The gnuplottex package*

Lars Kotthoff†, Udo Höfel‡and more contributors

December 8, 2015

1 Introduction

This package allows you to include gnuplot graphs in your LaTeX documents.

The gnuplot code is extracted from the document and written to `.gnuplot` files. Then, if shell escape is used, the graph files are automatically processed to graphics or LaTeX code files which will then be included in the document. If shell escape isn't used, the user will have to manually convert the files by running gnuplot on the extracted `.gnuplot` files.

Shell escape is available in the web2c TeX compiler, it allows the execution of shell code during the compilation of a TeX document. It's disabled by default, you'll have to edit your configuration files or give the `-shell-escape` option to `latex`.

The package also allows you to include gnuplot code in a file verbatim, generating and including the plot automatically.

2 Requirements

To use gnuplottex, you'll need the `graphicx`, `latexsym`, `keyval`, `ifthen`, and `moreverb` packages and, of course, gnuplot. If you want to use tikz-terminal you also have to use the `gnuplot-lua-tikz`-package from gnuplot. If you want to use the `eepic`-terminal, you'll need the `eepic`-, the `epic`- and (if you're not using dvips or dvipdfm) the `eepicemu`-package.

3 Usage

To load the package, simply write `\usepackage{gnuplottex}` in your document preamble. Options that can be passed to the package are

*This document corresponds to gnuplottex v0.9, dated 2015/12/08.

†`lars@larsko.org`

‡`udo.hoefel@pi1.physik.uni-stuttgart.de`

[⟨***shell***⟩] Use shell escape to automatically generate the graphs from the gnuplot source files. This is the default. Normally, you don't need to specify this option.

[⟨***noshell***⟩] Don't use shell escape, graphs must be generated manually.

[⟨***miktex***⟩] We're using mikTeX.

[⟨***siunitx***⟩] Use `siunitx` to typeset numbers in the graphs. You need to load the `siunitx` package before `gnuplottex` for this to work. If the gnuplot terminal does not support TEX, a warning will be given and the functionality not used.

[⟨***subfolder***⟩] Put the generated graphs in a "gnuplottex" subfolder, which will be created automatically.

[⟨***cleanup***⟩] Delete the `.gnuplot` files after conversion.

In addition, you can set `\gnuplotexe` to the path to the gnuplot executable. Normally, you don't need to do this; use this option only if LATEX doesn't find gnuplot.

The following environment can be used to include graphs:

`gnuplot` Within this environment, you can specify arbitrary gnuplot code, for example `plot sin(x)`.
The code necessary to write the plot to a file will be inserted by this package. It adds 'set terminal ⟨*terminal*⟩' and the name of the output file. The terminal can be specified by the user and defaults to `latex`. It may be set to anything supported by gnuplot. If set to a terminal which produces TEX output, such as latex, tex, epslatex, or pstricks, the file processed by gnuplot will be included with the `\include` command, else the `\includegraphics` command is used. The file extension of the intermediate file is in some cases different from the terminal name, this is taken care of for most common terminals in the package code. If graphics inclusion fails for a specific terminal, the intermediate file extension may be the cause.

The terminal name can be specified as a value to the key `terminal` as an argument to the environment,
`\begin{gnuplot}`[terminal=⟨*terminal*⟩]
...
`\end{gnuplot}`

The graph can be scaled by providing an argument to the `scale` key, similar to the specification of the terminal name. It defaults to 1, i.e. no scaling will be done. Additional options to the terminal can be given as argument to the `terminaloptions` key, e.g.
`\begin{gnuplot}`[terminal=pdf,terminaloptions=font ",10" linewidth 3]
...
`\end{gnuplot}`

`\gnuplotloadfile` In addition to the environment, you can use the command `\gnuplotloadfile` to directly include gnuplot source code. It accepts the same options as the environment, e.g.
`\gnuplotloadfile[terminal=pdf]{example.gnuplot}`

