`LaPrint` Users Guide

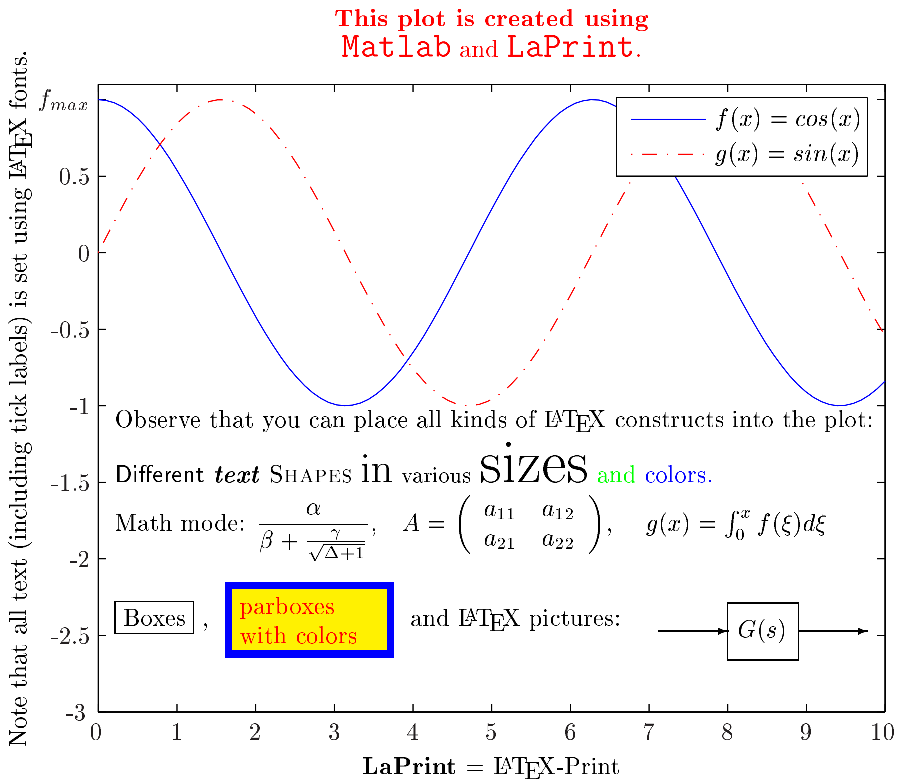
(`LaPrint` Version 3.16)

Arno Linnemann
Control and Automation
Department of Electrical and Computer Engineering
University of Kassel
D-34109 Kassel
Germany
`linnemann@uni-kassel.de`

September 13, 2004

Abstract

`LaPrint` is a Matlab function to print Matlab graphics for inclusion in LaTeX documents. `LaPrint` creates an eps-file and a tex-file. The tex-file contains the annotation of the figure such as titles, labels and texts. The eps-file contains the non-text part of the figure and is called by the tex-file. The main advantage of using `LaPrint` is that the annotation can be neatly (e.g., including math mode and fancy font constructs) set within LaTeX. `LaPrint` can be used from the command line or via a graphical user interface (GUI). This note introduces into `LaPrint`. It is assumed that the reader is familar with LaTeX and Matlab.



1 What `LaPrint` does ...

`LaPrint` performs the following tasks:

- Replace all occurences of text in the Matlab figure by tags;
- Save the modified figure in postscript format (eps-file);
- Create a tex-file which contains commands of the LaTeX psfrag package to replace the tags by the original text and to call the postscript file.

When the tex-file is included in a LaTeX - document, all text will be set within LaTeX. The effect of using `LaPrint` is illustrated by the example on the titlepage of this document.

2 Prerequisites

Matlab 6.1 or above is required. To process the LaTeX file, a LaTeX compiler (including the packages 'graphicx' and 'psfrag') and a dvi-to-postscript converter (like dvips) is required. Optional tools are a postscript-viewer (like ghostview), a postscript bounding box converter (like epstool) and the LaTeX packages color and epsfig. Full color support requires Matlab 6.5 or above.

3 Download and Installation

The most recent version of `LaPrint` is available from the 'Matlab Central File Exchange' at

```
http://www.mathworks.de/matlabcentral/fileexchange/
                                 loadFile.do?objectId=4638&objectType=file
```

The most recent version of this document is available from

```
http://www.uni-kassel.de/fb16/rat/matlab/laprint/laprintdoc.ps
```

`LaPrint` comes as a single ready-to-run m-file, which has to be placed somewhere in the Matlab search path. If `LaPrint` is intended to internally call LaTeX and friends, the (system dependent) executables have to be defined using the GUI (see below).

Before opening a Matlab figure to be printed with `LaPrint`, the Matlab TeX interpreter should be switched off by typing

```
>> set(0,'defaulttextinterpreter','none')
```

One might consider placing this command into the `startup.m` file.

