

Using **context** and **tikz** terminal for gnuplot

Mojca Miklavec, February 10, 2011

1	Requirements	1
2	Installation	1
3	Simple examples	2
4	Terminal options	6
5	High-level configuration from ConT _E Xt	6
6	Comparison of supported terminals	6

1 Requirements

1. Any working ConT_EXt installation (ConT_EXt Minimals, T_EX Live 2010 or MikT_EX 2.9 or newer).
2. The gnuplot binary (or gnuplot.exe under Windows) has to be in PATH and needs to have support for context and/or tikz terminal built in.
3. The latest version of gnuplot module for ConT_EXt.

2 Installation

2.1 ConT_EXt

TODO

2.2 Gnuplot

TODO

You can check the list of supported terminals by typing

```
gnuplot> set term
```

into gnuplot shell.

2.3 t-gnuplot module for ConT_EXt

Under ConT_EXt Minimals you can install the gnuplot module and TikZ with an additional switch when running `first-setup`, for example:

```
first-setup.sh --extras=t-gnuplot,t-tikz
```

If you have installed a complete or context scheme under T_EX Live, gnuplot module and TikZ might already be installed. Else you can use:

```
tlmgr install context-gnuplot
tlmgr install pgf
```

Under MikT_EX the module is installed automatically when it is first used.

3 Simple examples

3.1 Calling gnuplot directly

Let's first create a simple file (we will call it *example.plt*, but you may choose any name) with the contents below.

For context terminal:

```
set term context size 5in,3in standalone
set output "fullpage-example.tex"
plot sin(x)
plot cos(atan(x))*sin(x)
```

For tikz terminal:

```
set term tikz context size 5in,3in standalone createstyle
set output "fullpage-example.tex"
plot sin(x)
plot cos(atan(x))*sin(x)
```

In both cases the option `standalone` is used to create a complete ConT_EXt document with one plot per page, including header and `\starttext ... \stoptext`, so that it can be compiled directly. The option `createstyle` is used to create three files with required macros in working directory¹

Both terminals should give you almost equivalent results apart from default plot size. You are highly encouraged to specify the desired plot size explicitly. You may scale the plot later on, but you probably want to get the desired proportions from the start.

Run gnuplot with

```
gnuplot example.plt
```

and compile the result with any of the following three commands (depending on your preferred engine):

```
context fullpage-example.tex          # for LuaTEX
texexec fullpage-example.tex          # for pdfTEX
texexec --xT fullpage-example.tex      # for XTTEX
```

They are almost equivalent except that X_TT_EX lacks some advanced features (some patterns). The only major difference is the choice of fonts. If you want to typeset Arabic labels or use system fonts, you will probably want to choose LuaT_EX or X_TT_EX. If you are using many graphical elements (like in 3D plots), you might want to go for LuaT_EX.

You should get a PDF document with two full-page plots that you can include into your document with `\externalfigure[fullpage-example][page=2]` for example.

¹ An alternative is to place those three files somewhere where `kpathsea` can find them and omit the option `createstyle`, just make sure that the versions of `tikz` terminal and the files in your T_EX tree remain compatible.

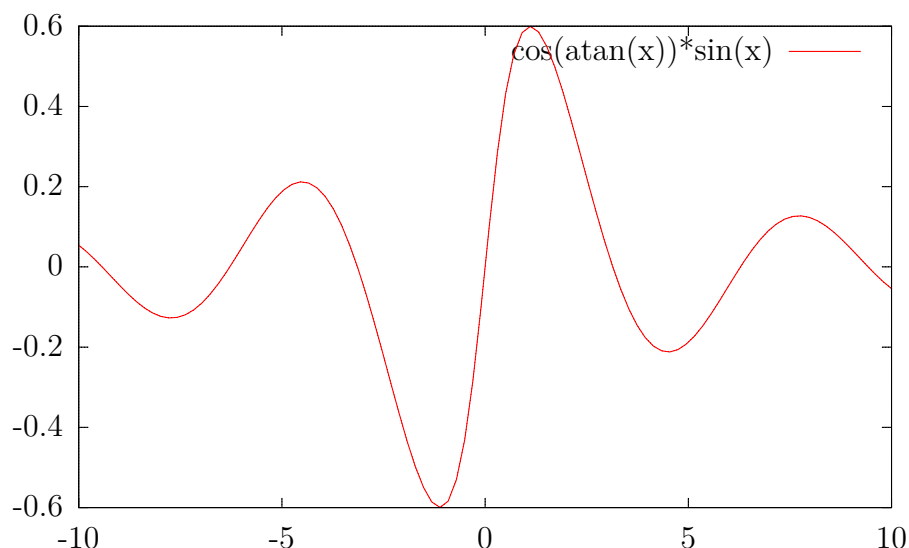


Figure 1

3.2 Calling gnuplot from T_EX

As you can see you will always get Latin Modern font at 12pt unless you explicitly change it with header `"\setupbodyfont[...]"`. An easier way to make sure that the same font is used and to avoid having to call gnuplot manually is to simply type the gnuplot code inside your ConT_EXt document:

