MiniPlot – A Package For Easy Figure Arrangements

Tobias Wahl*

15th June 2001

Abstract

MiniPlot is a package to help LaTeX's user to typeset eps figures using an easy-to-use interface.

Figures can be arranged as one-figure-only or as a collection of figure in columns and rows which can contain itself sub-figures in columns and rows. Also wrapped figures are supported.

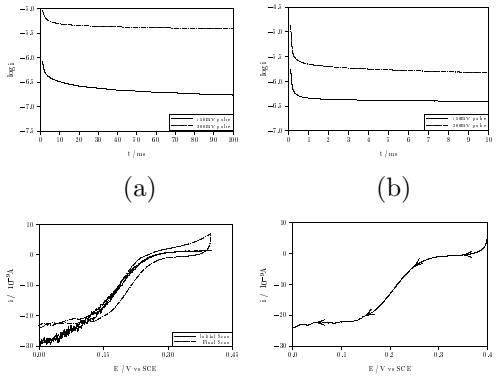
This package provides commands to display a framebox instead of the figure as the graphics package does already but additionally it writes useful information such as the label and scaling factor into these boxes.

Contents

*I'd like to thank all the people helping me in `comp.text.tex` for answering my dull questions.

1 Motivation

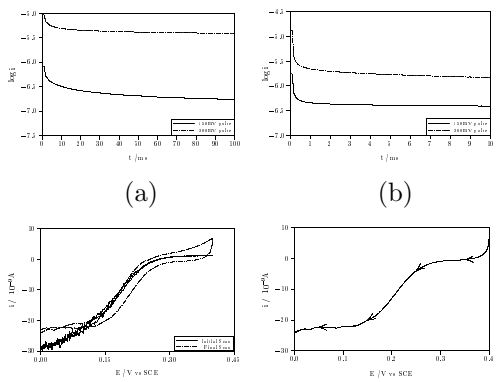
This section is only to show what MiniPlot is about. MiniPlot is a strong tool for plotting multiple figures and sub-figure as the following figures show. It's easy to use interface allows easy and fast access to professional looking figure presentation.



(a) (b)

(c) This is a very long sub-sub-caption (d)

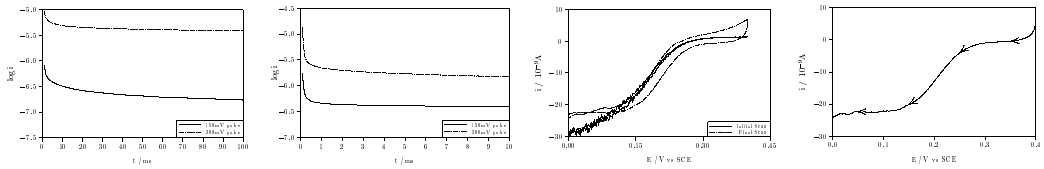
Figure 1: left sub-figure



(a) (b)

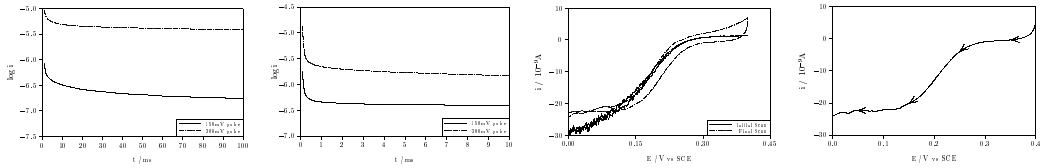
(c) (d)

Figure 2: right sub-figure



(a) (b) (c) (d)

Figure 3: top sub-figure



(a) (b) (c) (d)

Figure 4: bottom sub-figure

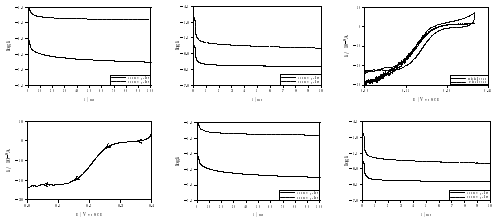


Figure 5: top left

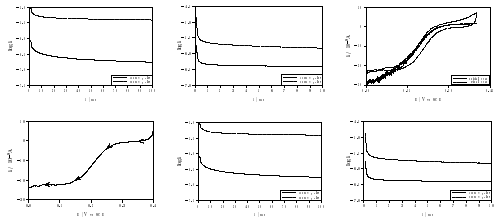


Figure 6: top right

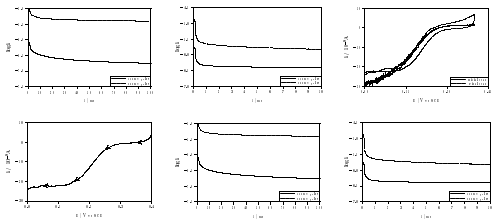


Figure 7: bottom

2 Loading the packages

In order to run MiniPlot the following lines are needed in the LaTeXfile

```
% needed packages
\usepackage{calc}
\usepackage{ifthen}
\usepackage{graphics}
\usepackage{epsfig}
\usepackage{miniplot}

% supplementary packages
\usepackage{wrapfig}
\usepackage{subfigure}
\usepackage{rotfloat}

% personal adjustments
\setcounter{plotFigures}{0}
\renewcommand{\floatpagefraction}{0.8}
```

The wrapfigure package is only needed when one uses the wrapfigure features of MiniPlot. The subfigure package is certainly something one might include since this gives access to the `arrangedFigure` environment, the most powerful feature of this package. Rotfloat, and rotating which is loaded by rotfloat, is needed when one likes to use the side-way-figure capabilities of MiniPlot. I recommend to install all packages.

Putting the line `setcounterplotFigures0` is not necessary but is recomended for editing a document with loads of figures or when one likes to obtain the label of figures for refences.

The last line is very much my own personal style and not needed for using MiniPlot.

3 Including single figures

The problem is how to arrange your eps figures efficiently in you document. The answer is given by MiniPlot. This section is included to provide an overview over the capabilities of MiniPlot.

We start with a simple implementation of a single eps figure as shown in Figure 8, the simple implementation is given by

```
 \includeEps{figures/1}{Simple figure included
with scaling factor equals one.}{singleSimple}{1}
```

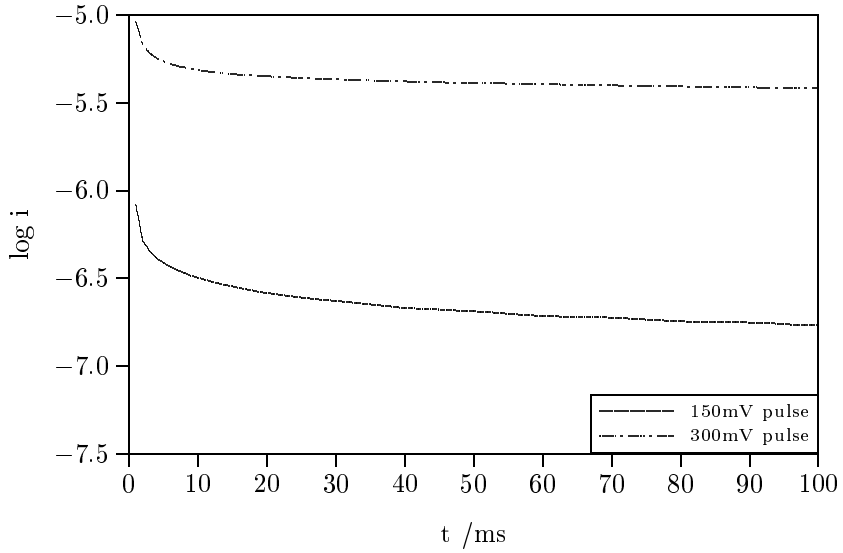


Figure 8: Simple figure included with scaling factor equals one.