4 Getting started

Let's start with a very simple Matlab graphics

```
>> set(0,'defaulttextinterpreter','none')
>> figure(1), clf
>> plot([1 2])
>> ylabel('A straight line')
```

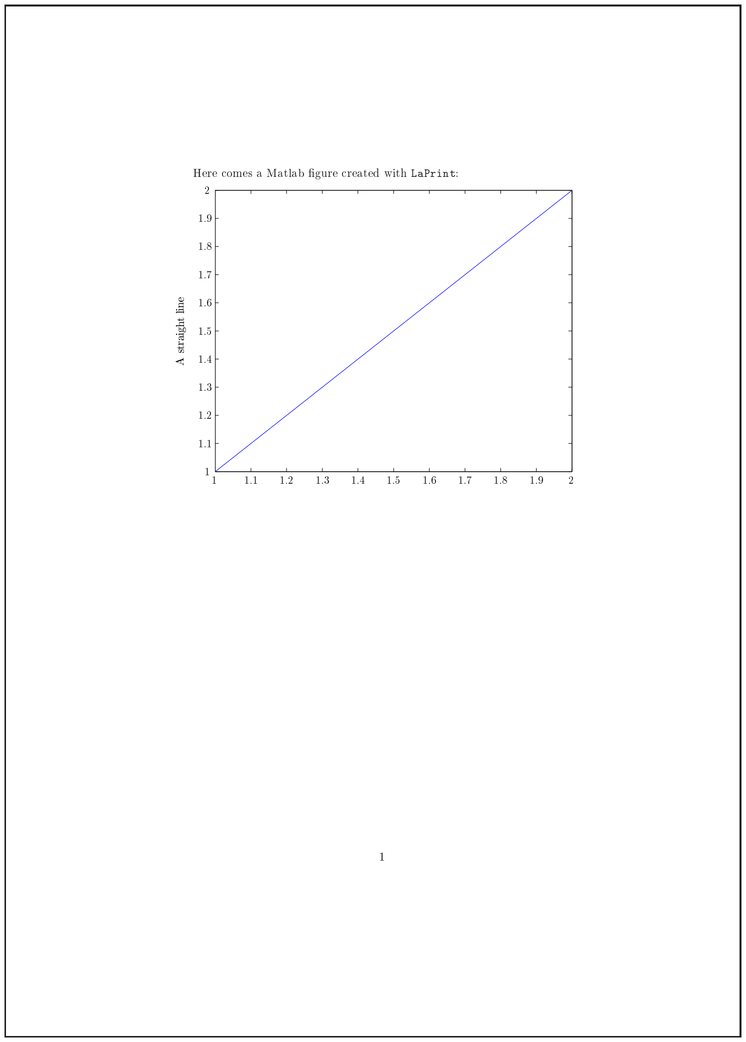
`LaPrint` can be used from the command line (equivalently, from an m-file) or via a GUI. Let's start with the GUI, which is opened by typing

```
>> laprint
```

Check that the first text field refers to Figure No. 1 and enter a filename in the second text field (e.g., `myfirstfig`). Pushing the "Go!"-Button prints Figure No. 1 into the files `myfirstfig.eps` and `myfirstfig.tex`. These files can be included in a LaTeX document (`myfirstdoc.tex`, say) as follows:

```
\documentclass{article}
\usepackage{graphicx,color,psfrag}
\begin{document}
Here comes a Matlab figure created with \texttt{LaPrint}:\\
\input{myfirstfig.tex}
\end{document}
```

Translating this file with LaTeX and converting the dvi-file to postscript (using, e.g., dvips) leads to a postscript document (file `myfirstdoc.ps`), which should look as follows (reduced in size for presentation purposes):



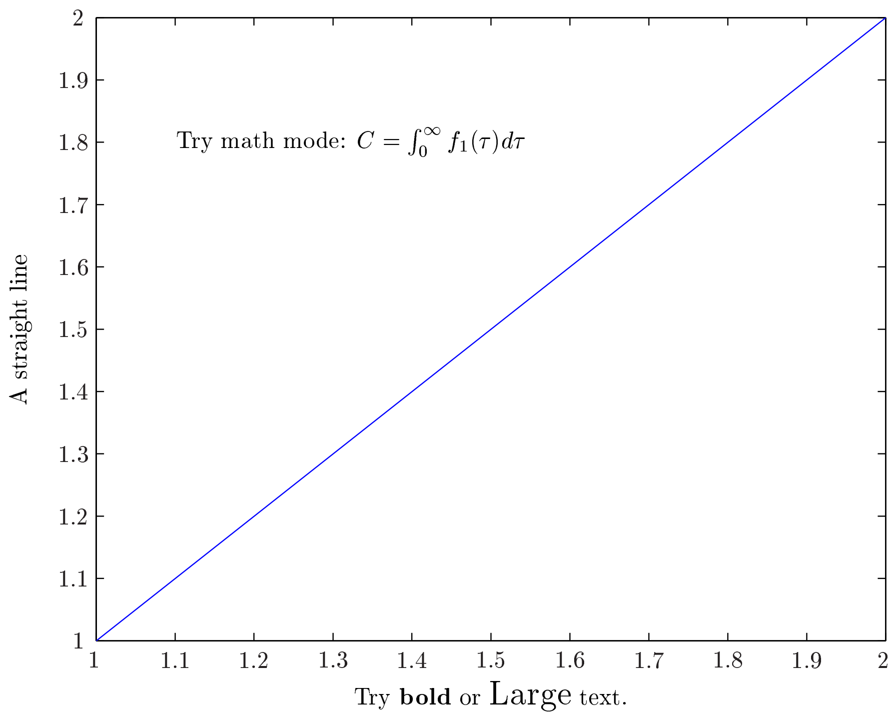
`myfirstdoc.ps`

Observe that the text part of the figure (the tick labels and the ylabel, in this case) are printed using LaTeX fonts.

