A plot package for LaTeX 2$_\varepsilon$*

Jose-Emilio Vila-Forcen
jemilio@gmail.com

2009/08/26

1 Introduction

The functions described in the present document have been created to help in the drawing of plots from Matlab to LaTeX. The overall objective is to create a easy and fast framework to create plots with the same style, proper fonts, and just enough parameters.

2 Known problems

- Sometimes a combination of numbers in the dimensions of the plot might cause an error in latex. That is because the combination of numbers has caused some kind of overflow. The source of the problem might be in the makeplot package or in the pstricks family (supposed to be solved in v2.64 of pstricks-add). Send an email to the package author with all the data and hopefully you will receive an answer with an update of the package: the workaround solution is to use a different number, a close one to the original numbe might work.

3 Description

The package Startplot is composed of several functions. Due to the author's limited knowledge of TeX, it is done following LaTeX macros and the robustness of the package is not fully demonstrated.

All positions, total size of the figure and so on are calculated automatically using package FP.

3.1 Definition of an especific style

`\MPbg` Some functions are provided to change the style of the plots: Execute `\MPbg` it in

*This file has version number v1.0.7, last revised 2009/08/26.

order to use black lines with different styles already predefined. It is by default

`\MPcolor` Execute `\MPcolor` it in order to use solid color lines already predefined. It can be selected using the `color` package option or executing the command.

`colorLine[ABCDEFG]` If you want to change the colors of the lines, you may change one by one redefining the colors. For example:

```
\definecolor{colorLineA}{rgb}{1,1,1}
```

`\styleofLine[ABCDEFG]` It is possible to modify the seven predefined styles one by one: simply recreate a new definition for the desired one(s).

`Font commands` The commands `\fontaxeX`, `\fontaxeY`, `\fonttitleX`, `\fonttitleY` and `\fontlegend` contains the definitions of the fonts used in different parts of the legend: axes, labels (or titles) and legend. You may change them as, for example:

```
\def\fontaxeY{\small}
```

`\defaultOptionsMakeplot` The `\defaultOptionsMakeplot` command is executed at the beginning of the definition of the plot. redefine it to your desired code if you need to execute something more for each picture.

3.2 Create the plot

The plot is composed of several orders consecutive. Besides the order described here, a lot of them exist in the file which obtain the same result, but using more input parameters. The ones that I describe here are enough for the most of the cases.

`makeplot` The `makeplot` environment is the starting point to create the plot. The command line is as follows:

`\begin{makeplot}[`⟨*keys*⟩`]{`⟨*label in Y*⟩`}{`⟨*label in X*⟩`}` ..... `\end{makeplot}`

The optional arguments are the keys from PsTricks, including the ones for `\psaxes` function. Also, some new keys are added:

Dx Distance between labels in X

Dy Distance between labels in Y

width Width of the plot in mm

heightFactor Factor of height length respect to the width

startX Minimum value in the axis X

startY Minimum value in the axis Y

endX Maximum value in the axis X

endY Maximum value in the axis Y

factorBoundaryX Separation to the border

factorBoundaryY Separation to the border

captionX Separation to the caption in X, over 100

captionY Separation to the caption in Y, over 100

changeEndXpos If it is needed to change the end of the axis X, because of external legend (yes or no)

changeEndXsize How many mm more than the estandard

tickyplot Special ticks (all, x, y, none)

drawmargins Draw margins in this figure (there exist an option in the package to do it globally)

orgX and orgY Not working properly, intended to change the origin of the graphic

For example, a typical configuration for semilog case it is to use the options:

ylogBase=10, logLines=y, subticks=10, xsubticks=1

and LogLines can take the values x, y or all.

`\makeplot` The `\makeplot` macro works as the environment, but once the plot is finished it is needed to write `\end{pspicture}`

3.3 Plotting the data from files

`\plotFile` The `\plotFile[`⟨*keys*⟩`]{`⟨*file*⟩`}` macro takes the data from a file and plot it using the existing properties for lines in PsTricks taking into account the optional keys given to the command.

`\plotFile[ABCDEFG]` These macros have the same configuration as `\plotFile`, but each one uses the line style predefined. It accept the optinal arguments plotNoMax and PlotNo to say how many different plots are in the file and which one is intented to be plot, and nStep, xStart, xEnd, yStart and yEnd in order to select which values of the file should be plot.

3.4 The legend

The `makeplot` package can do very nice legends and without effort. There exists the next keys to control the apearance of the legend:

fillcolorWhiteBG Select the color name for the background of the legend

fillstyleWhiteBG Select the fillstyle for the background of the legend

framearcWhiteBG dimension of the arc of the legend box

linewidthWhiteBG linewidth of the legend box

linestyleWhiteBG linestyle of the legend box

linecolorWhiteBG color of the line of the legend box

and

legendSep Separation between legends in vertical

lineLength Length of the line in the legend area

lineTextSep Separation between the line and the text

borderlineSep Separation from the border of the box to the line

The main commands are as follows:

`\legendXY` The `\legendXY` macro usage is as follows:

`\legendXY[`⟨*keys*⟩`](`⟨*coordinate X,coordinateY*⟩`){`⟨*number of lines to be in the legend*⟩`} {`⟨*width in mm*⟩`}`

`\legend[DL,DR,UL,UR]` These macros avoid the needed to specify a coordinate and draw the legend in the position down-left, down-right, up-left or up-right of the plot. Therefore, the syntax is:

`\legendDL[`⟨*keys*⟩`]{`⟨*number of lines to be in the legend*⟩`} {`⟨*width in mm*⟩`}`

There exist some different ways to write the legends into the box. The easiest one is to use the next commands:

`\legend[ABCDEFG]f` These macros write the legends into the box in the order of the tex source. Each one has a different line style as it was for the plot. The syntax is:

`\legendCf[`⟨*keys*⟩`]{`⟨*text*⟩`}`

where the keys might modify the line style.