Note that the label is given without `fig:`. The general parameters are

```
\includeEps[htpb]{path}{label}{caption}{scaling factor}
```

Now imagine one has one type of figure which occurs very often in the document. Then the user could use the following command

```
\includeStandardGraph[htpb]{path}{label}{caption}
```

where the standard scaling factor can be set generally for all the figures included by this command. The default is

```
\renewcommand{\standardGraphScale}{1}
```

However, by default this value is not used but the figure is scaled to achieve the width equal to the length of the following variable.

```
\setlength{\myStandardFigureWidth}{\linewidth}
```

If the figures are scaled to the width above or if a standard scaling factor is assumed depends on the following counter

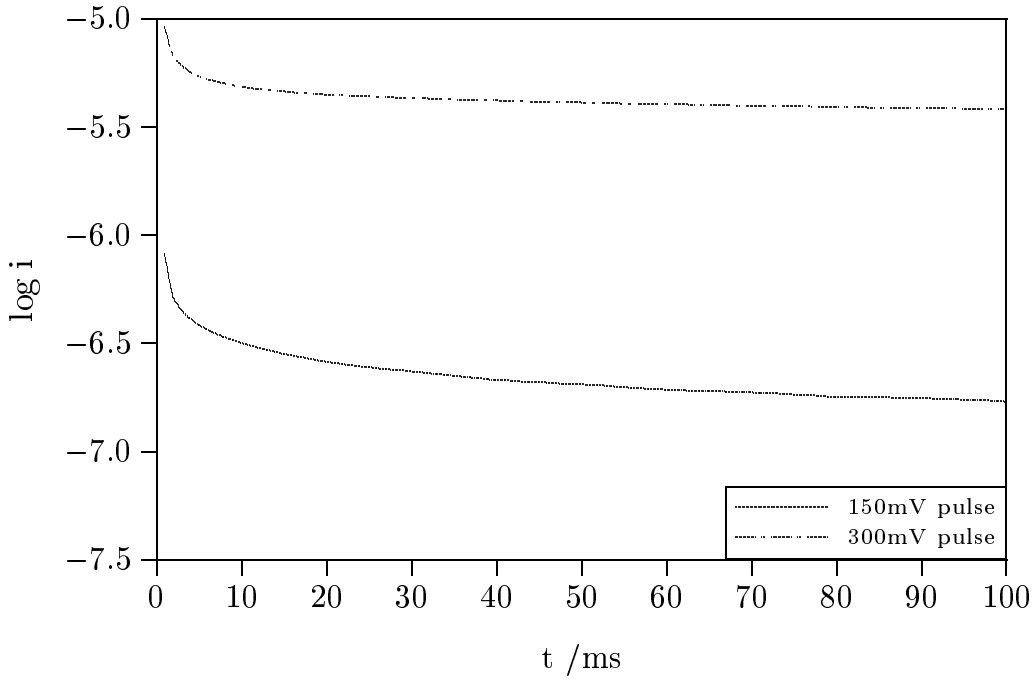


Figure 9: Simple figure included with scaling factor calculated to scale figure to meet specified width.

```
\setcounter{scaleToStandardFigureWidth}{1}
```

Set `1` to to scale to standard-figure-width or `0` to use standard-graph-scale.

Additionally one can set if each figure is to be scaled individually or if the scaling factor is to be calculated at the first time the include-standard-graph command is stated

```
\setcounter{scaleFigureIndividual}{1}
```

Set `1` to to scale individually or `0` to use scaling factor calculated when inserting the first standard-graph.

When we now include a figure using the command for standard figures

```
\includeStandardGraph{figures/1}{Simple figure
included with scaling factor calculated to scale figure to meet
specified width.}{singleStandard}
```

we obtain a figure scaled to the specified width in Figure 9.

To show the use of the counter `plotFigures` we set

```
\setcounter{plotFigures}{0}

\includeStandardGraph{figures/1}{Simple figure included with
scaling factor calculated to scale figure to meet specified
width.}{singleStandard2}

\setcounter{plotFigures}{1}
```

to obtain Figure 10. We can see that the scaling factor is no more equals one. Other usefull information is given by the label.

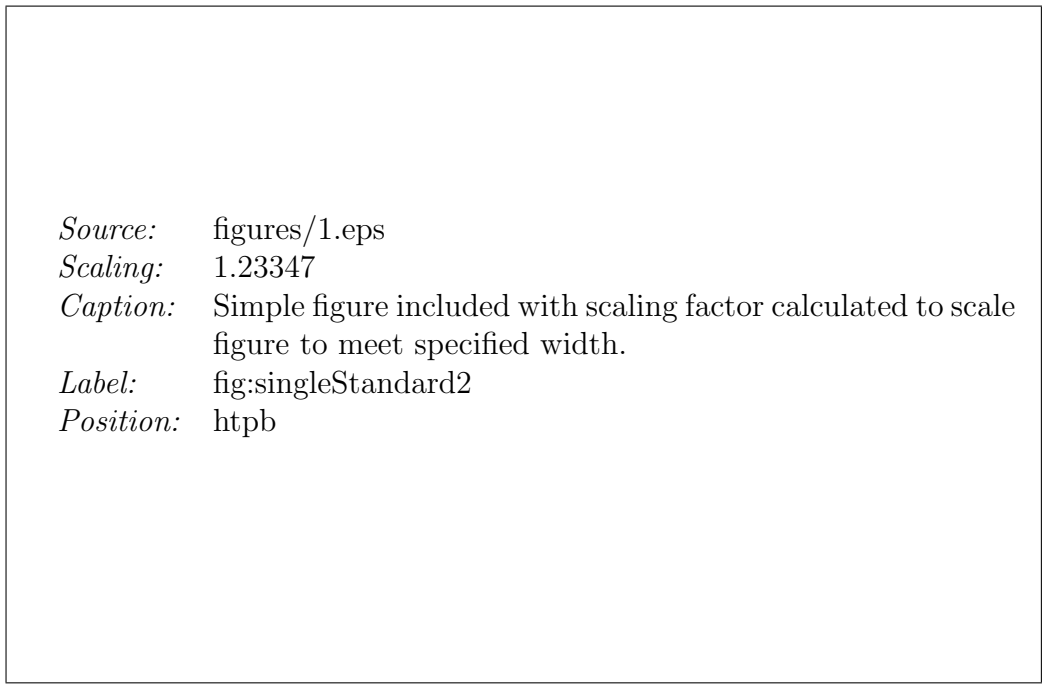


Figure 10: Simple figure included with scaling factor calculated to scale figure to meet specified width.

