Plotting functions with TEXMACS graphics and Scheme

Tests the version of the plotting software which loads the function definitions from a file and uses sandboxed evaluation of the file contents; please adjust the path of the file arguments according to where you put them.

Table of contents

1 Syntax

The `plotFun` command takes one arguments, the name of a file. The file contains the functions to be plotted, the range for each function, general plot options and additional options for each function (color, style, thickness, number of points); default options are substituted for missing options.

The functions and their ranges are expressed as a list of association lists (one association list for each function)

The function is expressed as a lambda in Scheme syntax (associated to the key `"function"`)

example: `(lambda (x) ( - (expt x 2) 2.))`

the range as a list (associated to the key `"range"`)

example: `(-2.  2.)`

Please refer to the example input files `fundefs_07.scm`, `fundefs_08.scm` and `fundefs_09.scm` in the `doc` directory.

2 Test files

A test file (`‘plotting_with_Scheme_plugin_examples.tm‘`) together with its input files is in the `/doc` directory. Its .pdf output (`‘plotting_with_Scheme_plugin_examples.pdf‘`), as well as the .pdf output of this document, is both in the `/doc` and in the top directory.

3 Input

In the input file an association list (and only that) must be present.

The association list contains the following keys

- `"xLabel"` (string)
  - test

- `"yLabel"` (string)
- `"title"` (string)
- `"sizeX"` (double) [optional, default 9]
- `"sizeY"` (double) [optional, default 6]
- `"plotsList"` (list of association lists, see below)

The value of the entry with the key "plotsList" is the list of plots, which is a list of association lists. Each element of the list corresponds to one function; in each element (association list) there are the following keys:

- `"function"` (a lambda expression)
- `"range"` (a list of two doubles)
- `"nPoints"` (positive integer) [optional, default 101]
- `"color"` (either a name or an HTML code) [optional, defaults to the color list]
- `"line-width"` (string containing a number and units) [optional, default `"1.5ln"`]
- `"dash-style"` (string containing ones and zeros) [optional, default `"11111"`]

Please see test input files (in directory `plotFun/doc`) for examples.

Remark. The input syntax (association list) is extremely finicky, an error makes the program fail and the error messages do not indicate that the input needs to be corrected. It is necessary to improve this part.

Possibly I will either change the syntax of the input file so that it is easier to type or will write a parser that will notify the user about incorrect syntax.

4 Security

I have added the `:secure` keyword to the Scheme functions, so in order to run this macro in TeXmacs it is not anymore necessary to set the security level to "Accept all scripts"; the security level can be left to "Accept no scripts".

Unsafe scripts could delete all the data on your hard drive. Since this TeXmacs package executes commands in a user-defined arbitrary file (the one specified in the argument to `plotFun`), it is quite dangerous: you have to ensure that the file you specify as argument contains only safe code. I have tried to mitigate the danger by defining a "safe module" with "safe commands" (see this message in the Guile mailing list). I do not know how effective the technique is, so **please be aware of what you put in the file you use as a macro argument**.

The technique is as far as I understand the one suited to the Guile 1 series - for the Guile 2/3 series (future $\text{TE}\text{X}_{\text{MACS}}$) I have seen that another technique exists, based on the `(ice-9 sandbox)` module; see this message in the Guile mailing list.

I have tried to make this program safe: this said, **one is responsible for making sure that his TeXmacs documents are running only safe scripts!**

5 Desirable features

The features marked with (present) are yet to be refined (result, implementation or both)

- Optional user-selected
  - graph size (present)
    - This probably needs improvement in the placement of numbers and axis labels with shifts and not multiplications
  - Axes labels (present) (not yet optional, has to be supplied)
  - Graph title (present) (not yet optional, has to be supplied)
- Optional user-selected for each plot
  - color (present)
  - line thickness (present)
  - line type (present)
  - number of points (present)
- User-selectable color lists
- Exponential notation for large/small numbers
- Error messages for wrong input
  - Optional comments in input file (to strip away line by line)
- Functions in standard notation
  - postpone