`\legendText` The `\legendText` macro includes some text in the legend without a line:

`\legendText[`⟨*keys*⟩`]{`⟨*text*⟩`}`

where the keys are not used, but kept in the definition of the macro for paralelism.

4 Record proper Matlab data

Matlab is one of the most famous data processing software in the research community. Because of its simplicity, the generation of the needed files can be done easily after their generation.

Each plot should be saved in a separate file in ASCII format. Thus, follow the next process for each one of the lines to plot:

1. Create a matrix Nx2, where N denotes the dimensionality of the data. Put in the first column the data of the X-axis, and in the second one the values in the Y-axis corresponding to the previous X one, let's call this matrix *values*;
2. Save *values* in a file as follows: `save file.mat values -ascii`.

Remarks

- Editing in a text editor should be possible, and numbers should be visible in ASCII format;
- The data should be real;
- It is possible to comment values in the file with `%`;
- It is possible to modify manually the values in the file.

5 Create EPS files

It is possible to create EPS files and use them directly in latex. The benefits are faster compilation and it is always good to have them in separate files, for example for distribution.

In order to create EPS files, follow the procedure:

1. Install perl from www.perl.com;
2. Copy bbox.exe from ps2eps.zip file to `c:\texmf\miktex\bin`. ps2eps.zip can be found at `www.telematik.informatik.uni-karlsruhe.de/~bless/ps2eps.html`;
3. `SET PATHEXT=.pl;%PATHEXT%` or use the start, settings, control panel, system, advanced tab, environment variables and edit the PATHEXT entry accordingly;
4. Rename ps2eps to ps2eps.pl and copy it to `c:\texmf\miktex\bin`.

Now, we create a batch file to help in the compilation of each time: Create a bat file something like: `makefigure.bat`, save it in `c:\texmf\miktex\bin`. It should contain the following lines:

```
echo off
del figures/%1.eps
latex %1eps.tex
dvips -t a4 -P pdf %1eps.dvi
ps2eps.pl %1eps.ps %1eps.eps
del %1eps.ps
del %1eps.eps
mkdir figures
rename %1eps.eps.eps %1.eps
move %1.eps figures
```

It is possible to change the contain above for different file names. In my case, I am creating files are follows:

```
\input{headers.tex}
```

```
\begin{document}
\pagestyle{empty}
\input{fig1.tex}
\end{document}
```

with the name `NAMEeps.tex` where `NAME` can be changed to the desired name. The file `headers.tex` contains all the headers of my report and therefore all proper fonts and so on will be used in this file and in the general one.

To create this file, it is possible to make another batch file called `fileForEPS.bat` such that:

```
echo off
echo \input{headers.tex} \begin{document} \emph{(continue in the same line)}
     \pagestyle{empty} \input{%1.tex} \end{document} > %1eps.tex
```

Execute `makefigure NAME` in the proper directory, and the EPS picture file will be created in the directory `figures/`. If you need to create the additional tex file, execute `fileForEPS NAME` and it will be done.

The procedure described above allows to use the following code in the main file:

```
\begin{figure}
\centering
%\input{fig.tex}
\includegraphics{figure/fig.eps}
\caption{Caption of the figure}
\label{fig:fig}
\end{figure}
```

where the commented line will describe which file is used: it is possible to choose between the tex file or (once picture is fixed and in order to improve speed) the eps one.

6 An example file

```
⟨*mptest⟩
\documentclass[]{article}
\usepackage[color]{makeplot}
\begin{document}
\begin{figure}
\centering

\begin{makeplot}[startX=-10, endX=5, startY=-1, endY=0, Dx = 5,%
   width=40, heightFactor=1, %
   ylogBase=10, logLines=y, subticks=10, xsubticks=1]%
   {$P_e$}{WNR, [dB]}
\plotFileA{data1.mat}
\plotFileB{data2.mat}
\legendDL{24.5}{2}
\legendAf{UDQ-QIM}
\legendBf{UQ-QIM}
\end{makeplot}

\caption{Performance analysis result.}
\label{fig:results}
\end{figure}

\end{document}
⟨/mptest⟩
```

6.1 Data files:

data1.mat

```
⟨*data1⟩
 -1.0000000e+001 -3.1186120e-001
 -9.5000000e+000 -3.1382220e-001
 -9.0000000e+000 -3.1610308e-001
 -8.5000000e+000 -3.1874224e-001
 -8.0000000e+000 -3.2177814e-001
 -7.5000000e+000 -3.2524775e-001
 -7.0000000e+000 -3.2918480e-001
 -6.5000000e+000 -3.3361772e-001
 -6.0000000e+000 -3.3856759e-001
 -5.5000000e+000 -3.4404614e-001
 -5.0000000e+000 -3.5005399e-001
 -4.5000000e+000 -3.5657933e-001
 -4.0000000e+000 -3.6359717e-001
 -3.5000000e+000 -3.7106902e-001
 -3.0000000e+000 -3.7894319e-001
 -2.5000000e+000 -3.8715537e-001
 -2.0000000e+000 -3.9562939e-001
 -1.5000000e+000 -4.0427825e-001
 -1.0000000e+000 -4.1300527e-001
```

```
 -5.0000000e-001 -4.2170561e-001
  0.0000000e+000 -4.3026847e-001
  5.0000000e-001 -4.5284882e-001
  1.0000000e+000 -4.8524584e-001
  1.5000000e+000 -5.2154969e-001
  2.0000000e+000 -5.6193985e-001
  2.5000000e+000 -6.0664420e-001
  3.0000000e+000 -6.5595407e-001
  3.5000000e+000 -7.1023618e-001
  4.0000000e+000 -7.6994081e-001
  4.5000000e+000 -8.3560655e-001
  5.0000000e+000 -9.0786295e-001
⟨/data1⟩
```