Of course there is also a side-ways version of a figure, see Figure 11. This figure is implemented using the include-standrd-side-ways-graph command.

```
\setcounter{plotFigures}{0}

\includeStandardSideWaysGraph{figures/1}{Simple figure included with
```

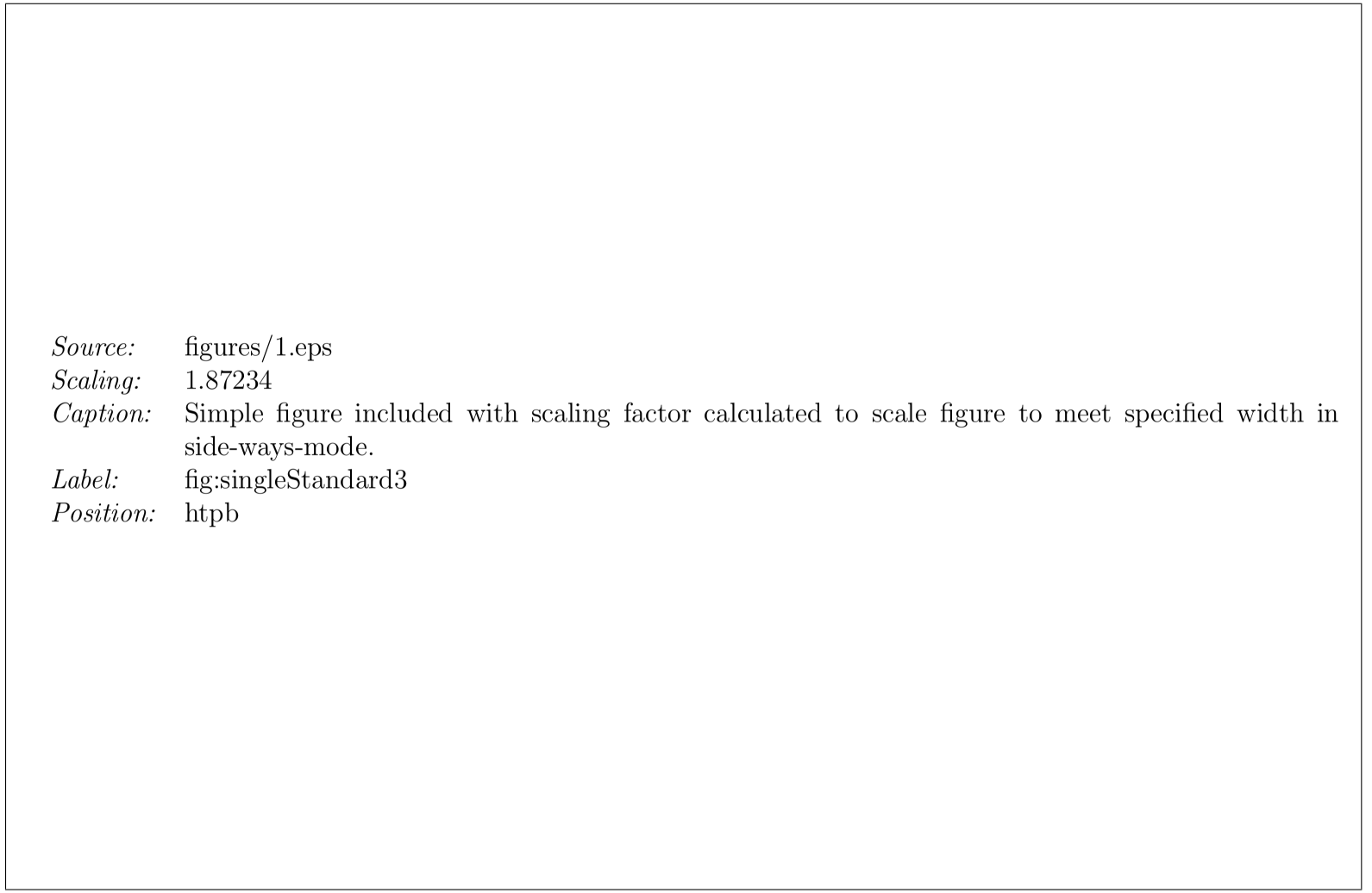
```
scaling factor calculated to scale figure to meet specified in width
in side-ways-mode.}{singleStandard3}

\setcounter{plotFigures}{1}
```

Note that the width of the side-ways-figure is now determined by

```
\setlength{\myStandardSideWaysFigureWidth}{\textheight}
```

Note that there is also an include-side-ways-eps command where you have to set a scaling factor.



Source: figures/1.eps
Scaling: 1.87234
Caption: Simple figure included with scaling factor calculated to scale figure to meet specified width in side-ways-mode.
Label: fig:singleStandard3
Position: htpb

Figure 11: Simple figure included with scaling factor calculated to scale figure to meet specified width in side-ways-mode.

4 Including wrapped figures

The use of the include-Eps-Wrap command is similar to the include-Eps command. By writing

```
\includeEpsWrap[20]{l}{figures/wrapMe}{Example of an
wrapped figure}{wrappedFigure1}{1}
```

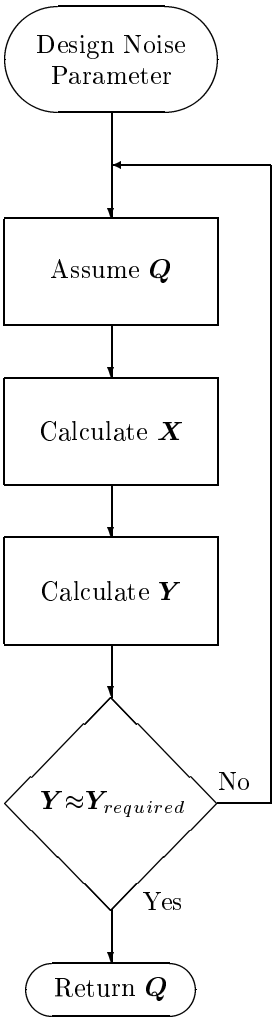


Figure 12: Example of an wrapped figure

we obtain Figure 12.
The general form is given by

```
\includeEpsWrap[lines]{side}{path}
{caption}{label}{scaling factor}
```

where lines stands for the number of narrow lines and side stands for left (l) or right (r) as specified in the documentation of the wrapfig package.