6 Examples

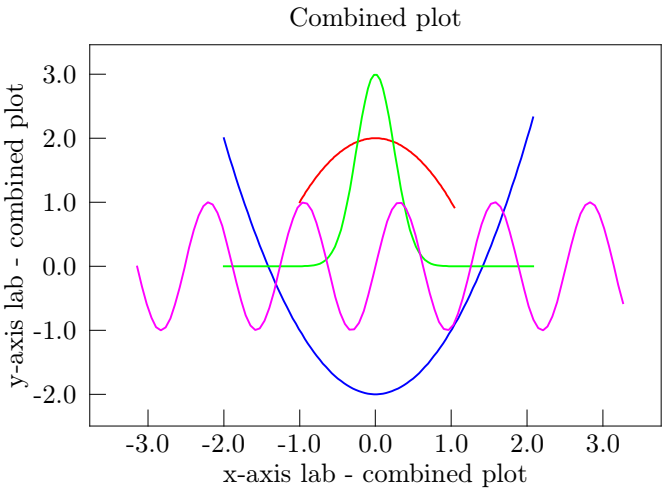

Figure 1. Two parabolae, a Gaussian and a sinusoid. Line style, color and width have default values, and the size of the plot has the default value too.

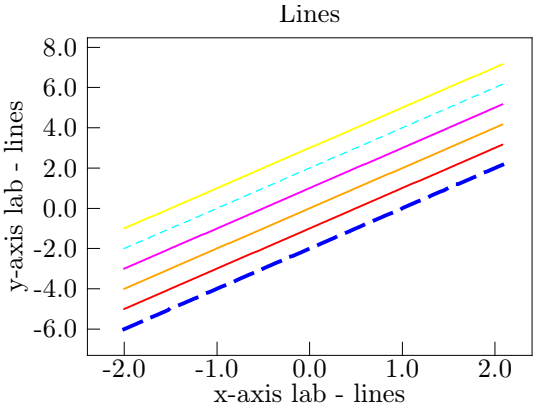

Figure 2. A series of straight lines. For some of them we choose style, color and width. The size of the plot is set by the user.

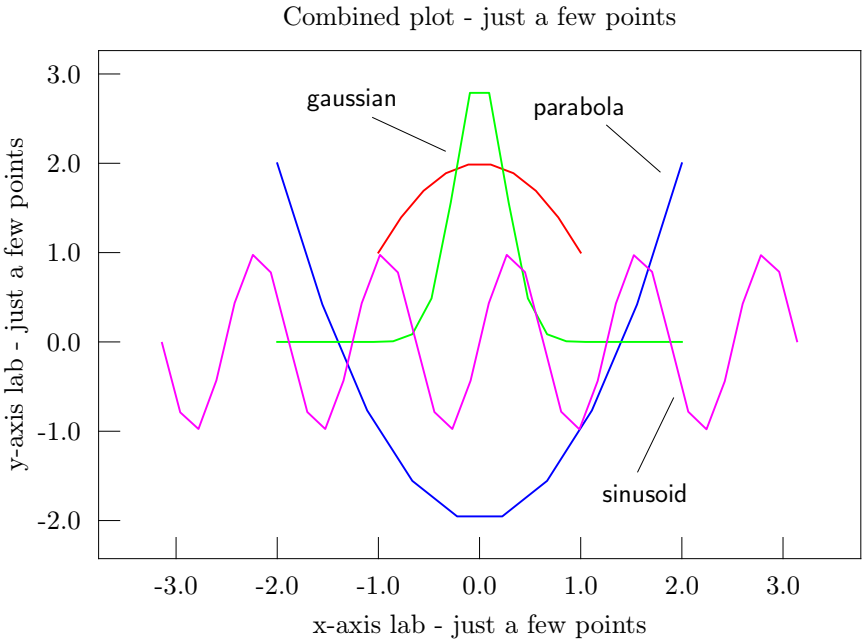

Figure 3. The same functions of figure 1 plotted with less points (default is 101) in a larger size. Annotations are inked over (`Insert` → `Image` → `Ink here`)