

NumericPlots - plot numeric data with latex

Thomas König, Alexander Michel, Michael Baumgart

July 13, 2011

Plotting numeric data is a task which often has to be done for scientific papers. In \LaTeX , however, it is only possible to include graphics created with an external program. The `pstricks`-packages provide many commands to generate graphics in \LaTeX . To generate simple graphics from numeric data, however, it is difficult to use. This package provides a simpler interface for the `pstricks`-package to plot numeric data.

Contents

1. Introduction	2
2. Multiple plots in one picture	15
3. Matlab Support	18
3.1. <code>export2latex.m</code>	18
3.2. <code>struct2latex.m</code>	19
3.3. <code>dspace2struct</code>	20
3.4. <code>dspace2latex</code>	20
4. package options	20
5. list of commands and options	20
6. Options	21
6.1. General	21
6.2. Automatic list with <code>xkvview</code>	21
7. Details	23
7.1. Coordinate Systems	23

8. Further Examples	25
8.1. Fill area between plots	26
9. Some test plots	28

Part I.

Introduction

Plotting numeric data is a task which often has to be done for scientific papers. In \LaTeX , however, it is only possible to include graphics created with an external program. The pstricks-packages provide very many commands to generate graphics in \LaTeX . To generate simple graphics from numeric data, however, it is difficult to use. This package provides a simpler interface for the pstricks-package to plot numeric data.

NumericPlots is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

NumericPlots is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with NumericPlots. If not, see <http://www.gnu.org/licenses/>.

Copyright 2010 Thomas König, Alexander Michel, Michael Baumgart

Part II.

Using the package

1. Basic Functionality

The package `NumericPlots`

```
\usepackage{NumericPlots}
```

is intended to be used to plot numeric data which may, e.g., be exported from Matlab by `export2latex.m`. The data must be defined in the form

```
\def\IdentI{
1.0 1.0e2
1.1 11e1
1.2 1.25e2
1.3 110
1.4 100
1.5 90
1.6 80
}
```

where the first column contains the x, the second column the y-data.

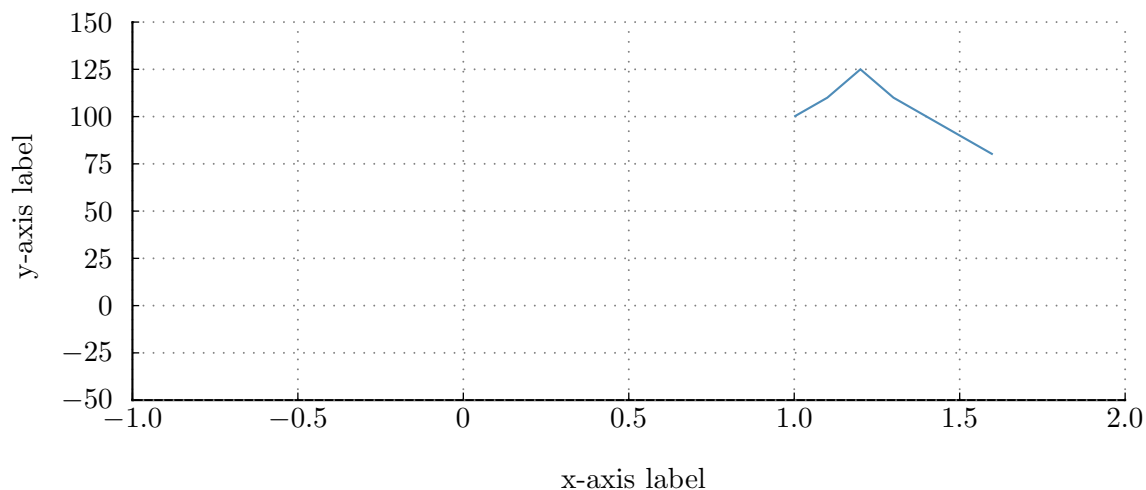
1.1. plots

The easiest plot may be done by

```
\begin{NumericDataPlot}{\textwidth}{5cm}
\setxAxis{xMin=-1, xMax=2, Dx=0.5}
\setyAxis{yMin=-50, yMax=150, Dy=25}

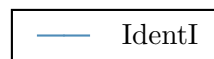
\plotxAxis{x-axis label}
\plotyAxis{y-axis label}

\listplot[style=StdLineStyleA]{\IdentI}
\end{NumericDataPlot}
```



if you want to add a legend, you simply call

```
\LegendDefinition{
\LegLine{style=StdLineStyA} & IdentI
}
```