Of course, if the counter `plotFigures` equals 0 then this figure is replaced by a framebox containing some useful information:

```
\setcounter{plotFigures}{0}
\includeEpsWrap[20]{l}{figures/wrapMe}
{Example of an wrapped figure}
{wrappedFigure2}{1}
\setcounter{plotFigures}{1}
```

The counter `plotFigures` is inteded to be `0` in draft mode to improve browsing speed through the document and to provide information for referencing. However, when a figure is to be implemented the first time one like to see the result. This can be done without changing the counter `plotFigures`. Simple one can add the term `Now` to any command provided to include figures. Therefore the following command would show the figure and not the framebox:

```
\setcounter{plotFigures}{0}
\includeEpsWrapNow{l}{figures/wrapMe}
{Example of an wrapped figure}
{wrappedFigure3}{1}
\setcounter{plotFigures}{1}
```

To customize the figure there is the width which is added to the wrapped figure in oder to have some space between figure and text

```
\setlength{\extraWrapWidth}{2mm}
```

The space above the wrapped figure is set to be

```
\setlength{\aboveWrapFigureSpace}{0mm}
```

5 Including arranged sub-figures and sub-sub-figures

The content of this subsection is the generation of figures which can consist of several columns and rows of figures with their own caption (Figure ...), I will call these figures sub-figures.

Inside these sub-figures can now be arranged several columns and rows of sub-sub-figures. These are placed using the subfigure package.

A simple example might be given by Figure 13 to 14

```
\begin{arrangedFigure}{2}{2}{arranged11}{This is the left
                        sub-figure of an arranged figure.}
    \subFig{figures/1}%
    \subFig{figures/2}%
    \subFig{figures/3}%
    \subFig{figures/4}%
\newSubFig{arranged12}{This is the right sub-figure of an
                                        arranged figure.}
    \subFig{figures/1}%
    \subFig{figures/2}%
    \subFig{figures/3}%
    \subFig{figures/4}%
\end{arrangedFigure}
```

The first parameter to the `arrangedFigure` environment is the number of columns of sub-figures, the second one specifies the number of columns of sub-sub-figures. The next parameters is the label and then the caption to the immediate following sub-sub-figures.

The command `subFig` should actually be called `subSubFig` since it inserts the sub-sub-figure specified by the path of the figure to be included. *It is important to enter the %-signs after the* ***`subFig`*** *command in order to avoid introducing unwanted spaces between the sub-sub-figures.*

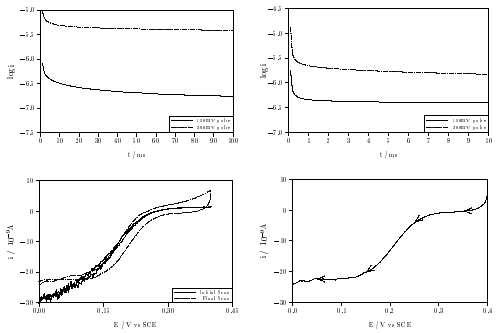


Figure 13: This is the left sub-figure of an arranged figure.

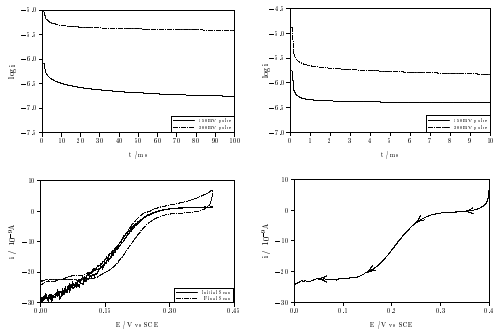
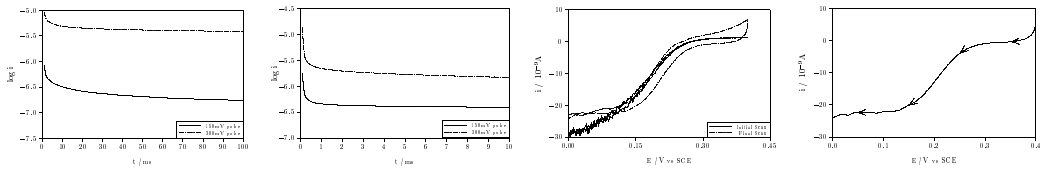


Figure 14: This is the right sub-figure of an arranged figure.

The command `newSubFig` opens the next que to fill up a sub-figure with sub-sub-figures. The parameter is the label and the caption for the immediately following sub-figure.

Another example could be given by Figure 15

```
\begin{arrangedFigure}{1}{4}{arranged2}{This is the
                only sub-figure of an arranged figure.}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```



(a) (b) (c) (d)

Figure 15: This is the only sub-figure of an arranged figure.

Note that the number of columns of sub-figures changed to one and that we arrange four sub-sub-figures in the width of one sub-figure.

Also the `subFig` command got now an empty optional command leading to the indexing of the sub-sub-figures from a to d. This behavior is identically with the one of the subfigure package.

To complete a list of examples a last one is given by Figure 16 to 18

```
\renewcommand{\subfigbottomskip}{0pt}%
\renewcommand{\subfigcapskip}{0pt}%
\renewcommand{\subfigcapskip}{0pt}%
\renewcommand{\subcapsize}{\scriptsize}%

\begin{arrangedFigure}{2}{4}{arranged31}{top left}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
\newSubFig{arranged32}{top right}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
\newSubFig{arranged33}{bottom}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```

In this figure we have an odd ratio of sub-figures to sub-figure-columns and we see that the remaining sub-figure is shifted to the right.

The same happens to the sub-sub-figures included in the top row of sub-figures.

The reader certainly notices that it takes longer to load the page when a lot of sub-sub-figures are arranged. This is an important reason to set the counter `plotFigures` to zero to save time as the reader might notice when loading Figure 19 to 20.

```
\setcounter{plotFigures}{0}
```

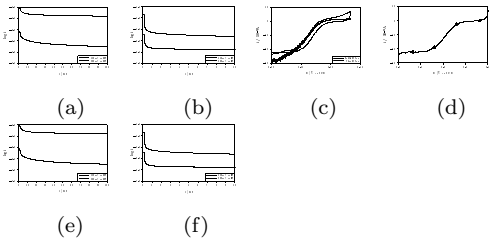


Figure 16: top left

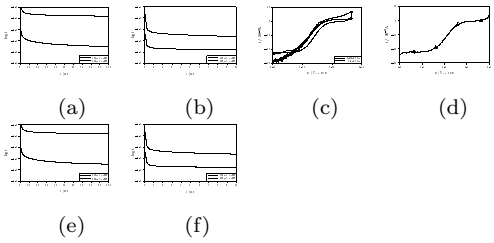


Figure 17: top right

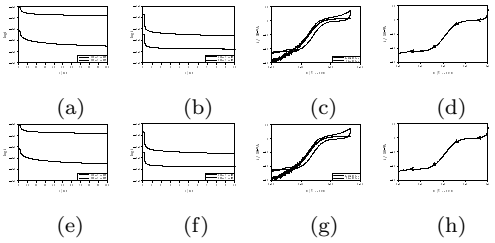


Figure 18: bottom

At the same time an example of the arranged-side-ways-figure is shown. That this page loads faster is not theonly difference one can notice. The label for reference is now added to the caption in order to be easier accessible. A disadvantage is that the caption might become bigger and that might influence the document layout. This might be changed in later versions and the label migth be moved to the margin or so.

```
\setcounter{plotFigures}{0}

\begin{arrangedSideWaysFigure}{2}{2}{arranged41}{left sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\newSubFig{arranged42}{right sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedSideWaysFigure}

\setcounter{plotFigures}{1}
```