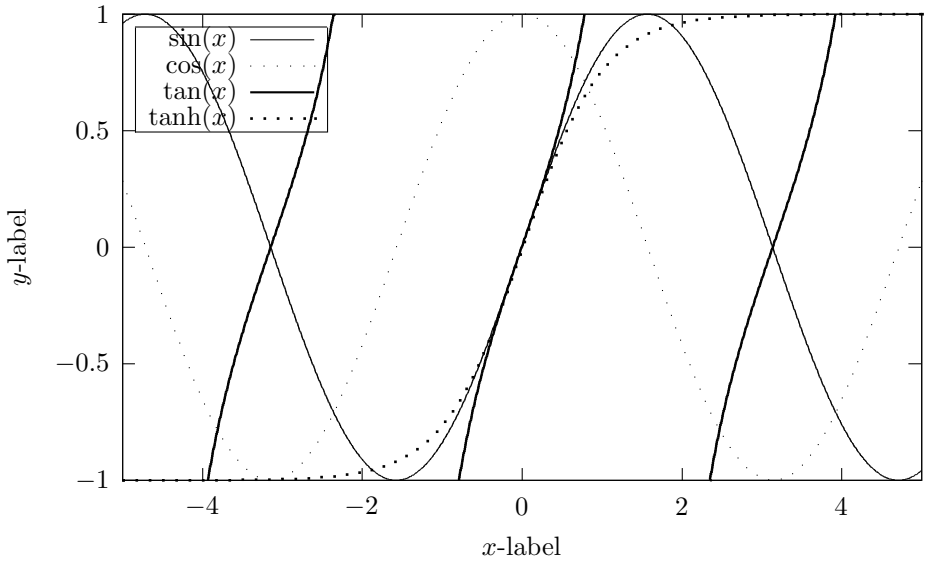

Figure 1: This is a simple example using the `latex`-terminal.

4 Examples

Here are short examples of each possible terminal.

`latex` This is a terminal that offers only basic support, as you won't be able to (easily) use colours. For an example see figure 1.

An advantage of this terminal is that it only relies on the picture-environment (which should be supported widely). Furthermore you can use the `siunitx`-option to get correct axis labeling and you can use LaTeX commands.

On the other hand, if you have multiple things to plot, it'll be hard to distinguish them, as you can't use colours (and the dash patterns are hard to distinguish). As this is something which should be avoided, especially for scientific work, if possible, you should try another terminal. The source code for figure 1 is:

```
\begin{figure}%
   \centering%
   \begin{gnuplot}[terminal=latex, terminaloptions=rotate]
      set key box top left
      set key width 4
      set sample 1000
      set xr [-5:5]
      set yr [-1:1]
      set xlabel '$x$-label'
      set ylabel '$y$-label'
      plot  sin(x) w l lc 1 t '$\sin(x)$',\
```

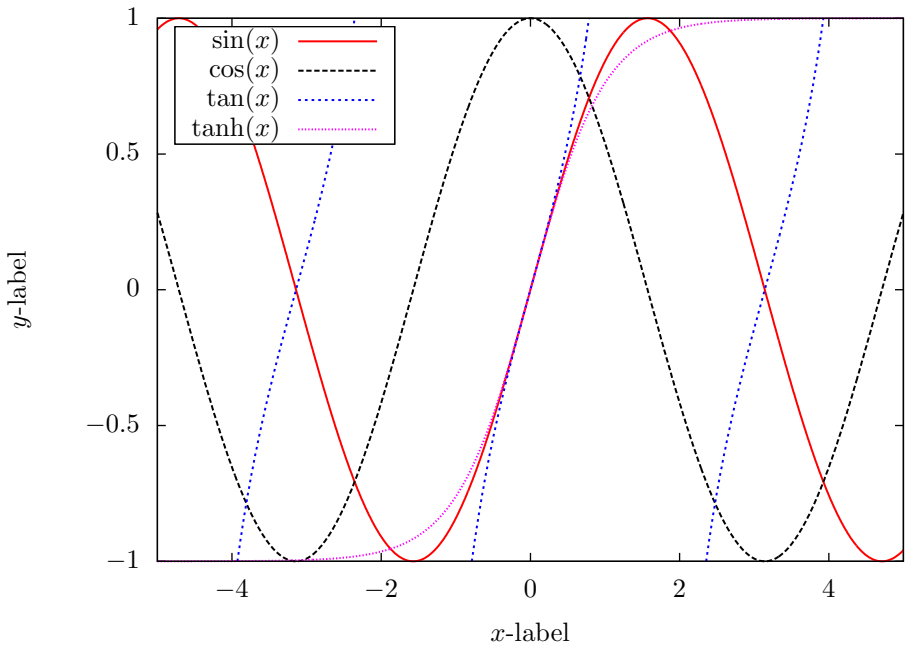

Figure 2: This is a simple example using the `epslatex`-terminal.

```
            cos(x) w l lc 2 t '$\cos(x)$',\
            tan(x) w l lc 3 t '$\tan(x)$',\
            tanh(x) w l lc 4 t '$\tanh(x)$'
    \end{gnuplot}
    \caption{This is a simple example using the latex-terminal.}%
    \label{pic:latex}%
\end{figure}%
```

`epslatex` This terminal produces high-quality output, you also can use the `siunitx`-option for axis labeling and the usual LaTeX commands.

There are rare examples where one sees flaws, like two lines not completely adhering (see e.g. some parts of the red line in figure 2). You have to use the `epstopdf`-package to use this terminal with pdfTeX.