data2.mat

```
⟨*data2⟩
 -1.0000000e+001 -3.0170171e-001
 -9.5000000e+000 -3.0182275e-001
 -9.0000000e+000 -3.0197731e-001
 -8.5000000e+000 -3.0217679e-001
 -8.0000000e+000 -3.0243768e-001
 -7.5000000e+000 -3.0278484e-001
 -7.0000000e+000 -3.0325664e-001
 -6.5000000e+000 -3.0391260e-001
 -6.0000000e+000 -3.0484295e-001
 -5.5000000e+000 -3.0617875e-001
 -5.0000000e+000 -3.0809964e-001
 -4.5000000e+000 -3.1083660e-001
 -4.0000000e+000 -3.1466753e-001
 -3.5000000e+000 -3.1990577e-001
 -3.0000000e+000 -3.2688343e-001
 -2.5000000e+000 -3.3593295e-001
 -2.0000000e+000 -3.4737023e-001
 -1.5000000e+000 -3.6148225e-001
 -1.0000000e+000 -3.7852034e-001
 -5.0000000e-001 -3.9869936e-001
  0.0000000e+000 -4.2220206e-001
  5.0000000e-001 -4.4918738e-001
  1.0000000e+000 -4.7980138e-001
  1.5000000e+000 -5.1418989e-001
  2.0000000e+000 -5.5251165e-001
  2.5000000e+000 -5.9495142e-001
  3.0000000e+000 -6.4173231e-001
  3.5000000e+000 -6.9312654e-001
  4.0000000e+000 -7.4946429e-001
  4.5000000e+000 -8.1114012e-001
  5.0000000e+000 -8.7861718e-001
⟨/data2⟩
```

7 The implementation

⟨∗makeplot⟩

7.1 Headers

First we test that we got the right format and name the package, and load external packages.

```
\NeedsTeXFormat{LaTeX2e}%
\ProvidesPackage{makeplot}[2009/08/26 %
                           v1.0.7 %
                           Plots utility from Jose-Emilio Vila-Forcen]%
\RequirePackage[nomessages]{fp}%
\RequirePackage{pst-plot}%
\RequirePackage{pstricks-add}%
\RequirePackage{xkeyval}%
```

`color` The option `color` leads to a redefinition of the line styles of the plot and uses colors with solid lines. Otherwise, black lines with different line styles are used

```
\newif\ifMP@color%
\MP@colorfalse%
\DeclareOption{color}{%
  \MP@colortrue%
  \typeout{ ------}%
  \typeout{ ------   makePlot package working in color}%
  \typeout{ ------}}%
```

`drawmargins` The option `drawmargins` draw the margins of the figues. This option is useful in order to check if something is going out of the margins, since there is not internal checking for that.

```
\newif\ifMP@drawmargins%
\MP@drawmarginsfalse%
\DeclareOption{drawmargins}{%
  \MP@drawmarginstrue%
  \typeout{ ------}%
  \typeout{ ------   makePlot package drawing the margins}%
  \typeout{ ------}}%
```

`big` The option `big` put the lines of the plot larger

```
\newif\ifMP@big%
\MP@bigfalse%
\DeclareOption{big}{%
\MP@bigtrue
  \typeout{ ------}%
  \typeout{ ------   makePlot package in big size}%
  \typeout{ ------}}%
```

If an unknown option is used, a warning will be displayed.

```
\DeclareOption*{%
```

```
  \PackageWarning{startPlot}{%
    Unknown option '\CurrentOption'}}%
```

Now we process the options.

```
\ProcessOptions\relax%
```

7.2 Commands to fix the style

`\MPbg` Command to use the styles of the lines in black and white, by default:

```
\newcommand{\MPbg}{%
 \def\styleoflineA{%
  \psset{linestyle=solid, dash=1pt 0pt 0pt 0pt, dotsep=0pt,%
     linecolor=black}}%
 \def\styleoflineB{%
  \psset{linestyle=dashed, dash=5pt 5pt 0pt 0pt, dotsep=0pt,%
     linecolor=black}}%
 \def\styleoflineC{%
  \psset{linestyle=dotted, dash=3pt 2pt 0pt 0pt, dotsep=1pt,%
     linecolor=black}}%
 \def\styleoflineD{%
  \psset{linestyle=dashed, dash=4pt 1.5pt 1pt 1.5pt, dotsep=3pt,%
     linecolor=black}}%
 \def\styleoflineE{%
  \psset{linestyle=dotted, dash=1pt 4pt 0pt 0pt, dotsep=3pt,%
     linecolor=black}}%
 \def\styleoflineF{%
  \psset{linestyle=dashed, dash=6pt 1.5pt 3pt 1.5pt, dotsep=3pt,%
     linecolor=black}}%
 \def\styleoflineG{%
  \psset{linestyle=dashed, dash=2pt 6pt 0pt 0pt, dotsep=0pt,%
     linecolor=black}}}%
```