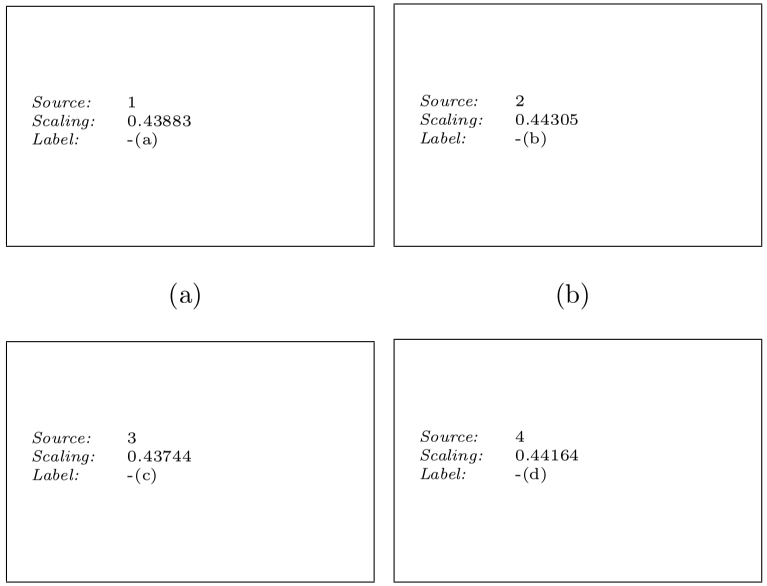


Figure 19: left sub-figure; LABEL⇒`fig:arranged41`

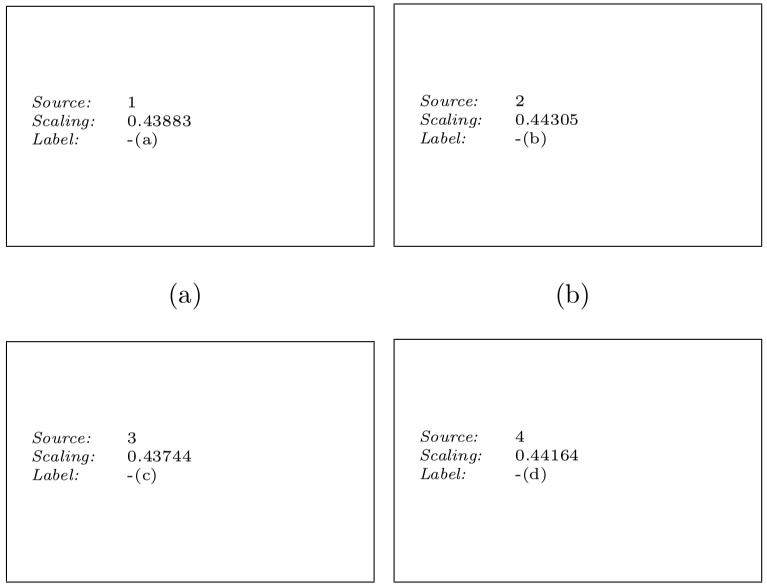


Figure 20: right sub-figure; LABEL⇒`fig:arranged42`

6 Customizing arranged figures

For cutomizing purposes one should know that the space between each sub-figure is specified by

```
\setlength{\interSubFigSpace}{8mm}
```

whereas the space between each sub-sub-figure is specified by

```
\setlength{\interSubSubFigSpace}{2.5mm}
```

The horizontal position of the numvering of sub-sub-figures *(a)-(..)* can be changed by changing the following length which default value is

```
\setlength{\subCaptionPenalty}{0mm}
```

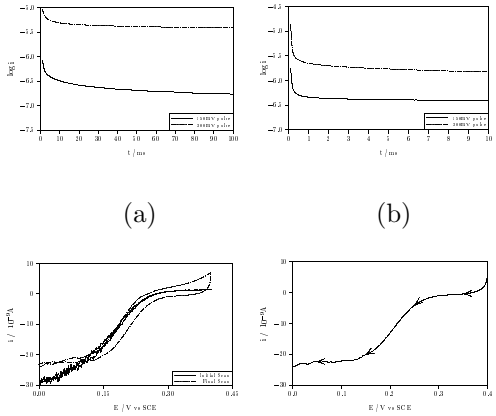
This has only an effect if the `subFig` command is issued with an empty optional parameter.

If a sub-sub-caption is given as in the following example

```
\begin{arrangedFigure}{2}{2}{arranged511}{left sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[This is a very long sub-sub-caption]{figures/3}%
    \subFig[]{figures/4}%
\newSubFig{arranged512}{right sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```

then the following length can be set in order to compensate for the extra space

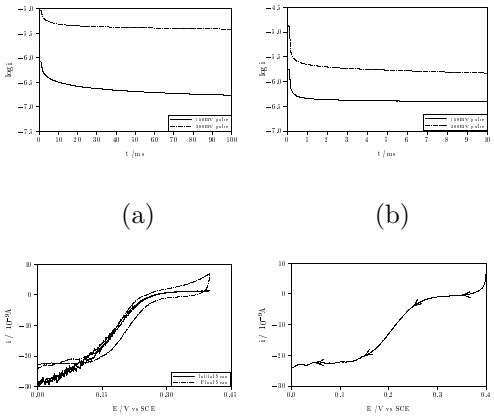
```
\begin{arrangedFigure}{2}{2}{arranged51}{left sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[This is a very long sub-sub-caption]{figures/3}%
    \subFig[]{figures/4}%
\newSubFig{arranged52}{right sub-figure}
    \subFig[]{figures/1}%
```



(a) (b)

(c) This is a very long sub-sub-caption (d)

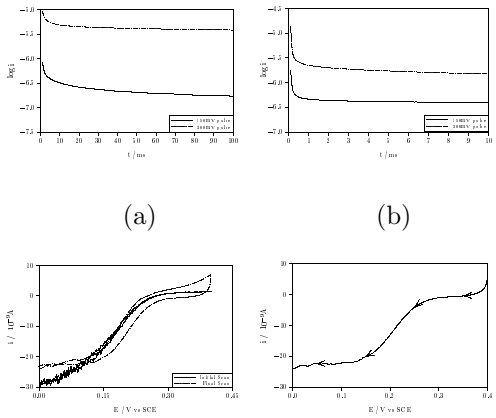
Figure 21: left sub-figure



(a) (b)

(c) (d)

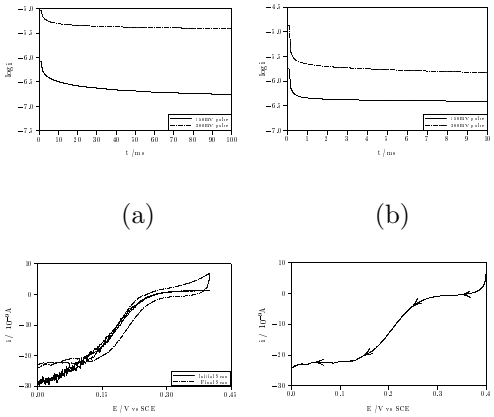
Figure 22: right sub-figure



(a) (b)

(c) This is a very long sub-sub-caption (d)

Figure 23: left sub-figure



(a) (b)

(c) (d)

Figure 24: right sub-figure