If you're using pdfTeX this or the `cairolatex`-terminal should be the terminals you use. Your pdf viewer can slow down dramatically, if you use high resolution 3d plots though, but there are seldom real problems. For an example see figure 2.

Its code is:

```
\begin{figure}%
    \centering%
```

```
   \begin{gnuplot}[terminal=epslatex, terminaloptions=color dashed]
      set key box top left
      set key width 2
      set key opaque
      set sample 1000
      set xr [-5:5]
      set yr [-1:1]
      set xlabel '$x$-label'
      set ylabel '$y$-label'
      plot  sin(x) w l lc 1 lw 3 t '$\sin(x)$',\
            cos(x) w l lc 2 lw 3 t '$\cos(x)$',\
            tan(x) w l lc 3 lw 3 t '$\tan(x)$',\
            tanh(x) w l lc 4 lw 3 t '$\tanh(x)$'
   \end{gnuplot}
   \caption{This is a simple example using the epslatex-terminal.}%
   \label{pic:epslatex}%
\end{figure}%
```

`jpeg` This is a terminal that offers (compared with other terminals) poor raster graphics output. If possible switch to another terminal (preferably to the `cairolatex`-, the `epslatex`- or the `tikz`-terminal) For an example see figure 3.

There are quite some disadvantages, as you cannot use LaTeX commands, it is hard to get the right font and the right font size, the `siunitx` option is useless, no vector graphics and there seems to be no support for dashed lines.

Its code is:

```
\begin{figure}%
   \centering%
   \begin{gnuplot}[terminal=jpeg, terminaloptions=crop]
      set key box top left
      set key width -2
      set key opaque
      set sample 1000
      set xr [-5:5]
      set yr [-1:1]
      set size 0.7,0.7
      set xlabel '$x$-label'
      set ylabel '$y$-label'
      plot  sin(x) w l lc 1 lw 3 t '$\sin(x)$',\
            cos(x) w l lc 2 lw 3 t '$\cos(x)$',\
            tan(x) w l lc 3 lw 3 t '$\tan(x)$',\
            tanh(x) w l lc 4 lw 3 t '$\tanh(x)$'
   \end{gnuplot}
   \caption{This is a simple example using the jpeg-terminal.}%
   \label{pic:jpeg}%
\end{figure}%
```

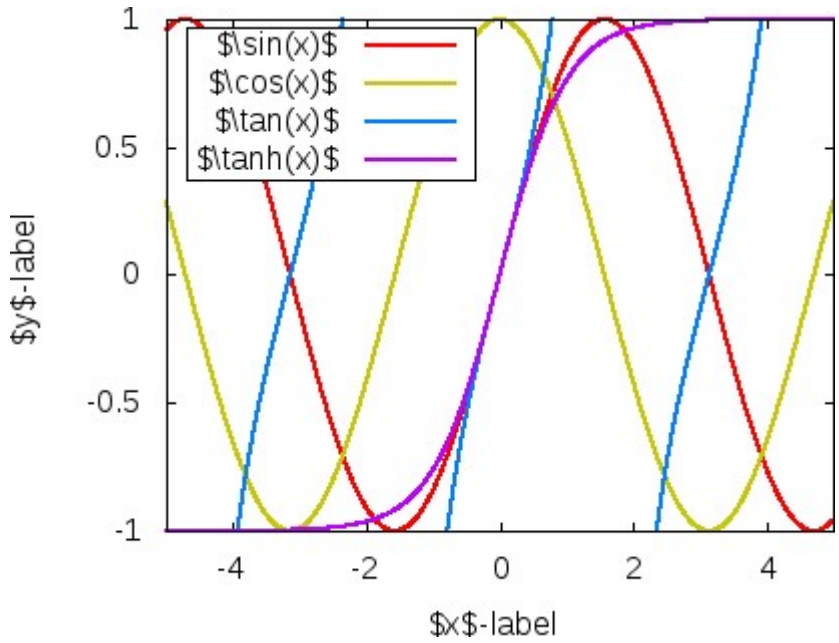

Figure 3: This is a simple example using the `jpeg`-terminal.

`cairolatex` This is a terminal that offers high-quality output. Unlike the `epslatex`-terminal it will generate *directly* pdf output. It allows the use of the `siunitx` option as well as LaTeX commands. If you're using pdfTeX this or the `epslatex`-terminal should be the terminals you use. `cairolatex` has the advantage of not needing the `epstopdf`-package. Your pdf viewer can slow down dramatically, if you use high resolution 3d plots though, but there are seldom real problems. For an example see figure 4.