`MPcolor` Definition of the standard colors for the lines, using the color option of the package. They might be changed by the user:

```
\definecolor{colorLineA}{rgb}{0,0,1}%
\definecolor{colorLineB}{rgb}{1,0,0}%
\definecolor{colorLineC}{rgb}{0,1,0}%
\definecolor{colorLineD}{rgb}{0,1,1}%
\definecolor{colorLineE}{rgb}{1,0,1}%
\definecolor{colorLineF}{rgb}{1,1,0}%
\definecolor{colorLineG}{rgb}{0.5,0.5,0.5}%
```

Command to set the styles of the lines using colors:

```
\newcommand{\MPcolor}{%
 \def\styleoflineA{\psset{linestyle=solid, linecolor=colorLineA}}%
 \def\styleoflineB{\psset{linestyle=solid, linecolor=colorLineB}}%
 \def\styleoflineC{\psset{linestyle=solid, linecolor=colorLineC}}%
 \def\styleoflineD{\psset{linestyle=solid, linecolor=colorLineD}}%
 \def\styleoflineE{\psset{linestyle=solid, linecolor=colorLineE}}%
 \def\styleoflineF{\psset{linestyle=solid, linecolor=colorLineF}}%
```

```
\def\styleoflineG{\psset{linestyle=solid, linecolor=colorLineG}}%
```

According to the package option, one of the styles will be loaded:

```
\ifMP@color%
 \MPcolor%
\else%
 \MPbg%
\fi%
```

```
\makeatletter%
```

`Font sizes` Definition of the standard font sizes in the plot. It is possible to redefine them.

```
\def\fontaxeY{\normalsize}%
\def\fontaxeX{\normalsize}%
\def\fonttitleY{\normalsize}%
\def\fonttitleX{\normalsize}%
\def\fontlegend{\footnotesize}%
```

7.3 Main macros: makeplot

Options for the command `makeplot` and the addition of the family of keys.

```
\define@key[psset]{makeplot}{Dx}[1]%
   {\def\gridDxmakeplot{#1}}%
\define@key[psset]{makeplot}{Dy}[1]%
   {\def\gridDymakeplot{#1}}%
\define@key[psset]{makeplot}{width}[50]%
   {\def\widthPlotmakeplot{#1}}%
\define@key[psset]{makeplot}{heightFactor}[1]%
   {\def\heightPlotFactormakeplot{#1}}%
\define@key[psset]{makeplot}{startX}[0]%
   {\def\xamakeplot{#1}}%
\define@key[psset]{makeplot}{startY}[0]%
   {\def\yamakeplot{#1}}%
\define@key[psset]{makeplot}{endX}[1]%
   {\def\xzmakeplot{#1}}%
\define@key[psset]{makeplot}{endY}[1]%
   {\FPmul\yzmakeplot{#1}{1.0001}}%
\define@key[psset]{makeplot}{factorBoundaryX}%
   {\FPdiv\factorBOUNDARYxmakeplot{#1}{1.5}}%
\define@key[psset]{makeplot}{factorBoundaryY}%
   {\FPdiv\factorBOUNDARYymakeplot{#1}{1.5}}%
\define@key[psset]{makeplot}{captionY}%
   {\FPdiv\factorXmakeplot{#1}{8}}%
\define@key[psset]{makeplot}{captionX}%
   {\FPdiv\factorYmakeplot{#1}{11}}%
\define@choicekey[psset]{makeplot}{changeEndXpos}[\var\nr]{no,yes}%
   {\def\endXamakeplot{\nr}}%
\define@key[psset]{makeplot}{changeEndXsize}%
   {\def\endXbmakeplot{#1}}%
```

```
\define@choicekey[psset]{makeplot}{tickyplot}[\val\nr]{all,x,y,none}%
   {\def\tickymakeplot{\nr}} %
\define@choicekey[psset]{makeplot}{drawmargins}[\vall\nrr]{yes,no}%
   {\def\drawmarginsmakeplot{\nrr}} %
\define@key[psset]{makeplot}{orgX}%
   {\def\orgXmakeplot{#1}} %
\define@key[psset]{makeplot}{orgY}%
   {\def\orgYmakeplot{#1}} %
\pst@addfams{makeplot}%
```

Default values for the above parameters

```
\psset{Dx=1, Dy=1,
width=50, heightFactor=1, startX=0, endX=1, startY=0, endY=1,
showorigin=true, axesstyle=frame, ticks=all, labels=all,
factorBoundaryX=1, factorBoundaryY=1, captionY=100, captionX=100,
xsubticks=0, subticksize=1, subtickcolor=black}%
\ifMP@big%
 \psset{tickwidth=0.5pt, subtickwidth=0.5pt, linewidth=1pt}%
\else%
 \psset{tickwidth=0.2pt, subtickwidth=0.2pt, linewidth=0.5pt}%
\fi%
\psset{tickyplot=all}%
\psset{changeEndXsize=0, changeEndXpos=no}%
\ifMP@drawmargins%
 \psset{drawmargins=yes}%
\else%
 \psset{drawmargins=no}%
\fi%
\psset{orgX=314, orgY=314}%
```