```
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\setlength{\subFigureAboveCaptionSpace}{7mm}
\end{arrangedFigure}
\setlength{\subFigureAboveCaptionSpace}{0mm}
```

with a resulting Figure 23 and 24.

If the figure looks like Figure 25 and 26 then one could set

```
\begin{arrangedFigure}{2}{2}{arranged531}{left sub-figure}
    \subFig[]{figures/1}%
    \subFig[This is a very long sub-sub-caption]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\newSubFig{arranged532}{right sub-figure}
    \setlength{\horizontalSubSubFigSpace}{4mm}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \setlength{\horizontalSubSubFigSpace}{0mm}
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```

to achieve Figure 27 and 28.

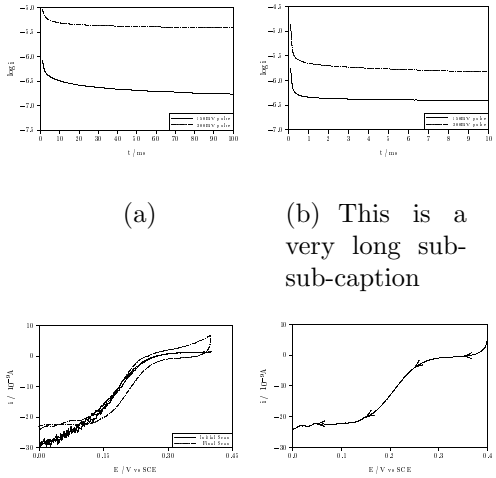
Another parameter for modifying the look is given by

```
\setlength{\subFigureBelowCaptionSpace}{3mm}
```

which sets the space below the main sub-figure caption and the next main sub-figure below it.

After all, it might be difficult to reset every variable on its own and to remenber their default values. Therefore, the command `resetMiniPlot` has been introduced. This sets variables and commands to their default values. Note that the counter `plotFigures` is not reset. A listing of the definition is given below.

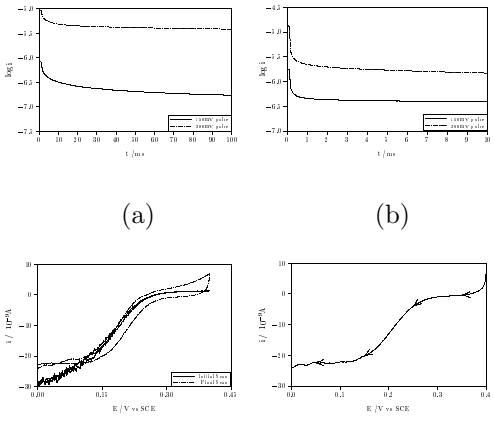
```
\newcommand{\resetMiniPlot}%
    {%
    \setcounter{printCaption}{1}
    \setlength{\extraWrapWidth}{2mm}
    \setlength{\aboveWrapFigureSpace}{0mm}
```



(a) (b) This is a very long sub-sub-caption

(c) (d)

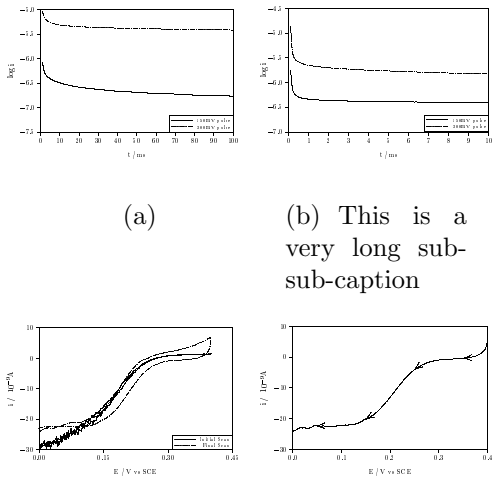
Figure 25: left sub-figure



(a) (b)

(c) (d)

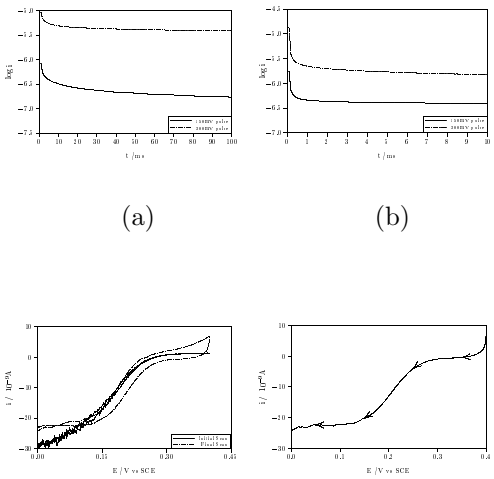
Figure 26: right sub-figure



(a) (b) This is a very long sub-sub-caption

(c) (d)

Figure 27: left sub-figure



(a) (b)

(c) (d)

Figure 28: right sub-figure

```
\renewcommand{\standardGraphScale}{1}
\setcounter{scaleToStandardFigureWidth}{1}
\setcounter{isFirstStandardFigure}{1}
\setcounter{scaleFigureIndividual}{1}
\setlength{\frameBoxPenalty}{2.4mm}
\setlength{\myStandardFigureWidth}{\linewidth}
\setlength{\myStandardSubFigureWidth}{\myStandardFigureWidth}
\setlength{\myStandardSideWaysFigureWidth}{\textheight}
\renewcommand{\isFirstSubFig}{false}
\setlength{\subFigPenalty}{0mm}
\setlength{\subSubFigPenalty}{0.001mm}
\setlength{\interSubFigSpace}{8mm}
\setlength{\interSubSubFigSpace}{2.5mm}
\setlength{\subFigureAboveCaptionSpaceDefaultOffset}{0mm}
\setlength{\subFigureAboveCaptionSpaceDefault}{0mm}
\setlength{\subFigureAboveCaptionSpace}{\subFigureAboveCaptionSpaceDefault
                                  + \subFigureAboveCaptionSpaceDefaultOffset}
\setlength{\horizontalSubSubFigSpace}{0mm}
\setlength{\subFigureBelowCaptionSpace}{3mm}
\setlength{\subCaptionPenalty}{0mm}
}%
```

7 Debugging arranged figures

Normally it shouldn't be necessary to do any debugging but in case there is I'd like to mention some tweaks here.