Its code is:

```
\begin{figure}%
   \centering%
   \begin{gnuplot}[terminal=epslatex, terminaloptions=color dashed]
      set key box top left
      set key width 4
      set key height 0.25
      set key spacing 1.2
      set key opaque
      set sample 1000
      set xr [-5:5]
      set yr [-1:1]
      set xlabel '$x$-label'
```

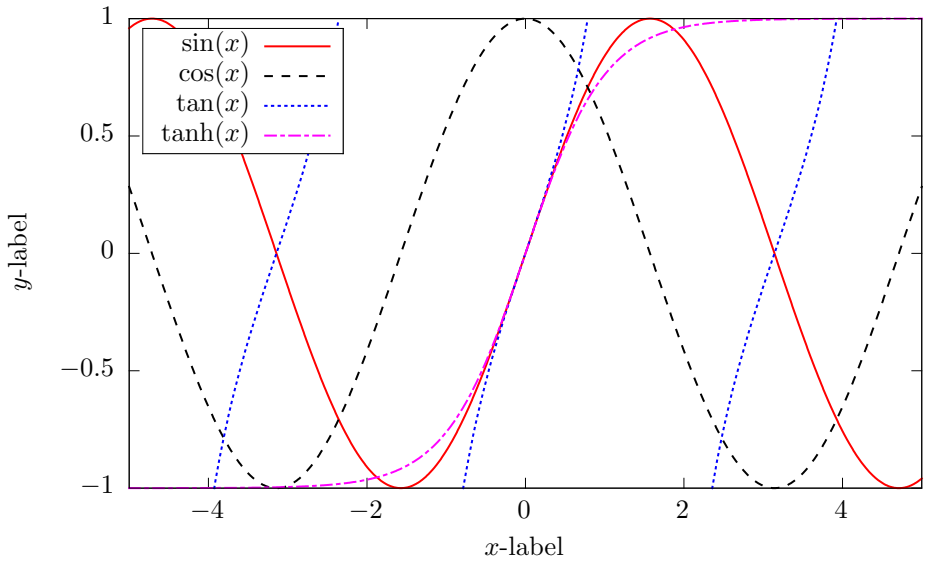

Figure 4: This is a simple example using the `cairolatex`-terminal.

```
    set ylabel '$y$-label'
    plot  sin(x) w l lc 1 lw 3 t '$\sin(x)$',\
          cos(x) w l lc 7 lw 3 t '$\cos(x)$',\
          tan(x) w l lc 3 lw 3 t '$\tan(x)$',\
          tanh(x) w l lc 4 lw 3 t '$\tanh(x)$'
  \end{gnuplot}
  \caption{This is a simple example using the cairolatex-terminal.}%
  \label{pic:cairolatex}%
\end{figure}%
```

`eepic` This is a terminal that offers better output than the `latex`-terminal, but it is quite worse compared to `epslatex/cairolatex/tikz`-terminal. You can use the `siunitx` option as well as LaTeX commands.

It seems one can use either coloured or dashed lines, not both combined. The lines are also pretty ugly and there is no `set key opaque`-option in `gnuplot` available.

For an example see figure 5. Avoid if possible. There are better terminals available.

Its code is:

```
\begin{figure}%
  \centering%
  \begin{gnuplot}[terminal=eepic, terminaloptions = color rotate]
```

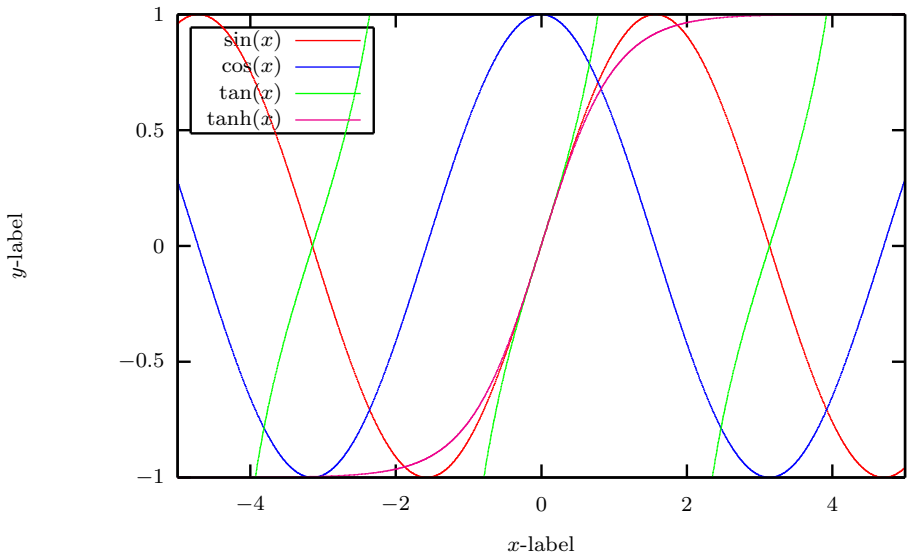

Figure 5: This is a simple example using the `eepic`-terminal.

```
        set key box top left
        #set key width -3
        set sample 1000
        set xr [-5:5]
        set yr [-1:1]
        set xlabel '$x$-label'
        set ylabel '$y$-label'
        plot  sin(x) w l lc 1 lw 3 t '$\sin(x)$',\
              cos(x) w l lc 2 lw 3 t '$\cos(x)$',\
              tan(x) w l lc 3 lw 3 t '$\tan(x)$',\
              tanh(x) w l lc 4 lw 3 t '$\tanh(x)$'
    \end{gnuplot}
    \caption{This is a simple example using the cairolatex-terminal.}%
    \label{pic:cairolatex}%
\end{figure}%
```