`makeplot` The `makeplot` environemnt perfoms most of the tasks needed for the plot. The axes are drawn, the legends and everything related with the background of the plot besides the legend, that will arrive later.

```
\newenvironment{makeplot}[3][]%
  {\makeplot[#1]{#2}{#3}}%
  {\end{pspicture}}%
```

`\defaultOptionsMakeplot` Default values, it is possible to change this command as desired to include some source to be executed in each `makeplot` command or enviroment call

```
\def\defaultOptionsMakeplot{}%
```

`\makeplot` This is the most important macro, it can be used instead of the makeplot enviroment just adding `\end{pspicture}` at the end of the plot.

```
\def\makeplot{\@ifnextchar[\@makeplot{\@makeplot[]}}%
\def\@makeplot[#1]#2#3{%
  % Use the default options command and process
  % the options of the command line
  \defaultOptionsMakeplot%
  \psset{#1}%
```

```
% Calculate the dimensions of the plot
\def\gridDx{\gridDxmakeplot}%
\def\gridDy{\gridDymakeplot}%
\def\widthPlot{\widthPlotmakeplot}%
\def\heightPlotFactor{\heightPlotFactormakeplot}%
\def\xa{\xamakeplot}%
\def\ya{\yamakeplot}%
\def\xz{\xzmakeplot}%
\def\yz{\yzmakeplot}%
\def\factorBOUNDARYx{\factorBOUNDARYxmakeplot}%
\def\factorBOUNDARYy{\factorBOUNDARYymakeplot}%
\def\factorX{\factorXmakeplot}%
\def\factorY{\factorYmakeplot}%
\def\endXa{\endXamakeplot}%
\def\endXb{\endXbmakeplot}%
\def\ticky{\tickymakeplot}%
\def\drawmargins{\drawmarginsmakeplot}%
\def\orgX{\orgXmakeplot}%
\def\orgY{\orgYmakeplot}%
%
% Use the fonts established
\def\pshlabel{\fontaxeX}%
\def\psvlabel{\fontaxeY}%
\def\pshlabel{\fontaxeX}%
\def\psvlabel{\fontaxeY}%
%
% Calculate the proper units for the plot
\FPadd\xDiff\xz{-\xa}%
\FPdiv\unitX\widthPlot\xDiff%
\FPadd\yDiff\yz{-\ya}%
\FPmul\ff\widthPlot\heightPlotFactor%
\FPdiv\unitY\ff\yDiff%
\psset{xunit=\unitX mm,yunit=\unitY mm}%
%
\FPadd\yinc\yz{-\ya}%
\FPdiv\ymid\yinc{2}%
\FPadd\ymid\ymid\ya%
\FPadd\xinc\xz{-\xa}%
\FPdiv\xmid\xinc{2}%
\FPadd\xmid\xmid\xa%
\FPadd\xzz\xz{0}%
%
% Related to the position of the legends
\FPdiv\factorX\factorX\unitX%
\FPadd\xaa\xa{-\factorX}%
\FPdiv\factorY\factorY\unitY%
\FPadd\yaa\ya{-\factorY}%
%
% Related to the size of the plot
\FPdiv\factorBOUNDARYx\factorBOUNDARYx\unitX%
```

```
\FPadd\xaaa\xaa{-\factorBOUNDARYx}%
\FPdiv\factorBOUNDARYy\factorBOUNDARYy\unitY%
\FPadd\yaaa\yaa{-\factorBOUNDARYy}%
%
% Change of the size of the plot
% if it is asked in the options,
% like putting the legend at the right
\ifnum \endXb>0
  \FPdiv\w{\endXb}\unitX%
  \ifnum \endXa=0
    \def\endXa{\xz}%
  \fi%
  \FPadd\ww{\endXa}{-\xz}%
  \def\sep{0.5}%
  \FPdiv\www\sep\unitX%
  \FPadd\xzz\xz\w%
  \FPadd\xzz\xzz\ww%
  \FPadd\xzz\xzz\www %
\fi%
%
% Starting the pspicture environment
\begin{pspicture}(\xaaa,\yaaa)(\xzz,\yz)%
%
% Draw a box with the margins if requested
\ifnum \drawmargins=0
  \psframe(\xaaa,\yaaa)(\xzz,\yz)%
\fi%
%
\FPadd\tickHerex{\yz}{-\ya}%
\FPmul\tickHerex\tickHerex\unitY%
\FPadd\tickHerey{\xz}{-\xa}%
\FPmul\tickHerey\tickHerey\unitX%
%
% To change the origin of the plot
% not working yet
\def\xaOrigin{\xa}%
\def\yaOrigin{\ya}%
\ifnum \orgX=314
\else%
  \def\xaOrigin{\orgX}%
\fi%
\ifnum \orgY=314
\else%
  \def\yaOrigin{\orgY}%
\fi %
%
% Put the ticks as given in the options
\ifnum \ticky=0
  \psaxes[Ox=\xa,Oy=\ya,Dx=\gridDx,Dy=\gridDy, #1,%
      ticksize=-4pt 4pt, subticks=0, subticksize=0,%
```

```
        tickwidth=0.5pt, linewidth=0pt,%
        axesstyle=axes, linecolor=white, #1, ticks=all, labels=none]%
        {-}(\xa,\ya)(\xa,\ya)(\xz,\yz)%
  \fi%
  \ifnum \ticky=1
    \psaxes[Ox=\xa,Oy=\ya,Dx=\gridDx,Dy=\gridDy, #1,%
        ticksize=-4pt 4pt, subticks=0, subticksize=0,%
        tickwidth=0.5pt, linewidth=0pt,%
        axesstyle=axes, linecolor=white, #1, ticks=x, labels=none]%
        {-}(\xa,\ya)(\xa,\ya)(\xz,\yz)%
  \fi%
  \ifnum \ticky=2
    \psaxes[Ox=\xa,Oy=\ya,Dx=\gridDx,Dy=\gridDy, #1,%
        ticksize=-4pt 4pt, subticks=0, subticksize=0,%
        tickwidth=0.5pt, linewidth=0pt,%
        axesstyle=axes, linecolor=white, #1, ticks=y, labels=none]%
        {-}(\xa,\ya)(\xa,\ya)(\xz,\yz)%
  \fi%
  \ifnum \ticky=3
  \fi%
  %
  % Put the main axes and the grid
  \FPround\tickHerex{\tickHerex}{10}%
  \FPround\tickHerey{\tickHerey}{10}%
  \psaxes[xticksize=0mm \tickHerex mm, yticksize=0mm \tickHerey mm,%
    Ox=\xa,Oy=\ya,Dx=\gridDx,Dy=\gridDy, #1]%
    {-}(\xaOrigin,\yaOrigin)(\xa,\ya)(\xz,\yz)%
\def\MP@xa{\xaOrigin}%
\def\MP@ya{\yaOrigin}%
\def\MP@xz{\xz}%
\def\MP@yz{\yz}%
  %
  % Write the legend of the plot
  \rput(\xaa,\ymid){\rotateleft{\fonttitleY #2}}%
  \rput(\xmid,\yaa){\fonttitleX #3}%
  \ifMP@big%
   \psset{linewidth=2pt}%
  \else%
   \psset{linewidth=1pt}%
  \fi%
}%
```