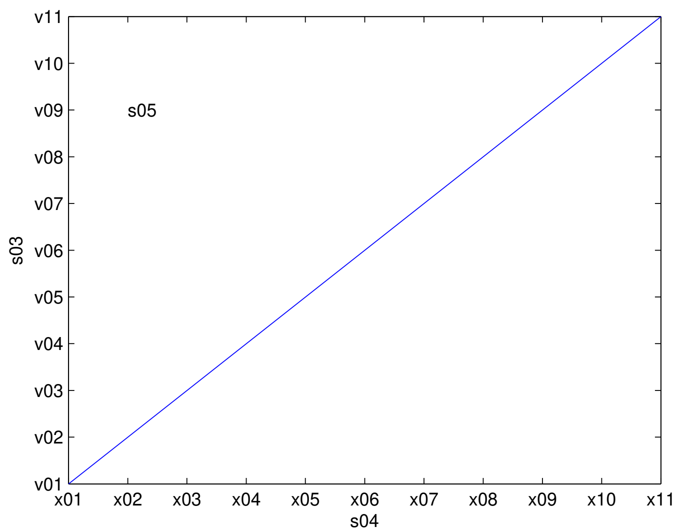
Let us now add specific LaTeX constructs to the figure:

```
>> xlabel('Try \textbf{bold} or {\Large Large} text.')
>> text(1.1,1.8,'Try math mode: $C=\int_0^\infty f_1(\tau)d\tau$')
```

Save the modified figure using `LaPrint` (keep the basename `myfirstfig` to overwrite the existing files) and push "Go!". The compilation of the file `myfirstdoc.tex` using latex/dvips leads to the following figure in the resulting document `myfirstdoc.ps`:



To understand what `LaPrint` has done, let us have a look at the files `myfirstfig.eps` and `myfirstfig.tex`, which have been created by `LaPrint`. `myfirstfig.eps` contains the following graphics:



Thus it contains, for each text portion of the figure, certain tags (s03, x01, etc.). Actually, `LaPrint` has created a copy of the original figure and has replaced all text by tags.

The file `myfirstfig.tex`, contains the following text:

```
% This file is generated by the MATLAB m-file laprint.m. It can be included
% into LaTeX documents using the packages graphicx, color and psfrag.
% It is accompanied by a postscript file. A sample LaTeX file is:
%    \documentclass{article}\usepackage{graphicx,color,psfrag}
%    \begin{document}\input{myfirstfig}\end{document}
% See http://www.mathworks.de/matlabcentral/fileexchange/loadFile.do?objectId=4638
% for recent versions of laprint.m.
%
% created by:           LaPrint version 3.16 (13.9.2004)
% created on:           13-Sep-2004 21:13:29
% eps bounding box:     15 cm x 11.25 cm
% comment:
%
\begin{psfrags}%
\psfragscanon%
%
% text strings:
\psfrag{s03}[b][b]{\color[rgb]{0,0,0}\setlength{\tabcolsep}{0pt}\begin{tabular}{c}A straight line\end{tabular}}%
\psfrag{s04}[t][t]{\color[rgb]{0,0,0}\setlength{\tabcolsep}{0pt}\begin{tabular}{c}Try \textbf{bold} or {\Large Large} text.\end{tabular}}%
\psfrag{s05}[l][l]{\color[rgb]{0,0,0}\setlength{\tabcolsep}{0pt}\begin{tabular}{l}Try math mode: $C=\int_0^\infty f_1(\tau)d\tau$\end{tabular}}%
%
% xticklabels:
\psfrag{x01}[t][t]{1}%
\psfrag{x02}[t][t]{1.1}%
\psfrag{x03}[t][t]{1.2}%
\psfrag{x04}[t][t]{1.3}%
\psfrag{x05}[t][t]{1.4}%
\psfrag{x06}[t][t]{1.5}%
\psfrag{x07}[t][t]{1.6}%
\psfrag{x08}[t][t]{1.7}%
\psfrag{x09}[t][t]{1.8}%
\psfrag{x10}[t][t]{1.9}%
\psfrag{x11}[t][t]{2}%
%
% yticklabels:
\psfrag{v01}[r][r]{1}%
\psfrag{v02}[r][r]{1.1}%
\psfrag{v03}[r][r]{1.2}%
\psfrag{v04}[r][r]{1.3}%
\psfrag{v05}[r][r]{1.4}%
\psfrag{v06}[r][r]{1.5}%
\psfrag{v07}[r][r]{1.6}%
\psfrag{v08}[r][r]{1.7}%
\psfrag{v09}[r][r]{1.8}%
\psfrag{v10}[r][r]{1.9}%
\psfrag{v11}[r][r]{2}%
%
% Figure:
\resizebox{12cm}{!}{\includegraphics{myfirstfig.eps}}%
\end{psfrags}%
```

```
%
% End myfirstfig.tex
```

Thus, it contains psfrag commands to replace the tags by proper LaTeX commands. At the end of the file, `myfirstfig.eps` is included.

5 The 'Options' Menu Bar

`LaPrints` default behaviour can be changed using the 'Options' menu bar. The features are described below.