`emtex` Similar to the `latex`-terminal (but with emtex specials), except it seems not to draw any borders (at least for pdfTEX). Only relies on the picture environment, so should be widely supported, the `siunitx` option is available and you can use LaTeX commands.

If you have multiple things to plot, it'll be hard to distinguish them, as you can't use colours (and the dash patterns are hard to distinguish), this is a no-no for scientific work, so avoid if possible.

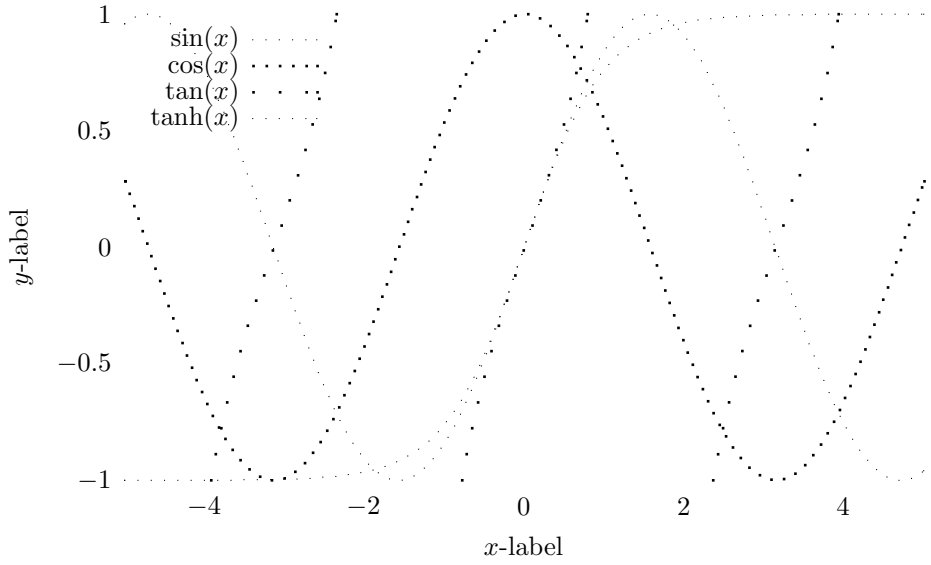

Figure 6: This is a simple example using the `emtex`-terminal.

The source code for figure 6 is:

```
\begin{figure}%
   \centering%
   \begin{gnuplot}[terminal=emtex, terminaloptions=rotate]
      set key box top left
      set key width 4
      set sample 1000
      set xr [-5:5]
      set yr [-1:1]
      set xlabel '$x$-label'
      set ylabel '$y$-label'
      plot  sin(x) w l lc 2 t '$\sin(x)$',\
            cos(x) w l lc 4 t '$\cos(x)$',\
            tan(x) w l lc 6 t '$\tan(x)$',\
            tanh(x) w l lc 8 t '$\tanh(x)$'
   \end{gnuplot}
   \caption{This is a simple example using the emtex-terminal.}%
   \label{pic:emtex}%
\end{figure}%
```

`tikz` Probably offers the best output terminal at the moment, even though the difference to `cairolatex` and `epslatex` isn't that big. The `siunitx` option is

