Using `context` and `tikz` terminals for gnuplot in ConTeXt

Mojca Miklavec, September 3, 2011

1 Requirements 1
2 Installation 1
3 Simple examples 3
4 Terminal options 8
5 Comparison of supported terminals 10

1 Requirements

1. Any working ConTeXt installation (ConTeXt Minimals, TeX Live 2011, MiKTeX 2.9 or newer -- if it starts supporting ConTeXt again).
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 ConTeXt.
4. For running GNUPLOT on the fly or when using ConTeXt MkII, you need to have `write18` enabled. Usually this can be set with `shell_escape = t` in `texmf.cnf`.

2 Installation

2.1 Gnuplot

At the time of writing gnuplot contains a buggy `tikz` terminal (it doesn't work with ConTeXt) and no `context` terminal. This may change in future, but at the moment you have to compile gnuplot yourself. The latest version of `context.trm` can be found at **https://github.com/mojca/gnuplot**.

You need to run

```
git clone git://github.com/mojca/gnuplot.git
cd gnuplot
```

```
./prepare
./configure
make
make install
```

You can also provide something like `--prefix=$PWD/install` to install gnuplot locally. Just make sure that you add the resulting binary to `PATH`.

Once you have the gnuplot binary running, you can check the list of supported terminals by typing

```
gnuplot> set term
```

into GNUPLOT shell. Make sure that it lists:

```
    context  ConTeXt with MetaFun (for PDF documents)
       tikz  Lua PGF/TikZ terminal for TeX and friends
```

The module also supports some other terminals like `png`, `metapost`, `postscript` and `pdf`, so it is usable even if you didn't compile GNUPLOT yourself, but their integration with ConTEXt is very limited.

2.2 `t-gnuplot` module for ConTEXt

Under ConTEXt Minimals you can install the gnuplot module and Tik*Z* 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 TEX Live, gnuplot module and Tik*Z* might already be installed. Else you can use:

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

Under MiKTEX 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 ConTEXt 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 directory[1].

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.

[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 TEX tree remain compatible.

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 --xtx fullpage-example.tex          # for XeTeX
```

They are almost equivalent except that XeTeX 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 LuaTeX or XeTeX. If you are using many graphical elements (3D plots, images, ...), you might want to go for LuaTeX.

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.

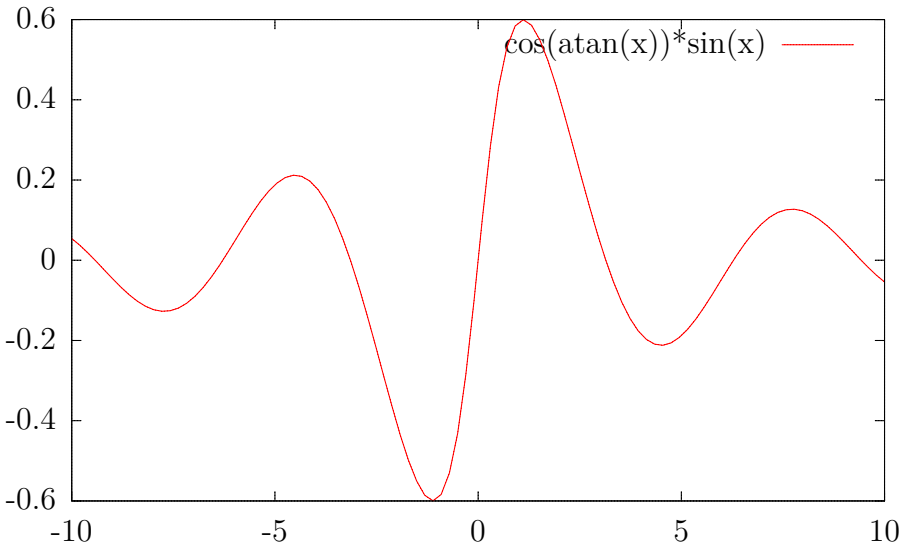

Figure 1 Second page from `fullpage-example`, included with `\externalfigure`

3.2 Calling gnuplot from TEX

As you can see you will always get Latin Modern font at 12pt unless you explicitely change it with `header "\setupbodyfont[somefontname,10pt]"` or with `font "somefontname,10pt"`. 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 ConTEXt document:

```
\usemodule
       [gnuplot]
\setupGNUPLOTterminal
       [context]
       [width=5in,height=2.5in,fontscale=0.9]
\setupGNUPLOTterminal
       [tikz]
       [width=5in,height=2.5in,fontscale=0.9]
\starttext

\startGNUPLOTinclusions
set samples 400
set key left Left reverse
\stopGNUPLOTinclusion

