space`.

Recent versions of TeXmacs integrate a parser for mathematical formulas and a syntax checker. When activating "semantic editing" from the icon menu, the focus box indicates the arguments of mathematical operators and its color changes to red whenever a formula is syntactically incorrect. The mathematical formula parser is based on a fixed grammar which works on a wide variety of mathematical texts. Nevertheless, the user can explicitly modify binding forces when needed and define specific notations using the standard macro mechanism. We refer to [Hoe11] for more details.

4 Interfaces with external software

TeXmacs has been interfaced with many external systems [Gro01]. In particular, we have interfaces for the computer algebra systems AXIOM, MACAULAY2, MAPLE, MATHEMAGIX, MATHEMATICA, MAXIMA, PARI, REDUCE, SAGE, etc. We also have interfaces for other mathematical software, such as OCTAVE, R, SCILAB, etc.

The traditional way to use "plug-ins" is through "shell sessions":

```
Maxima 5.9.0 http://maxima.sourceforge.net
[...]
(%i1) diff (x^x^x^x, x);
```

(%o1) $x^{x^{x^x}}\left(x^{x^x}\log(x)\left(x^x\log(x)\left(\log(x)+1\right)+x^{x-1}\right)+x^{x^x-1}\right)$

```
(%i2) integrate (%o1, x);
```

(%o2) $\mathrm{e}^{\mathrm{e}^{\mathrm{e}^{x\log(x)}\log(x)}\log(x)}$

TeXmacs also supports two-dimensional input, through toggling of Focus→Input options→Mathematical input:

(%i3) $\begin{vmatrix} x+y & y \\ x-y^2 & y+z \end{vmatrix}$

(%o3) $(y+x)\,(z+y)-y\,(x-y^2)$

The semantics of the mathematical output depends on the plug-in. For well-designed plug-ins, it is possible to copy and paste formulas between different systems.

Recent versions of TeXmacs feature several additional ways to use external software in a more implicit way. For instance, after selecting Maxima as the “scripting language” in Document→Scripts, one may evaluate a mathematical formula such as $1+2+3$ by putting the cursor inside it and pressing `Ctrl-↩`. For the preparation of interactive class material, one may also insert “executable switches” such as Maxima $\sum_{k=1}^{100} k^2$. We may toggle the states of such tags between evaluated and unevaluated. Table 1 finally shows an example of a Mathemagix spreadsheet inside TeXmacs.

x^x
$=\text{derive}(a1, x)$
$=\text{derive}(a2, x)$
$=\text{derive}(a3, x)$
$=\text{derive}(a4, x)$
$=\text{derive}(a5, x)$

$\rightsquigarrow$

x^x
$(\log(x)+1)\,x^x$
$\left(\log(x)^2+\frac{1}{x}+2\log(x)+1\right)x^x$
$\left(\log(x)^3+3\log(x)^2+3\left(\frac{1}{x}+1\right)\log(x)+\frac{3}{x}-\frac{1}{x^2}+1\right)x^x$
$\left(\log(x)^4+4\log(x)^3+6\left(\frac{1}{x}+1\right)\log(x)^2+\left(\frac{12}{x}-\frac{4}{x^2}+4\right)\log(x)+1+\frac{6}{x}-\frac{1}{x^2}+\frac{2}{x^3}\right)x^x$
$\left(\log(x)^5+5\log(x)^4+10\left(\frac{1}{x}+1\right)\log(x)^3+\left(\frac{30}{x}-\frac{10}{x^2}+10\right)\log(x)^2+\left(5+\frac{30}{x}-\frac{5}{x^2}+\frac{10}{x^3}\right)\log(x)+1+\frac{10}{x}+\frac{5}{x^2}-\frac{6}{x^4}\right)x^x$

Table 1. Computation of successive derivatives in a spreadsheet.

5 Towards a scientific office suite

We have seen that TeXmacs integrates a structured text editor, a formula editor, a spreadsheet facility and many interfaces to external programs. Let us describe a few other tools that are available nowadays inside our system, which make TeXmacs a fairly complete scientific office suite.

5.1 Presentation mode

A wysiwyg editor such as TeXmacs is particularly useful for preparing laptop presentations. For this, it suffices to select beamer as the document style in the Document→Style menu. In addition, several standard themes can be used. Presentations are organized as successions of “screens”.

Special markup is provided for showing and hiding content in specified orders. For instance, the “unroll” tag allows item lists to be unrolled progressively. There is also support for general “overlays”, where the user has full control over the order in which content appears and disappears on specified ranges of overlays.

The Insert→Animation menu allows for the insertion of animated content. For the moment, only simple animations are implemented, but more elaborate graphical effects and artwork are planned for future versions, as well as support for embedded videos.

5.2 Technical pictures

Existing pictures can be embedded inside a document using Insert→Image→Insert image or Insert→Image→Link image. In addition, TeXmacs includes a native editor for drawing simple technical pictures. One advantage of this integrated drawing tool is that it is easy to include mathematical formulas or other TeXmacs markup inside the picture. One may use Insert→Image→Draw image to start a new drawing and Insert→Image→Draw over selection to draw a picture on top of the current selection (typically an external picture, or a mathematical formula). Currently, TeXmacs implements the most basic primitives for drawing vector graphics. We have plans to extend the TeXmacs macro mechanism to graphics and to offer full support for the SVG standard.