7.4 How to plot the data of a file

`\plotFile` Function to plot a file with the linestyle given

```
\def\plotFile{\@ifnextchar[\@plotFile{\@plotFile[]}}%
\def\@plotFile[#1]#2{%
  \readdata[#1]{\data}{#2}\listplot[#1]{\data}}%
```

And now, a specific function for each linestyle predefined:

\plotFileA

```
\def\plotFileA{\@ifnextchar[\@plotFileA{\@plotFileA[]}}%
\def\@plotFileA[#1]#2{%
  \styleoflineA%
  \plotFile[#1]{#2}}%
```

\plotFileB

```
\def\plotFileB{\@ifnextchar[\@plotFileB{\@plotFileB[]}}%
\def\@plotFileB[#1]#2{%
  \styleoflineB%
  \plotFile[#1]{#2}}%
```

\plotFileC

```
\def\plotFileC{\@ifnextchar[\@plotFileC{\@plotFileC[]}}%
\def\@plotFileC[#1]#2{%
  \styleoflineC%
  \plotFile[#1]{#2}}%
```

\plotFileD

```
\def\plotFileD{\@ifnextchar[\@plotFileD{\@plotFileD[]}}%
\def\@plotFileD[#1]#2{%
  \styleoflineD%
  \plotFile[#1]{#2}}%
```

\plotFileE

```
\def\plotFileE{\@ifnextchar[\@plotFileE{\@plotFileE[]}}%
\def\@plotFileE[#1]#2{%
  \styleoflineE%
  \plotFile[#1]{#2}}%
```

\plotFileF

```
\def\plotFileF{\@ifnextchar[\@plotFileF{\@plotFileF[]}}%
\def\@plotFileF[#1]#2{%
  \styleoflineF%
  \plotFile[#1]{#2}}%
```

\plotFileG

```
\def\plotFileG{\@ifnextchar[\@plotFileG{\@plotFileG[]}}%
\def\@plotFileG[#1]#2{%
  \styleoflineG%
  \plotFile[#1]{#2}}%
```

7.5 Legends code

\legend Defines the general **\legend** command, the inputs are: the optional parameter with PSTricks keys, the position in X, the length of the line, the separation between line and text, the position in Y and the text of the legend.

```
\def\legend{\@ifnextchar[\@legend{\@legendf[]}}%
```

```
\def\@legend[#1]#2#3#4#5#6{%
  \def\xb{#2}%
  \def\x{#3}%
  \FPdiv\x\x\unitX%
  \FPadd\xbb\xb\x%
  \def\x{#4}%
  \FPdiv\x\x\unitX%
  \FPadd\xc\xbb\x%
  %
  \def\y{#5}%
  \psline[#1]{-}(\xb,\y)(\xbb,\y)%
  \rput(\xc,\y){\makebox[0 cm][l]{{\fontlegend #6}}}}%
```

`\legend[ABCDEFG]` One command for each line style: as the previous command, but now using the predefined styles

```
\def\legendA{\@ifnextchar[\@legendA{\@legendA[]}}%
\def\@legendA[#1]#2#3#4#5{%
  \styleoflineA%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendB{\@ifnextchar[\@legendB{\@legendB[]}}%
\def\@legendB[#1]#2#3#4#5{%
  \styleoflineB%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendC{\@ifnextchar[\@legendC{\@legendC[]}}%
\def\@legendC[#1]#2#3#4#5{%
  \styleoflineC%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendD{\@ifnextchar[\@legendD{\@legendD[]}}%
\def\@legendD[#1]#2#3#4#5{%
  \styleoflineD%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendE{\@ifnextchar[\@legendE{\@legendE[]}}%
\def\@legendE[#1]#2#3#4#5{%
  \styleoflineE%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendF{\@ifnextchar[\@legendF{\@legendF[]}}%
\def\@legendF[#1]#2#3#4#5{%
  \styleoflineF%
  \legend[#1]{#2}{#3}{#4}{#5}}%
\def\legendG{\@ifnextchar[\@legendG{\@legendG[]}}%
\def\@legendG[#1]#2#3#4#5{%
  \styleoflineG%
  \legend[#1]{#2}{#3}{#4}{#5}}%
```