If one is unluky one might find something like the following graph

```
\begin{arrangedFigure}{2}{2}{arranged51d}{left sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\newSubFig{arranged52d}{right sub-figure}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```

from which one would expect to yield two columns of sub-figures and inside those again two columns of sub-sub-figures. However, Figure **??** to **??** have only one column of sub-sub-figures. This is because the sub-sub-figures plus the space between them is too wide to fit into the width of the minipage environment they are in. So, the figures are shifted down. This effect can be removed by changing the following length which default value is

```
\setlength{\subSubFigPenalty}{0.001mm}
```

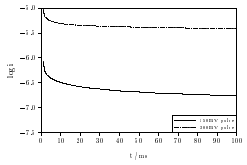
Theoretical this value should be zero but due to numerical inaccuracies the sub-sub-figures have to be made a little bit narrower. The above length is to compensate for the numerical error.

By increasing the above length one can yield a satisfactory figure layout as can be seen in Figure

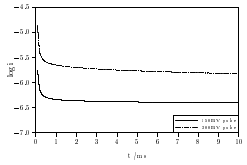
Another thing which is unlikely but not impossible to occur is that on has the displacment of figure as described above only when `plotFigures` is equals zeros, meaning the framebox(es) are shown with the information in it. The reason then is that the following length needs to be increased from its default value.

```
\setlength{\frameBoxPenalty}{2.4mm}
```

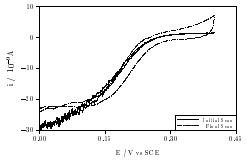
The reason is that the framebox is drawn around the actual size of the eps-figure, this space needs to be substracted.



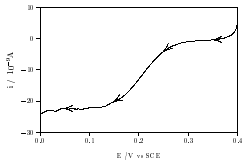
(a)



(b)

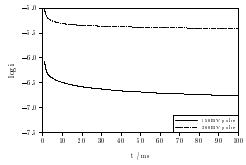


(c)

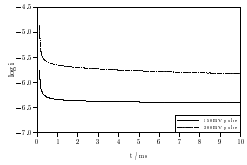


(d)

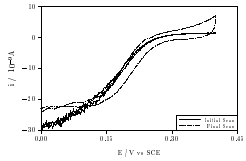
Figure 29: left sub-figure



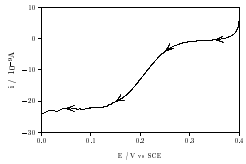
(a)



(b)

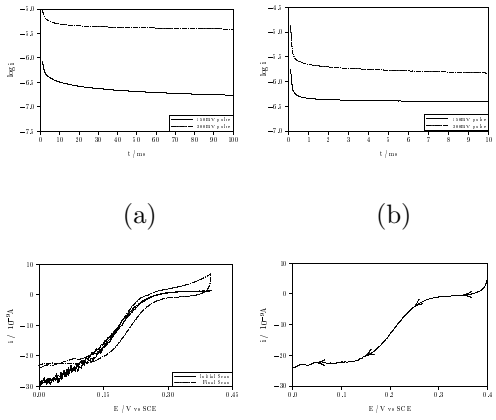


(c)



(d)

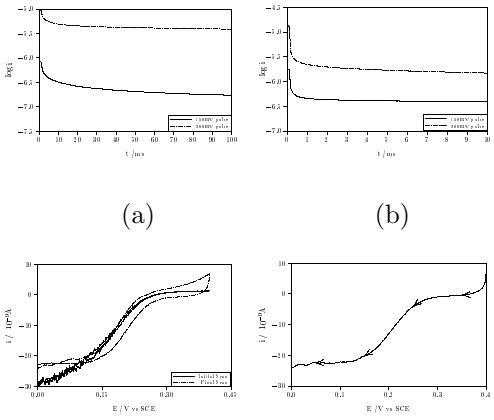
Figure 30: right sub-figure



(a) (b)

(c) (d)

Figure 31: left sub-figure



(a) (b)

(c) (d)

Figure 32: right sub-figure

8 Referencing to figures

Suppose the label given to a figure is `arranged51dd` then the actual label assign is `fig:arranged51dd` and thats what needs to be referenced to

```
... see Figure \ref{fig:arranged51dd} ...
```

... see Figure 31

Referencing to sub-sub-figures is not implemented yet but will be done in future versions as done as the subfigure package describes. That is for example

```
... see Figure \ref{fig:arranged51dd-c} ...
```

However, this does not work yet but if somebody knows how to extract the `a` from `(a)` then it wouldn't be a problem to implement it.

9 Recommendation for working with MiniPlot

The only recommendation there is is only to set

```
\usepackage{miniplot}
\setcounter{plotFigures}{0}
```

This will display only the frameboxes of the figures and this will increase browsing speed through the document significantly.

When implementing a figure for the first time one should use the `Now`-version of the implementation command, for example

```
\includeEpsNow{...
```

After the figure is set and everything works fine one should remove the `Now` to end up with

```
\includeEps{...
```

When the whole document is finished one simply sets

```
\setcounter{plotFigures}{1}
```

and all the figures will be shown.

Another thing regarding the subfig package is that I would suggest to use the following setting when implementing loads of figures in an arranged figure like Figure 16 to 18.

```
\renewcommand{\subfigbottomskip}{0pt}%
\renewcommand{\subfigcapskip}{0pt}%
\renewcommand{\subfigcapskip}{0pt}%
\renewcommand{\subcapsize}{\scriptsize}%
```

These settings are not used for this manual.

10 The code

I'd like to excuse that the code is not pasted in and not explained as proper package writers do. I just hope that the comments might give some clue when trying to understand pieces of `miniplot.sty`.

11 Restrictions when using MiniPlot

One restriction is that the underscore _ is not permitted in the labels of the figures.

This is due to to the fact that MiniPlot displays the labels and I need to find a proper way of doing this. The simplyfied explanation would trying to write

```
\newcommand{\examplea}{label_lab}
\examplea
```

either MiniPlot cannot display the underscore in text mode or in math-mode the underscore has a different function and is not displayed neither. I could imagine there is something like a temporary solution like using the let command but this goes beyond my LaTeX capabilities.

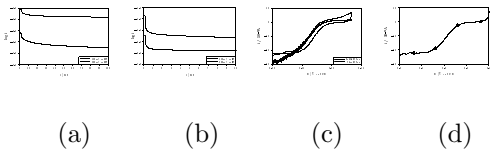
Another restriction is that the path to an arranged figure cannot contain the

```
\
```

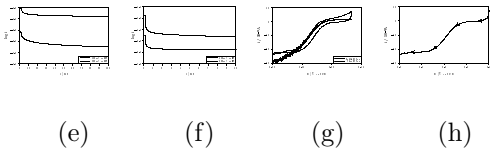
sign. This might not be a major restriction.

12 Known bugs

There is only one (fixable) bug known which is that when the columns of the sub-figures are not evenly filled up with sub-sub-figures the sub-figure shift a little bit as in Figure 33 to 35. Compare the upper boundaries of the two top sub-figures. There is a little vertical shift.

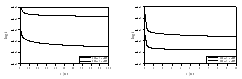


(a) (b) (c) (d)

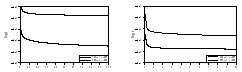


(e) (f) (g) (h)

Figure 33: top left

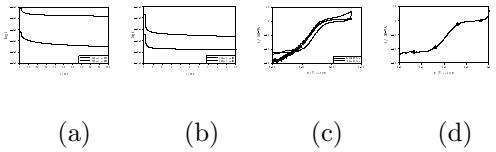


(a) (b) (c) (d)

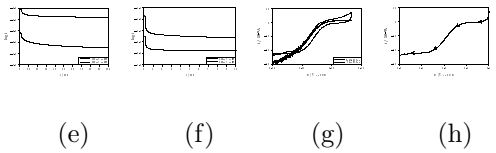


(e) (f)

Figure 34: top right



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 35: bottom

However, this can be solved by typing

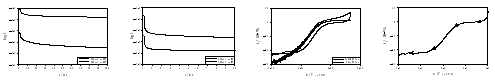
```
\begin{arrangedFigure}{2}{4}{arranged551}{top left}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
```