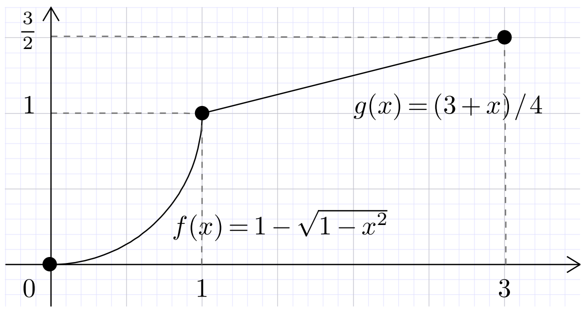


Figure 2. Toy example of a technical picture created with the drawing tool.

5.3 Version control

TeXmacs comes with an efficient tool for computing and visualizing "structured differences" between two versions of a document. The default way to visualize changes is to show the old and new versions side by side in different colors. Authors may quickly go trough the changes introduced by a coauthor and select which versions they prefer. TeXmacs also integrates support for external versioning software. For the moment, we only provide an interface for Svn, but it would be easy to add interfaces for other systems.

6 Compatibility with other formats

Unfortunately, TeXmacs is not yet as wide-spread as LaTeX. For the sake of backward compatibility, TeXmacs provides high quality converters from and to LaTeX. However, these converters cannot be perfect for several reasons.

The main reason is that LaTeX is *not* a format, like Html, but rather a programming language. In particular, the only program which parses all LaTeX files correctly is LaTeX itself. Since TeXmacs is *not* a LaTeX front-end, it follows that we can only ensure correct conversions for a (quite large) sublanguage of LaTeX.

The other main reason is that TeXmacs has a more powerful typesetting engine than LaTeX and that it provides several extensions (like a graphical editor or animations) which are not available in LaTeX. Therefore, a conversion to LaTeX may downgrade your document, both in typesetting quality and in structure. For instance, TeXmacs pictures are exported as postscript images, so their structure is lost.

Nevertheless, during recent years, we have invested a lot of energy in making the LaTeX converters as good as possible, in both directions. In particular, we support the most frequently used LaTeX styles. The behaviour of the converters can be fine-tuned for specific needs *via* the user preferences. For instance, should macros be expanded or not during conversions? Are preambles allowed to contain additional macro definitions? Etc.

An interesting recent addition is the possibility to use the converters in a "conservative" fashion [HP14]. For instance, assume that Alice writes a document in LaTeX and sends it to Bob. Bob opens the document in TeXmacs, makes a few modifications and exports the document back to LaTeX. Conservative converters have the property that the exported document will be almost the same as the Alice's original version: ideally speaking, only Bob's changes will really be exported.

Besides LaTeX, reasonably good converters for Html and MathML have also been implemented, again in both directions. For instance, the TeXmacs website is entirely generated from TeXmacs documents.

Last but not least, TeXmacs is wysiwyg, which means that TeXmacs traditionally features a lossless converter to Postscript. More recently, a native converter to Pdf has also been implemented. This new converter includes an improved support for images, fonts and certain types of graphics. In particular, the quality of Pdf documents with oriental languages is much better nowadays.

7 Customization

TeXmacs can be customized in many ways. Besides the possibility to write your own style files for the presentation of documents, it is also possible to customize the behaviour of the editor, or to write new plug-ins for external software. We will briefly give a few examples below. Questions can be asked on our mailing lists:

http://www.texmacs.org/tmweb/home/ml.en.html

7.1 User-defined macros

The user may define new typesetting constructs or customize the rendering of the standard styles using a special macro language. For instance, one may define a macro `cd` for square commutative diagrams using

$$\langle\mathsf{assign}|\mathit{cd}|\langle\mathsf{macro}|A|B|C|D|\begin{array}{ccc}A & \rightarrow & B\\ \downarrow & & \downarrow\\ C & \rightarrow & D\end{array}\rangle\rangle$$

This macro may then be used by typing `\CD↩`, as in

$$\begin{tikzcd} A\oplus B \arrow[r] \arrow[d] & C \arrow[d] \\ D \arrow[r] & E\otimes F \end{tikzcd}$$

For more information, we refer to Help→Manual→Writing your own style files and Help→Reference guide.

7.2 Customizing the interface and Scheme extensions

Following the example of GNU Emacs, the user interface and most of the editing functions of TeXmacs are written in Scheme, a high level “extension language”. This makes it possible for the user to customize the behaviour of TeXmacs and write extensions to the editor. Simple customizations can be put in the file `my-init-texmacs.scm` of the `.TeXmacs/progs` subdirectory of your home directory. For instance, assume that this file contains the following code:

```
(kbd-map
  (:mode in-text?)
  ("T h ." (make 'theorem))
  ("D e f ." (make 'definition)))
```

Then the keyboard shortcuts `⇧TH.` and `⇧DEF.` can be used inside text mode in order to insert a theorem resp. a definition. In a similar way, you may customize the menus, or add more complex extensions to the editor. For more details, we refer to Help→Scheme extensions.

Bibliography

[Gro01] A. G. Grozin. TeXmacs interfaces to Maxima, MuPAD and Reduce. In V. P. Gerdt, editor, *Proc. Int. Workshop Computer algebra and its application to physics*, number 11-2001-279 in JINR E5, page 149, Dubna, June 2001. Arxiv cs.SC/0107036.

[Hoe11] J. van der Hoeven. Towards semantic mathematical editing. Technical report, HAL, 2011. http://hal.archives-ouvertes.fr/hal-00569351, submitted to JSC.

[HP14] J. van der Hoeven and F. Poulain. Conservative conversion between LaTeX and TeXmacs. Technical report, HAL, 2014. http://hal.archives-ouvertes.fr/hal-00952926.

[Knu84] D.E. Knuth. *The TeXbook*. Addison Wesley, 1984.

[Lam94] L. Lamport. *LaTeX, a document preparation system*. Addison Wesley, 1994.