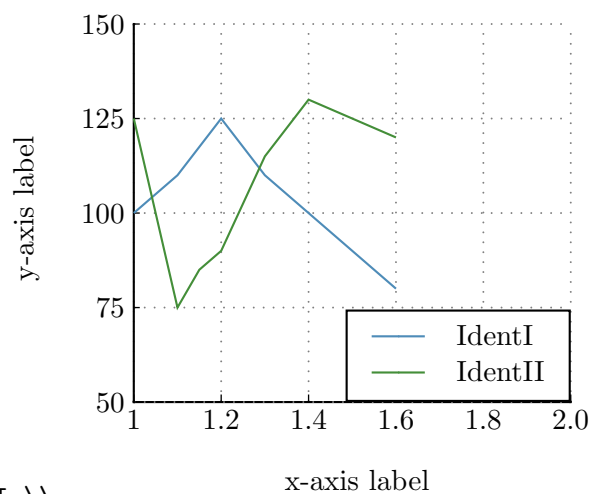


To plot multiple data in one plot call

```
\begin{NumericDataPlot}
{\textwidth}{5cm}
\setxAxis
{xMin=1, xMax=2, Dx=0.2}
\setyAxis
{yMin=50, yMax=150, Dy=25}

\plotxAxis{x-axis label}
\plotyAxis{y-axis label}

\listplot[style=StdLineStyA]
{\IdentI}
\listplot[style=StdLineStyB]
{\IdentII}
```



```
\putSE{\LegendDefinition{
\LegLine{style=StdLineStyA} & IdentI \
\LegLine{style=StdLineStyB} & IdentII
}}
\end{NumericDataPlot}
```

1.2. Label and TickLabels

The commands `plotxAxis` and `plotyAxis` take the options `NoLabel`, `NoTicks`, `NoTickLabel` as well as `LabelOption` and `TickLabelOption` which may be used to eliminate or change the look of the labels.

Standard values for `LabelOption` and `TickLabelOption` may be set by `\newcommand{\StdLabelOption}{\color{blue}}` and `\newcommand{\StdTickLabelOption}{\small}`.

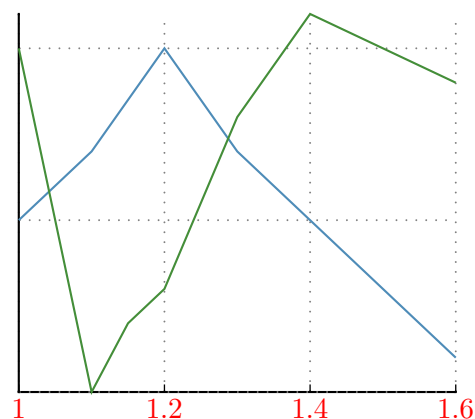
The option `xLabelSep` for `\plotxAxis` (and `yLabelSep` for `\plotyAxis`) may be used to set the separation between the axis and the label. Standard value is 10pt for the x-label and 20pt for the y-label.

```
\begin{NumericDataPlot}{\textwidth}{5cm}
\setxAxis{xMin=1, xMax=1.6, Dx=0.2}
\setyAxis{yMin=75, yMax=130, Dy=25}
```

```
\plotxAxis
[LabelOption=\LARGE,%
TickLabelOption=\color{red},%
xLabelSep=40pt]
{x-axis label}
\plotyAxis
[NoLabel, NoTicks, NoTickLabel]
{y-axis label}
```

```
\listplot[style=StdLineStyleA]{\IdentI}
\listplot[style=StdLineStyleB]{\IdentII}
```

```
\end{NumericDataPlot}
```



x-axis label

1.3. Place “Objects” in the plot.

There are basically two different options to place objects in the plot. To understand the difference one has to keep in mind that the axis have two different coordinate systems. One is the system defined by `xMin`, `xMax`, `yMin` and `yMax` (referred to as “DataCoordinateSystem”), the other is the system defined by `xCoordMin`, `xCoordMax`, `yCoordMin` and `yCoordMax` (referred to as “PictureCoordinateSystem”), see section 2.