`\usemodule`

`[gnuplot]`

`\setupGNUPLOTterminal`

`[context]`

`[width=5in,height=2.5in,textscale=0.8]`

`\setupGNUPLOTterminal`

`[tikz]`

`[width=5in,height=2.5in]`

`\starttext`

`\startGNUPLOTscript[myfunction]`

`set samples 400`

`set key left Left reverse`

`set format y "%.1f"`

`plot sin(x) t '$\sin(x)$' lw 3`

`plot cos(atan(x))*sin(x) t '$\cos(\arctan(x))\sin(x)$' lw 3 lc 3`

```
\stopGNUPLOTscript
```

```
\placefigure{none}{\useGNUPLOTgraphic[myfunction][2]}
```

```
\setupGNUPLOT
```

```
[terminal=tikz]
```

```
\placefigure{none}{\useGNUPLOTgraphic[myfunction][1]}
```

```
\stoptext
```

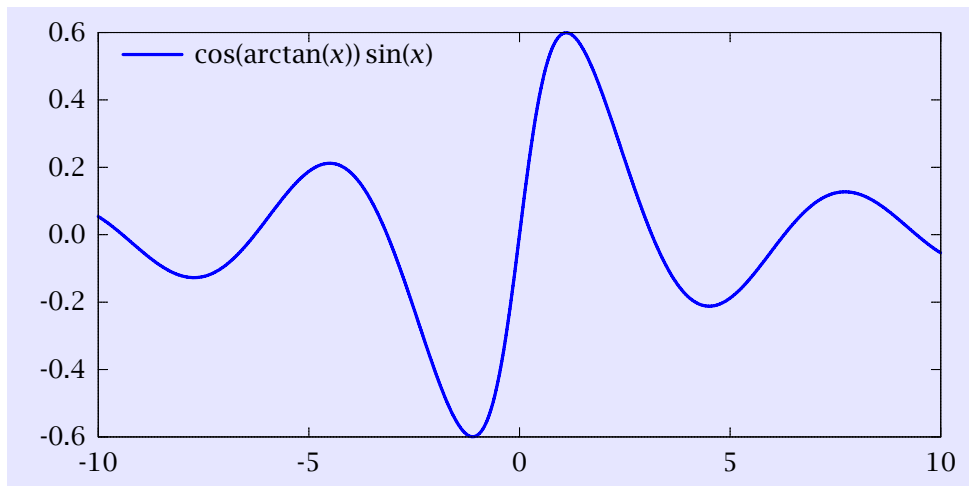


Figure 2 Framed second plot using context terminal

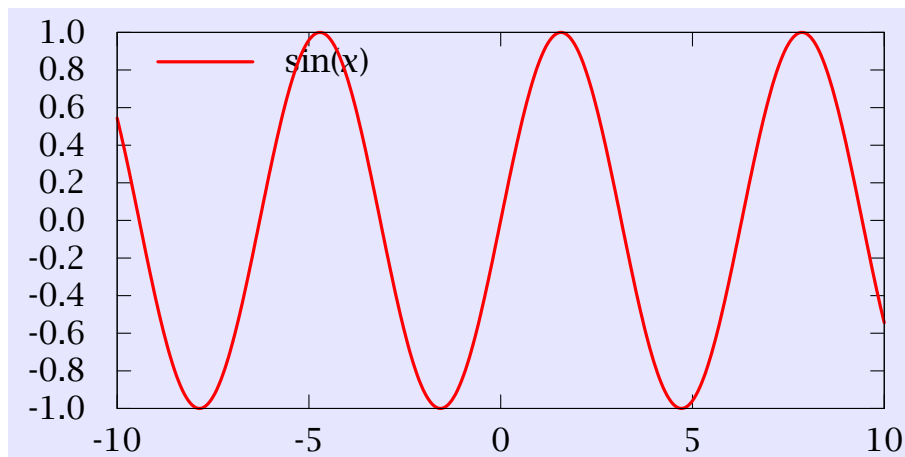


Figure 3 Framed first plot using tikz terminal

3.3 Including pre-generated plots

TODO

This is the syntax:

```
\processGNUPLOTfile[name][filename.tex]  
\useGNUPLOTgraphic[name]
```

4 Terminal options

4.1 context

TODO

```
set term context {default}  
    {defaultsize | size <scale> |  
    size <xsize> {in|cm}, <ysize> {in|cm}}  
    {input | standalone}  
    {noheader | header "<header>"}  
    {color | colour | monochrome}  
    {mitered | rounded | beveled}  
    {butt | round | squared}  
    {dashed | solid}  
    {dashlength | dl <DL>}  
    {linewidth | lw <LW>}  
    {textscale <textscale>}  
    {pointswithmetapost | pointswithmp | pointswithtex}  
    {defaultfont | font {<fontsize>} |  
    font "<fontname>{,<fontsize>}" {fontsize}}
```

4.2 tikz

TODO

5 High-level configuration from ConT_EXt

TODO

6 Comparison of supported terminals

The gnuplot module for ConT_EXt supports the following terminals:

- **bitmap terminals**
 - png, pngcairo
- **vector terminals**
 - **context, tikz**
 - metapost, postscript, pdf, pdfcairo