`\legend[ABCDEFG]f` An automatic legend mode, where the position is automatically calculated for each one in consecutive order

```
\def\legendAf{\@ifnextchar[\@legendAf{\@legendAf[]}}%
\def\@legendAf[#1]#2{%
  \legendA[#1]{\posx}{\l}{\s}{\posy}{#2}%
```

```
  \FPadd\posy\posy{-\dif}}%
\def\legendBf{\@ifnextchar[\@legendBf{\@legendBf[]}}%
\def\@legendBf[#1]#2{%
  \legendB[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
\def\legendCf{\@ifnextchar[\@legendCf{\@legendCf[]}}%
\def\@legendCf[#1]#2{%
  \legendC[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
\def\legendDf{\@ifnextchar[\@legendDf{\@legendDf[]}}%
\def\@legendDf[#1]#2{%
  \legendD[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
\def\legendEf{\@ifnextchar[\@legendEf{\@legendEf[]}}%
\def\@legendEf[#1]#2{%
  \legendE[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
\def\legendFf{\@ifnextchar[\@legendFf{\@legendFf[]}}%
\def\@legendFf[#1]#2{%
  \legendF[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
\def\legendGf{\@ifnextchar[\@legendGf{\@legendGf[]}}%
\def\@legendGf[#1]#2{%
  \legendG[#1]{\posx}{\l}{\s}{\posy}{#2}%
  \FPadd\posy\posy{-\dif}}%
```

`\legendText` A legend without a line: to include some text in the legend box

```
\def\legendText{\@ifnextchar[\@legendText{\@legendText[]}}%
\def\@legendText[#1]#2{%
  \rput(\posx,\posy){%
    \makebox[0 cm][l]{%
    \hspace{-0.1cm}{\footnotesize #2}}}%
  \FPadd\posy\posy{-\dif}}%
```

Keys to be used in the legend environment:

```
\define@key[psset]{whiteBG}{fillcolorWhiteBG}{\def\fillcolorWhiteBG{#1}}%
\define@key[psset]{whiteBG}{fillstyleWhiteBG}{\def\fillstyleWhiteBG{#1}}%
\define@key[psset]{whiteBG}{framearcWhiteBG}{\def\framearcWhiteBG{#1}}%
\define@key[psset]{whiteBG}{linewidthWhiteBG}{\def\linewidthWhiteBG{#1}}%
\define@key[psset]{whiteBG}{linestyleWhiteBG}{\def\linestyleWhiteBG{#1}}%
\define@key[psset]{whiteBG}{linecolorWhiteBG}{\def\linecolorWhiteBG{#1}}%
\pst@addfams{whiteBG}%
```

Default values for the above keys:

```
\psset{fillcolorWhiteBG=white, fillstyleWhiteBG=solid,
  framearcWhiteBG=0.3, linewidthWhiteBG=0.01,
  linestyleWhiteBG=solid, linecolorWhiteBG=black}%
```

`\whiteBG` Create the box of the legend in the given position

```
\def\whiteBG{\@ifnextchar[\@whiteBG{\@whiteBG[]}}%
```

```
\def\@whiteBG[#1]#2#3#4#5{%
  \psframe[framearc=\framearcWhiteBG, fillcolor=\fillcolorWhiteBG,%
    fillstyle=\fillstyleWhiteBG, linewidth=\linewidthWhiteBG cm,%
    linestyle=\linestyleWhiteBG, linecolor=\linecolorWhiteBG, #1]%
    (#2,#3)(#4,#5)}%
```

Keys to control the line length and separation with the text in the legend:

```
\define@key[psset]{legendXY}{legendSep}{\def\sepYlegendXY{#1}} %
\define@key[psset]{legendXY}{lineLength}{\def\llegendXY{#1}} %
\define@key[psset]{legendXY}{lineTextSep}{\def\slegendXY{#1}} %
\define@key[psset]{legendXY}{borderlineSep}{\FPmul\sepTxlegendXY{#1}{2}} %
\pst@addfams{legendXY}%
```

Default values:

```
\psset{legendSep=4, lineLength=5, lineTextSep=1, borderlineSep=1}%
```