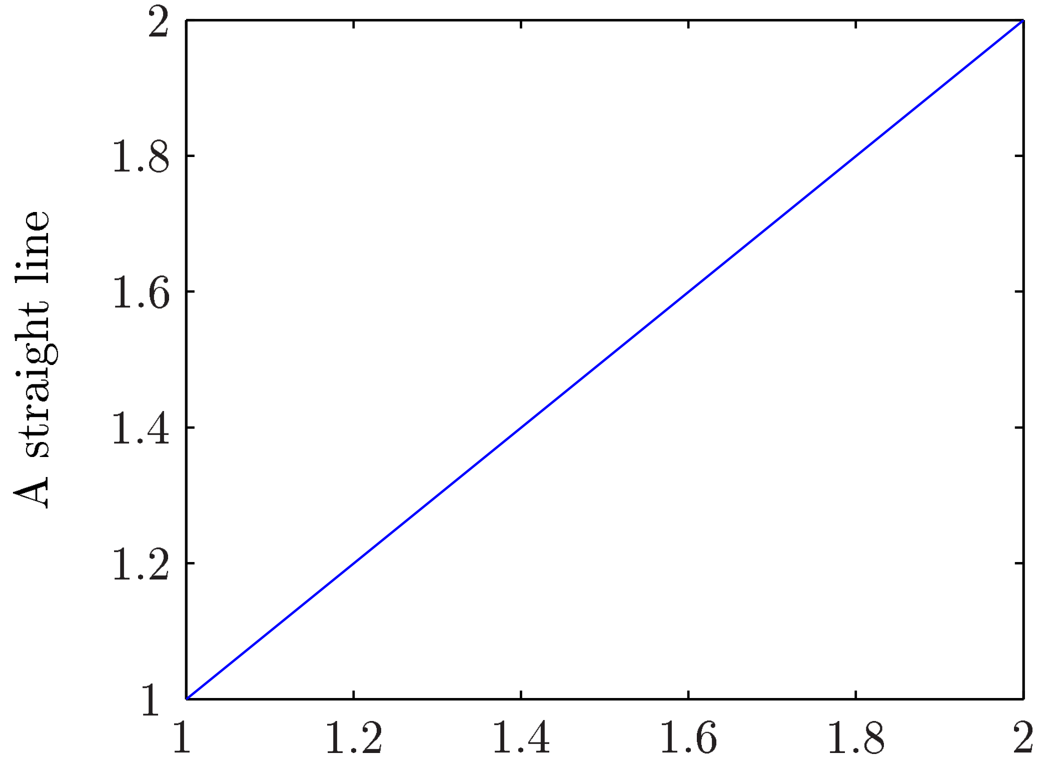
5.1 Sizes and Scalings ...

By default, `LaPrint` creates a graphics which is 12 cm wide. A different width can be entered here. The height of the graphics is always such that the aspect ratio of the figure on screen is retained.

Observe, that the text of the Figures in Section 4 is slightly smaller than the surrounding text. By default, `LaPrint` scales the figure down by a factor 0.8. This implies that the fonts are smaller and the lines are thinner. The factor can be changed here. A factor greater than 1 scales the figure up, which is particularly useful for presentations.

Switching font scaling off scales the graphics only. The default value is 'on'.

The user might want to experiment with some width/factor/scalefont combinations to find the personal preference. For instance, the following figure is created with width=14 and factor=1.7.



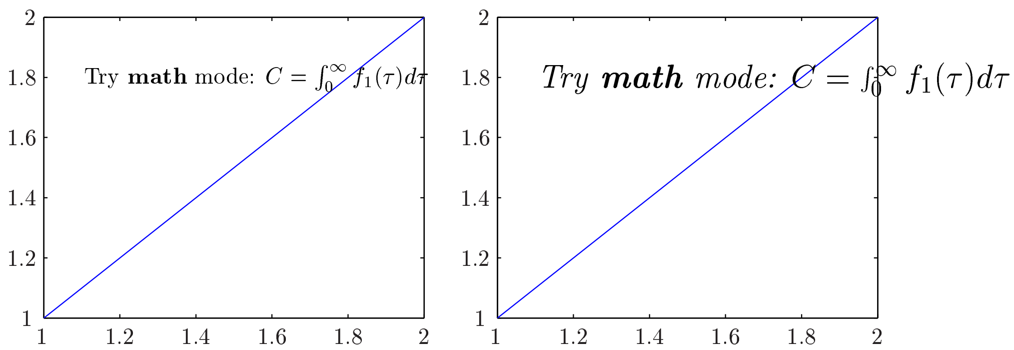
5.2 Translate Matlab Font Properties to LaTeX

This option can be switched on/off. When 'off' (which is the default value), all text will be set using the LaTeX font which is active while entering the tex-file (Matlab font settings are ignored). When 'on', Matlab font settings are translated to LaTeX font settings.

As an example, let us consider a graphics created by

```
figure(1), clf
plot([1 2])
h=text(1.1,1.8,'Try \textbf{math} mode: $C=\int_0^\infty f_1(\tau)d\tau$');
set(h,'fontsize',14.4,'fontangle','italic')
```

The following graphics result with 'Translate Matlab Font Properties to LaTeX' switched 'off' (left) and 'on' (right), respectively.