```
    \subFig[]{figures/4}%
\newSubFig{arranged552}{top right}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \setlength{\subFigureAboveCaptionSpace}{2.5mm}
\newSubFig{arranged553}{bottom}
    \setlength{\subFigureAboveCaptionSpace}{0mm}
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
    \subFig[]{figures/1}%
    \subFig[]{figures/2}%
    \subFig[]{figures/3}%
    \subFig[]{figures/4}%
\end{arrangedFigure}
```

using `subFigureAboveCaptionSpace` and it follows Figure 36 and 38

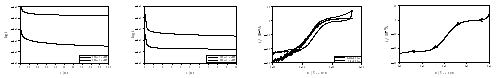
This bug might be removed one time in the future. Sugestions are very much welcome!

(a) (b) (c) (d)

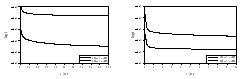


(e) (f) (g) (h)

Figure 36: top left

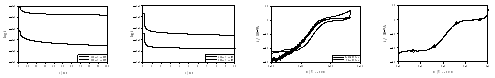


(a) (b) (c) (d)

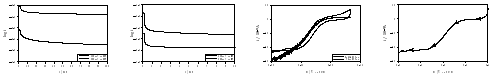


(e) (f)

Figure 37: top right



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 38: bottom

Generally bugs can please be reported to *enxtw1@nottingham.ac.uk* or *tobias_wahl@gmx.de*.

One thing is that one gets a lot of warnings about overfull hboxes. I am personally not really bothered but if somebody might come up with an idea I would be happy to implement it.

There are just few awkward things about the code and maybe somebody has an idea how solve things. One thing is that in the following function the first line seems to introduce a vertical length even if one simply writes `vspace{0pt}`. Due to this strange behavior I had to introduce the length `subFigureAboveCaptionSpaceDefaultOffset` equals 3mm[1].

```
\newcommand{\newSubFig}[3][\subFigureAboveCaptionSpaceDefault]%
    {%
    \vspace{\subFigureAboveCaptionSpace}%
    \ifthenelse{\value{plotFigures}>0}%
        {\caption{\myCaptionText}}%
        {\caption{\myCaptionText\printLabel{\myLabel}}}%
    \label{\myLabel}%
    \end{minipage}%
    \renewcommand{\myCaptionText}{#3}%
    \renewcommand{\myLabel}{#2}%
    \vspace{\subFigureBelowCaptionSpace}%
    \hspace{\interSubFigSpace}%
    \begin{minipage}{\myStandardSubFigureWidth}%
    \aboveCaptionSpace{#1}%
    }%
```

Furthermore in the above command the following command is called

```
\newcommand{\aboveCaptionSpace}[1]{\setlength{\subFigureAboveCaptionSpace}
                                {#1 + \subFigureAboveCaptionSpaceDefaultOffset}}
```

For some strange reason does the length `subFigureAboveCaptionSpace` keep its value only in the minipage where I define it. I also cannot set the length outside the minipage since it did not recognize the cange within it. I would think that this has something to do with local and global variables but I normally would have expected the the length in question is a global variable and known everywhere.

The command `aboveCaptionSpace` was initially thought to be used instead of setting the length `subFigureAboveCaptionSpace` directly.

[1] I have to note that this behavior happened at one point and does not occur annymore. So that `subFigureAboveCaptionSpaceDefaultOffset` is set to zero. However, I decided to leave this in the code in case it might occur again??

13 List of variables, commands and environments

Note that only those variables, commands and environments are cited which are useful for the user.

Counters:

```
\setcounter{scaleToStandardFigureWidth}{1}
\setcounter{isFirstStandardFigure}{1}
\setcounter{scaleFigureIndividual}{1}
\setcounter{plotFigures}{1}
```

Lengths:

```
\setlength{\myStandardFigureWidth}{\linewidth}
\setlength{\myStandardSideWaysFigureWidth}{\textheight}

\setlength{\aboveWrapFigureSpace}{0mm}

\setlength{\interSubFigSpace}{8mm}
\setlength{\interSubSubFigSpace}{2.5mm}

\setlength{\horizontalSubSubFigSpace}{0mm}
\setlength{\subFigureAboveCaptionSpaceDefault}{0mm}
\setlength{\subFigureAboveCaptionSpace}{0mm}
\setlength{\subFigureBelowCaptionSpace}{3mm}

\setlength{\subFigPenalty}{0mm}
\setlength{\subSubFigPenalty}{0.001mm}
\setlength{\subCaptionPenalty}{0mm}
\setlength{\frameBoxPenalty}{2.4mm}
```

Commands:

```
\newcommand{\isFirstSubFig}{false}
\newcommand{\standardGraphScale}{1}

\includeEps[htpb]{path}{caption}{label}{scaling}
\includeEpsNow[htpb]{path}{caption}{label}{scaling}
\includeSideWaysEps[htpb]{path}{caption}{label}{scaling}
\includeSideWaysEpsNow[htpb]{path}{caption}{label}{scaling}

\includeStandardGraph[htpb]{path}{caption}{label}
\includeStandardGraphNow[htpb]{path}{caption}{label}
\includeStandardSideWaysGraph[htpb]{path}{caption}{label}
\includeStandardSideWaysGraphNow[htpb]{path}{caption}{label}

\includeEpsWrap[narrow lines]{position}{path}{caption}{label}{scaling}
\includeEpsWrapNow[narrow lines]{position}{path}{caption}{label}{scaling}

\newcommand{\printLabel}[1]{; \large\textsc{Label}$\Rightarrow$\texttt{fig:#1}
\newSubFig[\subFigureAboveCaptionSpace]{label}{caption}
\subFig[sub-sub-caption]{path}%

\resetMiniPlot
```

Environments:

```
arrangedFigure[htpb]{sub-fig columns}{sub-sub-fig columns}{label}{caption}
arrangedFigureNow[htpb]{sub-fig columns}{sub-sub-fig columns}{label}{caption}

arrangedSideWaysFigure[htpb]{sub-fig columns}{sub-sub-fig columns}
                                                          {label}{caption}
arrangedSideWaysFigureNow[htpb]{sub-fig columns}{sub-sub-fig columns}
                                                          {label}{caption}
```