It is now possible to place stuff in the graph with the DataCoordinates with the command `NDPput`, see the following example.

```

\begin{NumericDataPlot}
  {\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=2, Dx=0.2}
\setyAxis
  {yMin=50, yMax=150, Dy=25}

\plotxAxis{x-axis label}
\plotyAxis[NoLabel]{}

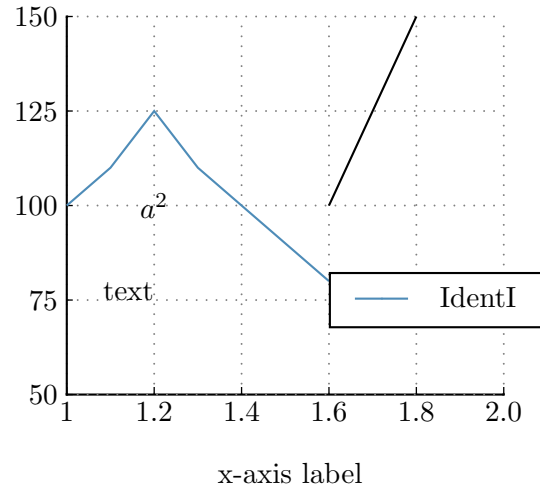
\listplot[style=StdLineStyleA]
  {\IdentI}

% put some stuff somewhere
\NDPput[x=1.2, y=75, RefPoint=br]{text}
\NDPput[x=1.2, y=100]{ $a^2$ }

% or put nodes...
\NDPput[x=1.6, y=100]{\pnode{A}}
\NDPput[x=1.8, y=150]{\pnode{B}}
% ...and draw a line between them
\ncline{A}{B}

% or put the legend at a specific position
\NDPput[x=1.8, y=75]{\LegendDefinition{
\LegLine{style=StdLineStyleA} & IdentI
}}
\end{NumericDataPlot}

```



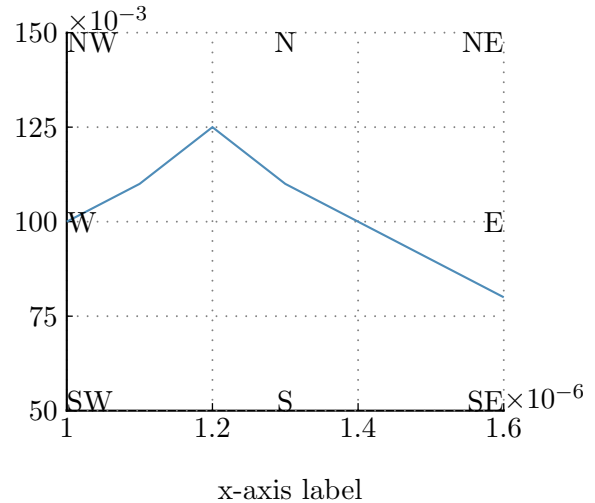
For convenience the commands `\putXX{object}` where $XX \in (N, S, E, W, NW, NE, SW, SE)$ are defined to place something in the North, South, ..., SouthEast corner of the plot. Also, the command `\putExpY{xx}` and `\putExpX{xx}` may be used to place exponents at the axes.

```

\begin{NumericDataPlot}
...
\putExpY{$\times 10^{-3}$}
\putExpX{$\times 10^{-6}$}

\putN{N}
\putS{S}
\putW{W}
\putE{E}
\putNW{NW}
\putNE{NE}
\putSW{SW}
\putSE{SE}
\end{NumericDataPlot}