5.3 Print Limits and Ticks as on Screen

This option can be switched on/off. When 'on', the axes limits and ticks are frozen. They appear in the LaTeX document as on screen. When 'off' (which is the default value), they are adapted by Matlab.

5.4 Keep Tick Labels within eps File

This option can be switched on/off. When 'on', the tick labels aren't replaced by their LaTeX equivalents. This is useful for some 3D-plots. The default value in 'off'.

5.5 Set Tick Labels in LaTeX Math Mode

This option can be switched on/off. When 'on', the tick labels are surrounded by a '$'. The default value in 'off'.

5.6 Equip the tex File with a Head

This option can be switched on/off. When 'on' (which is the default value), the tex-file is equipped with a header containing a bunch of comments (date, files, example, comment text,...).

5.7 Comment in the Head of the tex File ...

This opens a window where a comment to be placed in the header of the tex file can be entered.

5.8 Place a LaTeX caption in the tex File ...

This opens a window where to enter a caption text. If this caption text is nonempty, it is placed as a `\caption{}` into the tex-file. Thus the text will appear in the LaTeX document. In addition a `\label{}` is placed in the tex-file which can be used to refer to the graphics. The label will be `fig:` followed by the basename of the files, e.g., `\label{fig:unnamed}`. The default value is `''`, which means that the tex file doesn't contain a caption and label.

5.9 Place an Extra Picture in each Axes

This option can be switched on/off. When 'on', each axes in the figure is equipped with an empty LaTeX picture environment. It can be used to place some additional material into the figure by editing the tex-file. The default value in 'off'.

5.10 Length of the psfrag Replacement Strings

This opens a window to enter the length of the psfrag tags. If there are large amounts of text ($\geq$99 entries), the default length of 3 might not suffice. The length can also be used to control the distance of the ylabel from the main figure.

5.11 Call LaPrint in verbose mode

This option can be switched on/off. When 'on', `LaPrint` issues some messages and warns before overwriting files. When 'off' (which is the default value), **`LaPrint` does not care overwriting files!**

5.12 Copy Figure and Modify that Copy

This option can be switched on/off. When 'on' (which is the default value), `LaPrint` takes a copy of the figure to be printed and places its tags into that copy. When done, the copy is deleted. When 'off', the original figure is messed up by tags. It gets unusable, but one can see what `LaPrint` is doing. Besides, there are bugs in the Matlab 'copyobj' command. If these show up, this option is useful.

5.13 Matlab Print Command ...

This opens a window to enter the Matlab command to save the graphics in an eps-file. The standard print command can be modified to include further options (like `-loose`) or replaced by a different command (like `exportfig`). When specifying the print command, the tags `<figurenumber>`, `<filename.eps>` and `<filename>` can be used. They will be internally replaced by the number of the figure, the full name of the eps-file and the basename of the files, respectively, as entered in the main `LaPrint` window. The default value is `print('-f<figurenumber>','-depsc','-painters',<filename.eps>)`.

5.14 Matlab Graphics Package

There is the choice between 'graphicx' (which is recommended and the default) and 'epsfig'.

5.15 Use LaTeX 'color' package

This option can be switched on/off. When 'on' (which is the default value for Matlab 6.5 or above), `LaPrint` places color commands into the tex file to optain colored text. For colored text, the calling document has to include

```
\usepackage{color}
```

5.16 View file ...

`LaPrint` can create and a third file (called view-file), which is a complete LaTeX document. It basically contains

```
\documentclass{article}
\usepackage{graphicx,color,psfrag}
\begin{document}
\pagestyle{empty}
    \input{unnamed.tex}
\end{document}
```

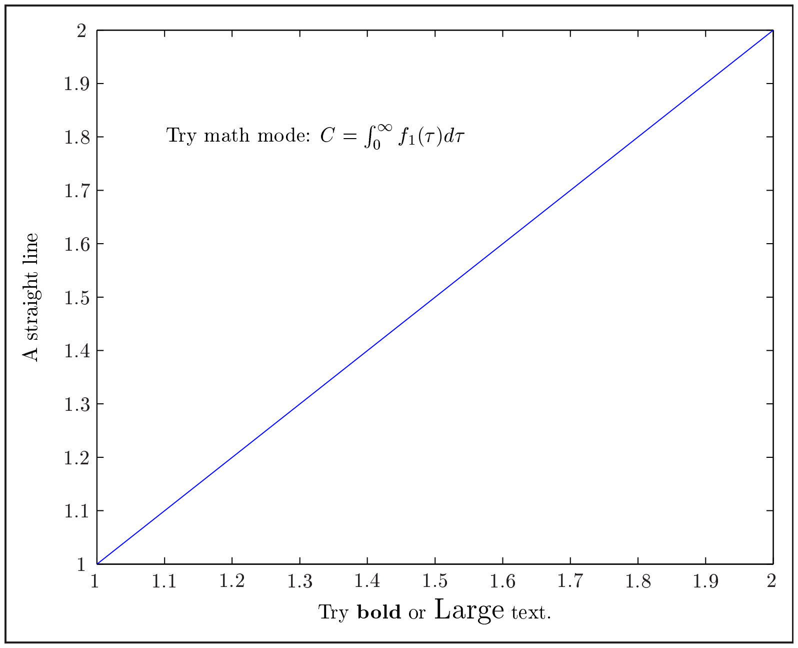
where `unnamed` is the basename of the other two files. `LaPrint` can also automatically call LaTeX / dvips and other programs to process this file. This feature is useful to have a quick look at the result. It can also be used to create an eps-file which contains everything (Matlab figure and LaTeX annotation), see below.