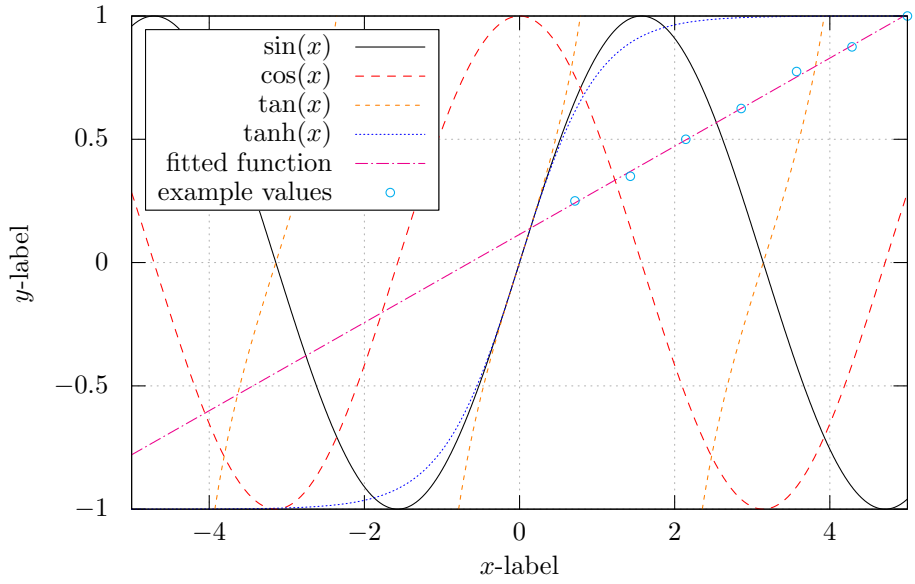

Figure 7: This is a simple example using the `tikz`-terminal. The slope of the fitted function $(a * x + c)$ is $a = 0.178\,749\,999\,989\,22$, while its offset is $c = 0.114\,285\,714\,323\,53$.

available, LaTeX commands can be used and you can easily export constants from within gnuplot (see the example code for an example on how to export them).

On the other hand requires this terminal the `gnuplot-lua-tikz`-package and you cannot use many plots if you use pdfTeX, as TeXs memory will be exceeded pretty fast. If you use LuaTeX, this is no problem.

The source code for figure 7 is:

```
\begin{figure}%
   \centering%
   \begin{gnuplot}[terminal=tikz, terminaloptions={color dashed providevars a,c}]
      set key box top left
      set key width -0.9
      set key height 0.25
      set key spacing 1.25
      set sample 1000
      set grid xtics lt 0 ls 0
      set grid ytics lt 0 ls 0
      set xr [-5:5]
```

```
      set yr [-1:1]
      set key opaque
      set xlabel '$x$-label'
      set ylabel '$y$-label'
      f(x) = a*x+c
      fit f(x) "SomeValuesForGnuplot.txt" u ($1*5/7):($2/20) via a,c
      plot  sin(x) w l lc 2 t '$\sin(x)$',\
            cos(x) w l lc 4 t '$\cos(x)$',\
            tan(x) w l lc 6 t '$\tan(x)$',\
            tanh(x) w l lc 3 t '$\tanh(x)$',\
            f(x) t 'fitted function' lc 4,\
            "SomeValuesForGnuplot.txt" u ($1*5/7):($2/20) w points lc 5\
            t 'example values'
   \end{gnuplot}
   \caption{This is a simple example using the \texttt{tikz}-terminal.
            The slope of the fitted function ($a*x+c$) is $a=\num{\gpgetvar{a}}$,
            while its offset is $c=\num{\gpgetvar{c}}$.}%
   \label{pic:tikz}%
\end{figure}%
```

As compiling all pictures everytime can take quite a long time, the `tikz`-package offers the possibility to externalize the pictures, resulting in a great speedup. An example for LuaTEX is shown below, pictures are not generated if they are up-to-date. In the preamble write:

```
\usepackage{tikz}
\usetikzlibrary{external}
\tikzexternalize[prefix=gnuplottex/]  %-- Use the gnuplottex-subfolder. Deactivate
                                      %-- the subfolderoption of gnuplottex to use!
\tikzset{external/system call={lualatex -shell-escape -halt-on-error
-interaction=batchmode -jobname "\image" "\texsource"}}
\tikzexternalize
```

In the document, write e. g.:

```
\begin{figure}%
   \centering%
   \tikzsetnextfilename{ThisIsASinus}
   \begin{tikzpicture}[gnuplot]
   \begin{gnuplot}[terminal=tikz, terminaloptions=color dashed nopicenvironment]
      set grid xtics mxtics lt 0 ls 0
      set grid ytics mytics lt 0 ls 0
      set key box top left
      plot sin(x) w l t 'Sinus'
   \end{gnuplot}
   \end{tikzpicture}
   \caption{Captiontext.}%
   \label{pic:ThisIsASinus}%
\end{figure}%
```

`pstex,pslatex,pstricks` The `pstex`- and `pslatex`-terminal produces unusable output with pdfTeX. I wasn't able to get any output using the `pstricks`-terminal with pdfTeX (if you manage to get reasonable output via one of these terminals, please send me an email).

5 Acknowledgements

In addition to the people mentioned in the changelog, I would like to thank Roy Ratcliffe for the suggestion and basic code for the gnuplot terminal specification and handling. Additional thanks to Michel Voßuhle for the implementation of `\gnuplotloadfile`. I would also like to thank all the people who sent me bug reports, feature requests and patches – Philip Vetter, sdaau, Mika Pflüger, Henri Menke, Michel Voßkuhle, Udo Höfel, Thomas Schmid and Manuel Solano. Gnuplottex wouldn't be what it is today without you.