`\legendXY` Plot a legend box and calculate the parameters of the legend in the position given in round brackets, for `#4` different legends and with a width of `#5`.

```
\def\legendXY{\@ifnextchar[\@legendXY{\@legendXY[]}}%
\def\@legendXY[#1](#2,#3)#4#5{%
  \psset{#1}%
  \def\sepY{\sepYlegendXY}%
  \def\l{\llegendXY}%
  \def\s{\slegendXY}%
  \def\sepTx{\sepTxlegendXY}%
  \def\xaXY{#2}%
  \def\yzXY{#3}%
  \FPdiv\sep\sepY{8}%
  \FPdiv\sepYy\sepY{1.33} %
  \FPdiv\x\sep\unitX% To separate \sep mm the legend from the axe
  \FPdiv\y\sep\unitY% To separate \sep mm the legend from the axe
  \FPadd\leftX\xaXY{\x}%
  \FPadd\highY\yzXY{-\y}%
  \FPdiv\w{#4}\unitX%
  \FPadd\rightX\leftX{\w}%
  \FPdiv\sepTx\sepTx\unitX%
  \FPadd\posx\leftX\sepTx%
  \FPdiv\sepTy\sepYy\unitY%
  \FPadd\posy\highY{-\sepTy}%
  \FPdiv\dif{\sepY}\unitY% Separation of the legends
  \FPadd\n{#5}{-1}%
  \FPmul\lowY\dif\n%
  \FPadd\lowY\lowY\sepTy%
  \FPadd\lowY\posy{-\lowY}%
  \whiteBG[#1]{\leftX}{\lowY}{\rightX}{\highY}}%
```

`\legend[UL,UR,DL,DR]` Put the legend box in the positions upper-left, upper-right, down-left and down-right. The input parameters are the optional keys, the number of legends and the width.

```
\def\legendUL{\@ifnextchar[\@legendUL{\@legendUL[]}}%
```

```
\def\@legendUL[#1]#2#3{%
  \FPmul\xUL\MP@xa{1}%
  \FPmul\yUL\MP@yz{1}%
  \legendXY[#1](\xUL,\yUL){#2}{#3}}%
  %
  \def\legendUR{\@ifnextchar[\@legendUR{\@legendUR[]}}%
  \def\@legendUR[#1]#2#3{%
  \FPmul\xUR\MP@xz{1}%
  \FPmul\yUR\MP@yz{1}%
  \def\sep{0.5}%
  \FPdiv\xp\sep\unitX% To separate \sep mm the legend from the axe
  \FPmul\xp\xp{2}%
  \FPadd\xp\xUR{-\xp}%
  \FPdiv\xx{#2}\unitX%
  \FPadd\xp\xp{-\xx}%
  \legendXY[#1](\xp,\yUR){#2}{#3}}%
  %
\def\legendDL{\@ifnextchar[\@legendDL{\@legendDL[]}}%
\def\@legendDL[#1]#2#3{%
  \psset{#1}%
  \def\sepY{\sepYlegendXY}%
  \def\sepTx{\sepTxlegendXY}%
  \FPmul\xzDL\MP@xz{1}%
  \FPmul\xaDL\MP@xa{1}%
  \FPmul\yzDL\MP@yz{1}%
  \FPmul\yaDL\MP@ya{1}%
  \FPdiv\sep\sepY{8}%
  \FPdiv\sep\sepY{8}%
  \FPdiv\sepYy\sepY{1.33} %
  \FPdiv\x\sep\unitX% To separate \sep mm the legend from the axe
  \FPdiv\y\sep\unitY% To separate \sep mm the legend from the axe
  \FPadd\leftX\xaDL{0}%
  \FPdiv\w{#2}\unitX%
  \FPadd\rightX\leftX{\w}%
  \FPdiv\sepTx\sepTx\unitX%
  \FPadd\posx\leftX\sepTx%
  \FPadd\lowY\yaDL{\y}%
  \FPadd\lowY\lowY{\y}%
  \FPdiv\sepTy\sepYy\unitY%
  \FPadd\posy\lowY{\sepTy}%
  \FPdiv\dif\sepY\unitY% Separation of the legends
  \FPadd\n{#3}{-1}%
  \FPmul\highY\dif\n%
  \FPadd\highY\highY\sepTy%
  \FPadd\highY\posy{\highY}%
  \FPadd\posy\highY{-\sepTy}%
  \legendXY[#1](\leftX,\highY){#2}{#3}}%
  %
\def\legendDR{\@ifnextchar[\@legendDR{\@legendDR[]}}%
\def\@legendDR[#1]#2#3{%
```

```
  \psset{#1}%
  \def\sepY{\sepYlegendXY}%
  \def\sepTx{\sepTxlegendXY}%
  \FPmul\xzDR\MP@xz{1}%
  \FPmul\xaDR\MP@xa{1}%
  \FPmul\yzDR\MP@yz{1}%
  \FPmul\yaDR\MP@ya{1}%
  \FPdiv\sep\sepY{8}%
  \FPdiv\sepYy\sepY{1.33} %
  \FPdiv\x\sep\unitX% To separate \sep mm the legend from the axe
  \FPdiv\y\sep\unitY% To separate \sep mm the legend from the axe
  \FPadd\rightX\xzDR{-\x}%
  \FPdiv\w{#2}\unitX%
  \FPadd\leftX\rightX{-\w}%
  \FPadd\leftX\leftX{-\x}%
  \FPdiv\sepTx\sepTx\unitX%
  \FPadd\posx\leftX\sepTx%
  \FPadd\lowY\yaDR{\y}%
  \FPadd\lowY\lowY{\y}%
  \FPdiv\sepTy\sepYy\unitY%
  \FPadd\posy\lowY{\sepTy}%
  \FPdiv\dif{\sepY}\unitY% Separation of the legends
  \FPadd\n{#3}{-1}%
  \FPmul\highY\dif\n%
  \FPadd\highY\highY\sepTy%
  \FPadd\highY\posy{\highY}%
  \FPadd\posy\highY{-\sepTy}%
  \legendXY[#1](\leftX,\highY){#2}{#3}}%
```

This is the end

```
\makeatother%
```

⟨/makeplot⟩