The option **Create a View File** can be switched on/off. When 'on' the view file is created. The name of the view file can be specified using the option **Name of the View File**. The default is not to create a view-file.

The option **Process the View File** can be switched on/off. When 'on' the view file is processed using LaTeX and friends. The names of the programs to process the view file and associated command line options are specified using the option **Executables for Processing View File**. The executables are highly system dependent (depending on operating system, LaTeX distribution, installation paths, etc.). Thus, almost certainly, the user has to adapt these values. The factory default values assume a typical installation and create an eps-file containing everything. When the Matlab figure in Section 4 is processed with the default executables, the resulting eps-file can be included with

```
\fbox{\includegraphics[width=10cm]{myfirstfig_final.eps}}
```

The result is shown in the following figure.



6 The 'Preferences' Menu Bar

The settings entered using the 'Options' menu bar are normally lost when closing the `LaPrint` GUI window. It is however possible to store the current settings of `LaPrint` in 'Preferences' by chosing

Set Preferences to Current Settings

This means that future sessions of `LaPrint` load these settings on startup. These settings can also be manually loaded by chosing

Get Preferences

`LaPrint` has built-in (factory default) preferences, which show up when using `LaPrint` the first time. These defaults can be restored by chosing

Get Factory Defaults

The choice

Remove Preferences

completely removes the personal preferences (switching back to factory defaults on startup). If there are multiple projects running with different `LaPrint` settings, these can be saved/loaded to/from mat-files by chosing

Save Current Settings to a File

and

Load Settings from a File

These open a new window asking for the name of a mat-file.

7 Using `LaPrint` from the command line

The `LaPrint` GUI is good for quickly handling some few figures. For larger projects and for an automated creation of LaTeX figures within m-files, the command line version of `LaPrint` is more appropriate. The following syntax is required:

```
laprint(figno,filename)
laprint(figno,filename,opta,vala,optb,valb,...)
```

where

`figno`	:	handle of figure to be printed (an integer)
`filename`	:	basename of files to be created (a character array)
`opta,vala,...`	:	option and value pairs, where `opta` is one of the options in the table below (a character array) and `vala` is the corresponding value.

When `LaPrint` is called from the command line without option/value pairs, i.e.,

```
laprint(figno,filename)
```

then settings/options are used which are identical to the ones of the GUI on startup. This means that `LaPrint` uses

- either personal preferences (created using 'Set Preferences to Current Settings' in the GUI)

- or factory default settings.

The factory default settings can be enforced by typing

```
laprint(figno,filename,'options','factory')
```

Preferences in a mat-file (created using 'Save Current Settings to a File' in the GUI), can be employed by typing

```
laprint(figno,filename,'options','my-laprint-prefs.mat'),
```

where `my-laprint-prefs.mat` is the name of the mat-file. The settings described so far are overwritten by option/value pairs `opta, vala, optb, valb, ...` according to the following table. These options correspond to the ones described in Section 5.

Option	Value	Remarks
`width`	a double	the value specifies the width of the figure in the LaTeX document (in cm), see section 5.1
`factor`	a double	the value is a factor by which the figure in the LaTeX document is smaller than the figure in the postscript file, see section 5.1
`scalefonts`	`'on'` or `'off'`	if `'on'`, the fonts are scaled with the figure, see section 5.1
`keepfontprops`	`'on'` or `'off'`	if `'on'`, Matlab font properties (size, width, angle) are translated to corresponding LaTeX font properties, see section 5.2
`asonscreen`	`'on'` or `'off'`	if `'on'`, ticks and lims are printed as on screen, see section 5.3
`keepticklabels`	`'on'` or `'off'`	if `'on'`, the tick labels are kept within the eps-file (and are thus not processed by LaTeX), see section 5.4
`mathticklabels`	`'on'` or `'off'`	if `'on'`, tick labels are set in LaTeX math mode, see section 5.5
`head`	`'on'` or `'off'`	if `'on'`, a commenting header is placed into the tex-file, see section 5.6
`comment`	a char array	the value is a commenting text to be placed into the header of the tex-file, see section 5.7
`caption`	a char array	the commands `\caption{<string>}` and `\label{fig:<filename>}` are placed into the tex-file, where `<string>` is the value and `<filename>` is the 2nd argument of laprint, see section 5.8
`extrapicture`	`'on'` or `'off'`	if `'on'`, empty picture environments are placed to the axes, see section 5.9
`nzeros`	an integer	the value specifies the length of the replacement strings in the eps-file, see section 5.10
`verbose`	`'on'` or `'off'`	if `'on'`, verbose mode is used; it asks before overwriting files and issues some more messages, see section 5.11
`figcopy`	`'on'` or `'off'`	if `'on'`, a copy of the figure is modified, otherwise, the original figure is modified directly, see section 5.12
`printcmd`	a char array	the value is the Matlab print command to be used, see section 5.13