6 Implementation

6.1 Initialization

```
\newif\ifShellEscape
\newif\ifmiktex \miktexfalse
\newif\ifusesiunitx
\newif\ifcleanup
\newif\ifusesubfolder

\newwrite\verbatim@out

\DeclareOption{shell}{\ShellEscapetrue}
\DeclareOption{noshell}{\ShellEscapefalse}
\DeclareOption{miktex}{\global\miktextrue}
\DeclareOption{siunitx}{\usesiunitxtrue}
\DeclareOption{cleanup}{\cleanuptrue}
\DeclareOption{subfolder}{\usesubfoldertrue}

\@ifundefined{gnuplotexe}{\def\gnuplotexe{gnuplot}}{}

\ExecuteOptions{shell}
\ProcessOptions\relax
%% test if shell escape really works
\ifShellEscape
\def\tmpfile{w18-test-\the\year\the\month\the\day\the\time.tex}
\ifmiktex
\immediate\write18{echo t > "\tmpfile"}
\else
\immediate\write18{touch \tmpfile}
\fi
\ifmiktex
```

```
\IfFileExists{\tmpfile.}{\ShellEscapetrue}{\ShellEscapefalse}
\immediate\write18{del "\tmpfile"}
\else
\IfFileExists{\tmpfile}{\ShellEscapetrue}{\ShellEscapefalse}
\immediate\write18{rm -f \tmpfile}
                \immediate\write18{rm -f "\jobname.gnuploterrors"}
\fi
\fi

\ifusesubfolder
    \ifmiktex
        \immediate\write18{mkdir gnuplottex}
        \immediate\write18{echo test > "gnuplottex/\tmpfile"}
    \else
        \immediate\write18{mkdir -p "gnuplottex"}
        \immediate\write18{touch "gnuplottex/\tmpfile"}
    \fi
    \IfFileExists{gnuplottex/\tmpfile}{
        \ifmiktex
            \immediate\write18{del "gnuplottex\@backslashchar\tmpfile"}
        \else
            \immediate\write18{rm -f "gnuplottex/\tmpfile"}
        \fi
        \def\subfolder{gnuplottex/}
        }{
            \PackageWarningNoLine{gnuplottex}
                {Creation of subfolder failed.\MessageBreak
                You'll need to create the folder yourself}
            \def\subfolder{}
        }
\else
    \def\subfolder{}
\fi


\ifShellEscape
    \PackageInfo{gnuplottex}
    {Automatically converting gnuplot files.}
\else
    \PackageWarningNoLine{gnuplottex}
        {Shell escape not enabled.\MessageBreak
        You'll need to convert the graphs yourself.}
\fi
\newcounter{fignum}
```

6.2 .gnuplot write out

```
\def\figname{\jobname-gnuplottex-fig\thefignum}

\def\usesiunitxingnuplot{\immediate\write\verbatim@out{set format '\@backslashchar num{\@per
```

```
\def\gnuplotverbatimwrite#1{%
    \def\BeforeStream
    {\message{Opening gnuplot stream #1}%
        \immediate\write\verbatim@out{\string set terminal \gnuplotterminal \gnuplotterminal
\immediate\write\verbatim@out{\string set output '\subfolder\figname.\gnuplottexextension{\g
    \ifusesiunitx
        \ifthenelse{\equal{\extension}{\string tex}}{\usesiunitxingnuplot}{\PackageWarningNo
    \else
        \relax
    \fi
    }
    \@bsphack
    \immediate\openout \verbatim@out #1
    \BeforeStream%
    \let\do\@makeother\dospecials
    \catcode`\^^M\active
    \def\verbatim@processline{%
        \immediate\write\verbatim@out
        {\the\verbatim@line}}%
    \verbatim@start}
\def\endgnuplotverbatimwrite{%
    \immediate\closeout\verbatim@out
    \@esphack
\catcode`\\0
\catcode`\{1
\catcode`\}2
\catcode`\$3
\catcode`\&4
\catcode`\^^M5
\catcode`\#6
\catcode`\^7
\catcode`\_8
\catcode`\ 10
\catcode`\%14}
```