\startGNUPLOTscript[myfunction]
set zeroaxis
set format y "%.1f"
plot [-4:2][0:2] 1 t '' lt 0, exp(x) t '$e^x$' lt 1 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
```

With `\setupGNUPLOT[terminal=<termname>]` you can select any supported gnuplot terminal before drawing a plot.

With `\setupGNUPLOT[<termname>][<option>=<value>]` you can set some terminal-specific options.

Anything inside `\startGNUPLOTinclusions ... \stopGNUPLOTinclusion` will be applied to every plot.

The command `\startGNUPLOTscript[<name>]` creates new plots that can be included with `\useGNUPLOTgraphic[<name>][<number>][<option>=<value>]`. Both the number of plot and additional parameters (like `width=.7\textwidth` for example) are optional.

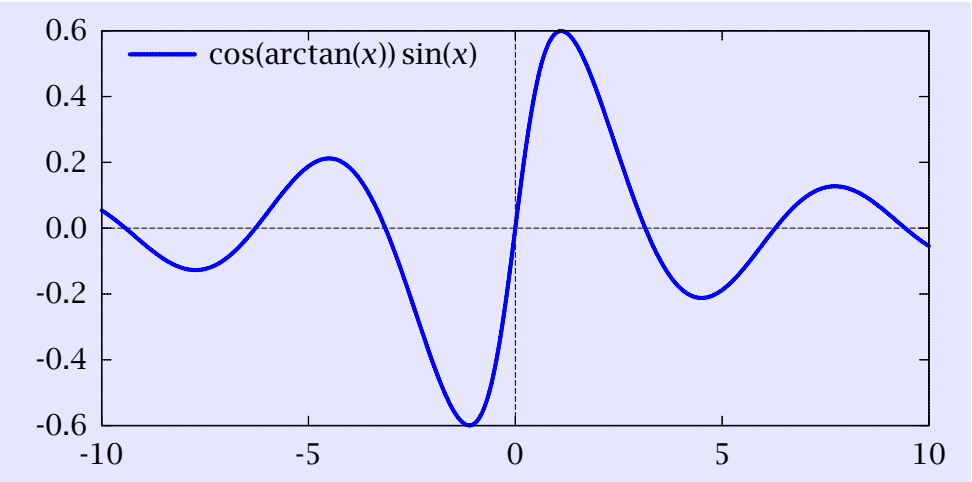

Figure 2 Framed second plot using `context` terminal

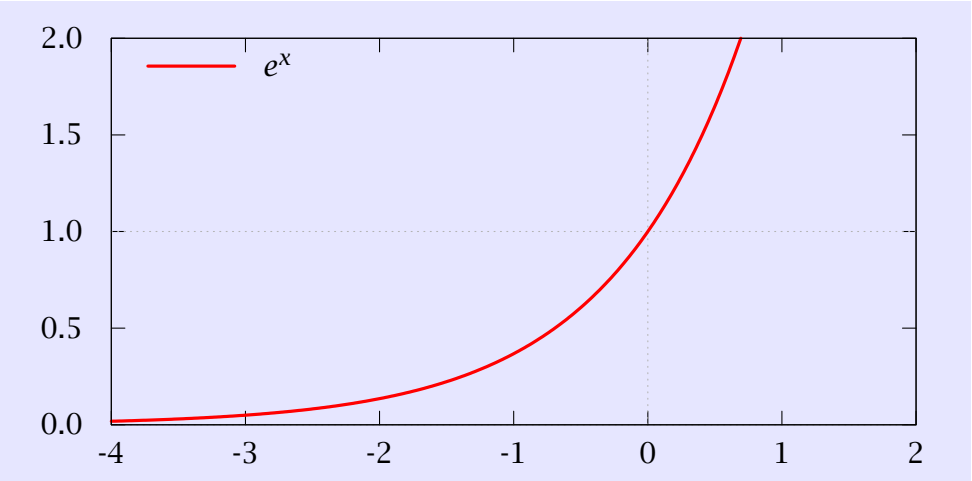

Figure 3 Framed first plot using `tikz` terminal

3.3 Including pre-generated plots

Instead of defining `\startGNUPLOTscript` and letting ConTEXt call GNUPLOT on the fly, you can also run GNUPLOT in advance and only include the resulting `filename.tex`. This is something that you might want to do when running calculation-intensive GNUPLOT scripts which take a long time.

You can follow the same steps as in section 3.1, except that you should not specify the `standalone` flag (and you should not compile the plot, only the main document).

The resulting file can be included[2] with

```
\processGNUPLOTfile[<name>][<filename.tex>]
```

and you can get the graphic with the same command as usual:

```
\useGNUPLOTgraphic[<name>]
```

plus any optional parameters.

[2] `\include filename.tex` won't work

4 Terminal options

4.1 context

```
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> }
                 { fontscale <fontscale> }
                 { mppoints | texpoints }
                 { defaultfont | font {<fontsize>} |
                   font "<fontname>{,<fontsize>}" {fontsize} }
```

4.2 tikz

```
set term tikz { latex | tex | context }
              { size <x>{unit},<y>{unit} }
              { scale <x>,<y> }
              { nofulldoc | nostandalone | fulldoc | standalone
}
              { color | monochrome }
              { dashed | solid }
              { nooriginreset | originreset }
              { nogparrows | gparrows }
              { nogppoints | gppoints }
              { picenvironment | nopicenvironment }
              { noclip | clip }
              { notightboundingbox | tightboundingbox }
              { background "<colorpec>" }
              { plotsize <x>{unit},<y>{unit} }
              { charsize <x>{unit},<y>{unit} }
              { font "<fontdesc>" }
              { fontscale <fontscale> }
              { {preamble | header} "<preamble_string>" }
              { tikzplot <ltn>,... }
              { notikzarrows | tikzarrows }
              { rgbimages | cmykimages }
              { noexternalimages|externalimages }
              { bitmap | nobitmap }
              { providevars <var name>,... }
              { createstyle }
              { help }
```

5 Comparison of supported terminals

The gnuplot module for ConTEXt supports the following terminals:

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

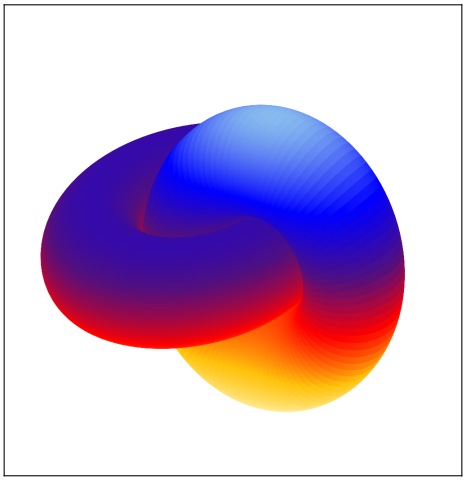

Figure 4 An example of using png terminal