Option	Value	Remarks
`package`	a char array	the value is the LaTeX graphics package to be used. Possible values are `graphicx` and `epsfig`, see section 5.14
`color`	`'on'` or `'off'`	if `'on'`, colored fonts are set, using the LaTeX `color` package, see section 5.15
`createview`	`'on'` or `'off'`	if `'on'`, a view-file is created, which calls the tex-file, see section 5.16
`viewfilename`	a char array	the value is the basename of the view-file, see section 5.16
`processview`	`'on'` or `'off'`	if `'on'`, the view-file is processed by LaTeX and friends, see section 5.16
`cmd1`	a char array	the value is the 1st command in the sequence of LaTeX and friends, see section 5.16
`cmd2`	a char array	2nd LaTeX/friend command, see section 5.16
`cmd3`	a char array	3rd LaTeX/friend command, see section 5.16
`cmd4`	a char array	4th LaTeX/friend command, see section 5.16
`cmd5`	a char array	5th LaTeX/friend command, see section 5.16
`cmd6`	a char array	6th LaTeX/friend command, see section 5.16
`cmd7`	a char array	7th LaTeX/friend command, see section 5.16
`cmd8`	a char array	8th LaTeX/friend command, see section 5.16
`options`	a char array or `'factory'`	If the value is `'factory'`, the factory default settings are employed. If the value is another character array, then it is assumed to be the name of a mat-file to be loaded. The usage of this option is discussed above.

If `LaPrint` is called as follows:

```
>> laprint helpwindow
```

then a window with some online help opens.

The `LaPrint` GUI and the command line version allow identical options concerning the creation of figures. Actually, the GUI internally calls the command line version. However, preferences can only be saved using the GUI.

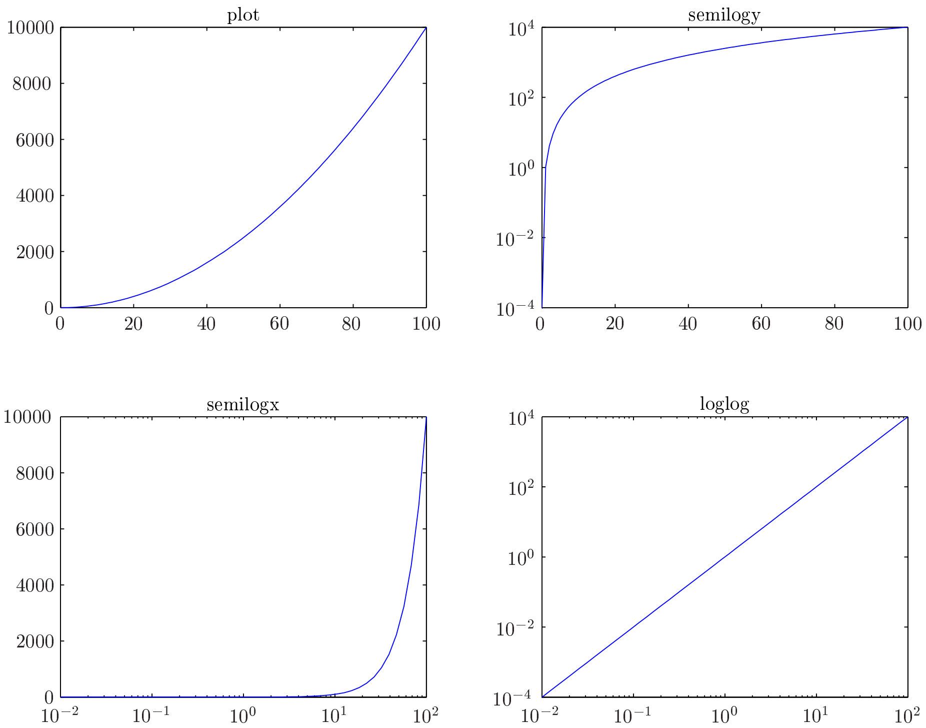
8 Examples

This section contains some Examples for the command line usage of `LaPrint`. The same results can also be obtained using the GUI. In each of the examples it is assumed that the Matlab TeX interpreter is switched off and that Figure 1 is active and cleared. This is be accomplished by

```
set(0,'defaulttextinterpreter','none')
figure(1), clf
```

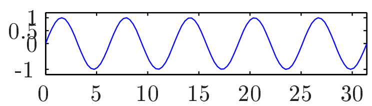
8.1 Subplots and log scales

```
x=linspace(1e-2,1e2); y=x.^2;
w=logspace(-2,2);     z=w.^2;
subplot(2,2,1), plot(x,y),     title('\vspace{-3mm}plot')
subplot(2,2,2), semilogy(x,y), title('\vspace{-3mm}semilogy')
subplot(2,2,3), semilogx(w,z), title('\vspace{-3mm}semilogx')
subplot(2,2,4), loglog(w,z),   title('\vspace{-3mm}loglog')
laprint(1,'ex_subplot','options','factory','factor',0.6)
```