6.3 Environment definition

```
\def\gnuplottexextension@latex{\string tex}
\def\gnuplottexextension@epslatex{\string tex}
\def\gnuplottexextension@cairolatex{\string tex}
\def\gnuplottexextension@eepic{\string tex}
\def\gnuplottexextension@pstricks{\string tex}
\def\gnuplottexextension@pslatex{\string tex}
\def\gnuplottexextension@pstex{\string tex}
\def\gnuplottexextension@emtex{\string tex}
\def\gnuplottexextension@jpeg{\string jpg}
\def\gnuplottexextension@tikz{\string tex}
\def\gnuplottexextension@lua{\string tex}
\def\gnuplottexextension#1{\@ifundefined{gnuplottexextension@#1}{#1}{\csname gnuplottexexter
```

```
\define@key{pic}{scale}[1]{\def\gnuplotscale{#1}}
\define@key{pic}{terminal}[latex]{\def\gnuplotterminal{#1}}
\define@key{pic}{terminaloptions}{\def\gnuplotterminaloptions{ #1}}
\newenvironment{gnuplot}[1][]{\stepcounter{fignum}%
\def\gnuplotterminal{latex}
\def\gnuplotterminaloptions{}
\def\gnuplotscale{1}
\setkeys{pic}{#1}
    \xdef\gnuplotCutFile{\subfolder\figname.gnuplot}
    \gnuplotverbatimwrite{\gnuplotCutFile}}
    {\endgnuplotverbatimwrite%
    \gnuplotgraphicsprocess%
    \gnuplotgraphicsinclude}
```

6.4 .gnuplot file processing

```
\def\extension{\gnuplottexextension{\gnuplotterminal}}
\long\gdef\gnuplotgraphicsprocess{%
\ifShellEscape
\IfFileExists{\subfolder\figname.gnuplot}{%
                        \ifmiktex
                            \immediate\write18{\gnuplotexe\space \subfolder\figname.gnuplot}
                        \else
                            \immediate\write18{\gnuplotexe\space \subfolder\figname.gnuplot\
                        \fi
\IfFileExists{\subfolder\figname.\extension}{%
\PackageInfo{gnuplottex}{\subfolder\figname.gnuplot converted}
                \ifcleanup
                    \ifmiktex
                        \immediate\write18{del "\subfolder\figname.gnuplot"}
                    \else
                        \immediate\write18{rm -f "\subfolder\figname.gnuplot"}
                    \fi
                \fi
            }
{\PackageWarningNoLine{gnuplottex}
{Conversion of \subfolder\figname.gnuplot failed}}}{}
\fi}
\long\def\gnuploterrors@eatpar#1#2\@nil{\def\gnuploterrors@{#2}}
\AtEndDocument{%
  \CatchFileDef\gnuploterrors@{\jobname.gnuploterrors}{\endlinechar=`^^J \catcode`\ =12 }%
  \expandafter\gnuploterrors@eatpar\gnuploterrors@\@nil
  \ifx\gnuploterrors@\@empty\else
    \PackageWarningNoLine{gnuplottex}{Gnuplot execution produced errors:^^J%
      \detokenize\expandafter{\gnuploterrors@}}%
  \fi
}
```

6.5 Graph inclusion

```
\long\gdef\gnuplotgraphicsinclude{%
```

```
\IfFileExists{\subfolder\figname.\extension}{%
\ifthenelse{\equal{\extension}{\string tex}}
{\scalebox{\gnuplotscale}{\input{\subfolder\figname.\extension}}}
{\includegraphics[scale=\gnuplotscale]{\subfolder\figname.\extension}}
}
{\PackageWarningNoLine{gnuplottex}
{Please convert \subfolder\figname.gnuplot manually}}
}
```

6.6 .gnuplot file processing

```
\newcommand{\gnuplotloadfilewrite}[2]{%
  \immediate\openout \verbatim@out #1%
  \message{Opening gnuplot stream #1}%
  \immediate\write\verbatim@out{\string set terminal \gnuplotterminal \gnuplotterminaloption
  \immediate\write\verbatim@out{\string set output '\subfolder\figname.\gnuplottexextension{
  \ifusesiunitx
    \ifthenelse{\equal{\extension}{\string tex}}{\usesiunitxingnuplot}{\PackageWarningNoLine
  \else
    \relax
  \fi
  \ifusesiunitx
      {\escapechar=-1\edef\percentforgnuplot{\string\%}
          \escapechar=-1\edef\backslashforgnuplot{\string\\}
      \immediate\write\verbatim@out{set format '\backslashforgnuplot num{\percentforgnuplot
  \else
      \relax
  \fi
  \immediate\write\verbatim@out{\string load '#2'}%
  \immediate\closeout\verbatim@out%
 }

 \newcommand{\gnuplotloadfile}[2][]{
  \stepcounter{fignum}%
  \def\gnuplotterminal{latex}
  \def\gnuplotterminaloptions{}
  \def\gnuplotscale{1}
  \setkeys{pic}{#1}
  \xdef\gnuplotCutFile{\subfolder\figname.gnuplot}
  \gnuplotloadfilewrite{\gnuplotCutFile}{#2}
  \gnuplotgraphicsprocess%
  \gnuplotgraphicsinclude
 }
```