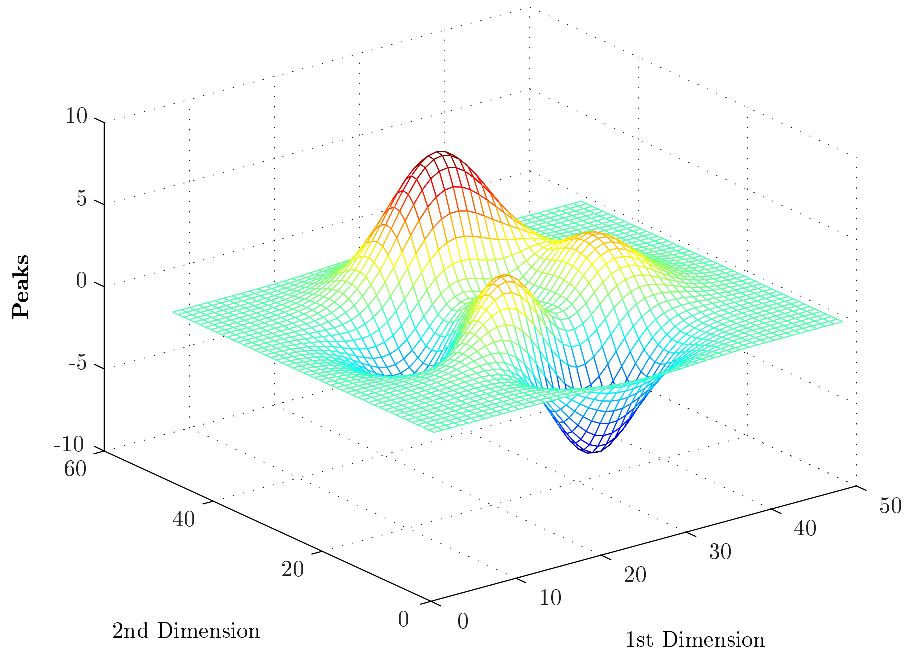
8.2 Resized figure

```
x=linspace(0,10*pi); y=sin(x);
plot(x,y)
pf=get(1,'position');
set(1,'Position',[pf(1) pf(2) 800 200])
axis([0 10*pi -1.2 1.2])
set(gca,'ytick',[-1 0 0.5 1]);
laprint(1,'ex_ratio','options','factory','width',5,'asonscreen','on')
```



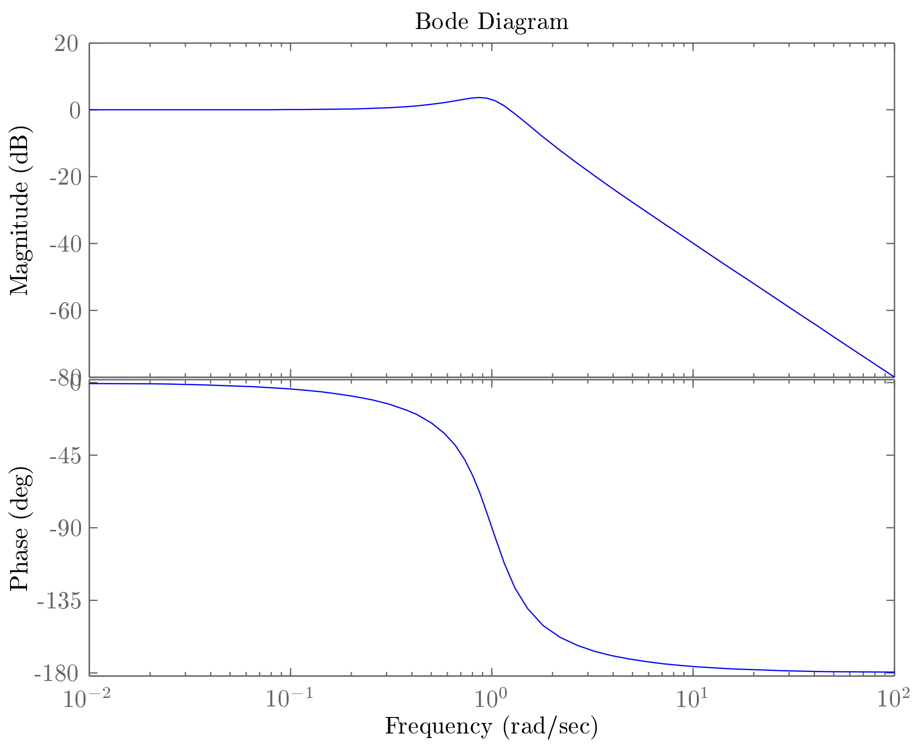
8.3 3D-plots

```
mesh(peaks)
xlabel('1st Dimension'),
ylabel('2nd Dimension'),
zlabel('\textbf{Peaks}'),
laprint(1,'ex_mesh','options','factory')
```



8.4 Bode from the Control System Toolbox

```
bode(tf(1,[1 0.7 1]))
laprint(1,'ex_bode','options','factory','figcopy','off')
```



8.5 Aligned figures using `-loose`

```
w=linspace(0,2*pi); s=sin(w); c=cos(w);
figure(1), clf, plot(s,c)
figure(2), clf, plot(s,c), axis equal, axis([-1 1 -1 1])
printcmd = ['print(''-f<figurenumber>'',''-depsc'','...
       '''-painters'',''-loose'',''<filename.eps>'')'];
laprint(1,'ex_align1','options','factory','width',7,'printcmd',printcmd)
laprint(2,'ex_align2','options','factory','width',7,'printcmd',printcmd)
```