```



Alternatively, stuff can be placed within the plot with `\rput`.

```

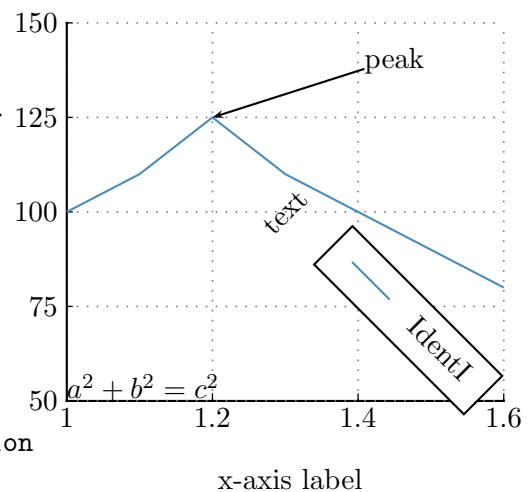
\begin{NumericDataPlot}
...

% put text in the middle of the plot
\rput{45}(500,500){text}
% put a formula in the lower left corner
\rput[b1](0,0){$a^2+b^2=c^2$}

% or put nodes...
\NDPput[x=1.2, y=125]{\pnode{A}}
\rput(750,900){\Rnode{B}{peak}}
% ...and draw a line between them
\ncline{<-}{A}{B}

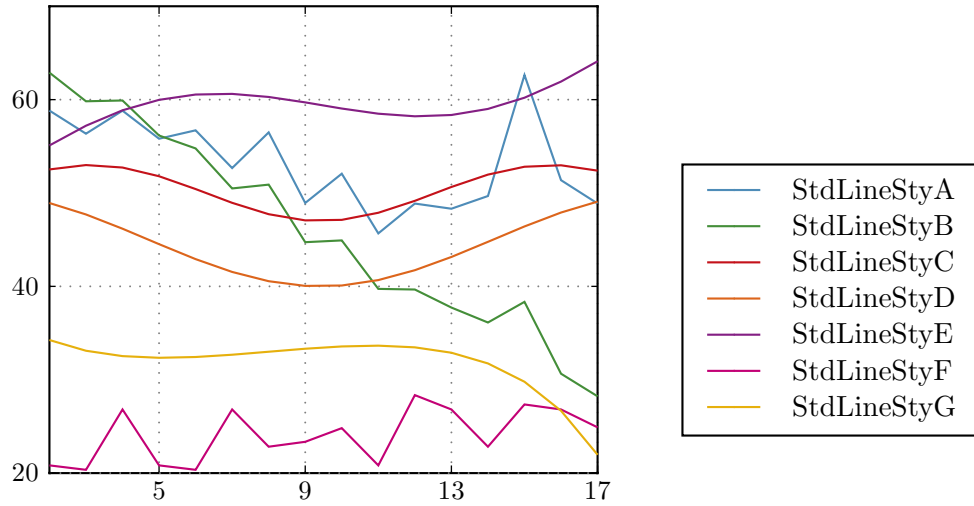
% or put the legend at a specific position
\rput{-45}(750,250){\LegendDefinition{
\LegLine{style=StdLineStyleA} & IdentI
}}
\end{NumericDataPlot}

```

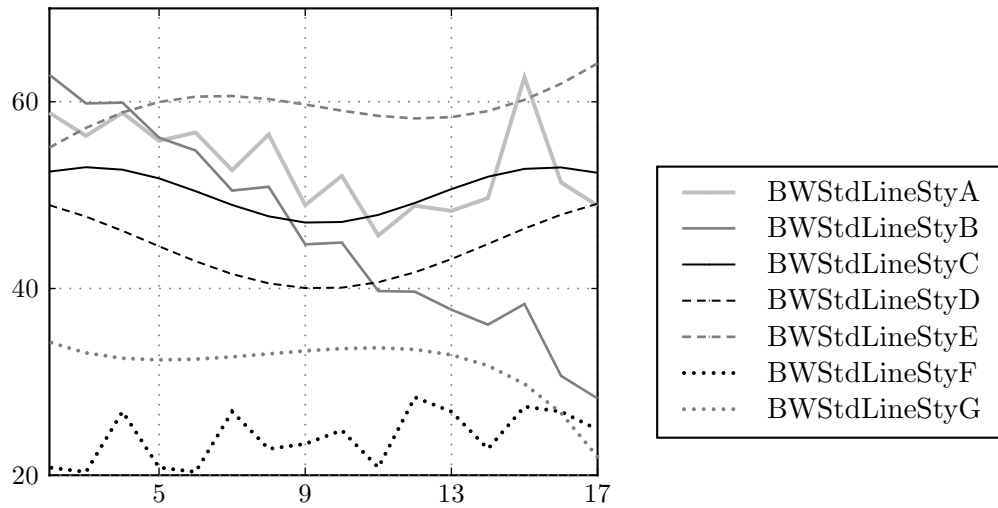


1.4. Linestyles and colors

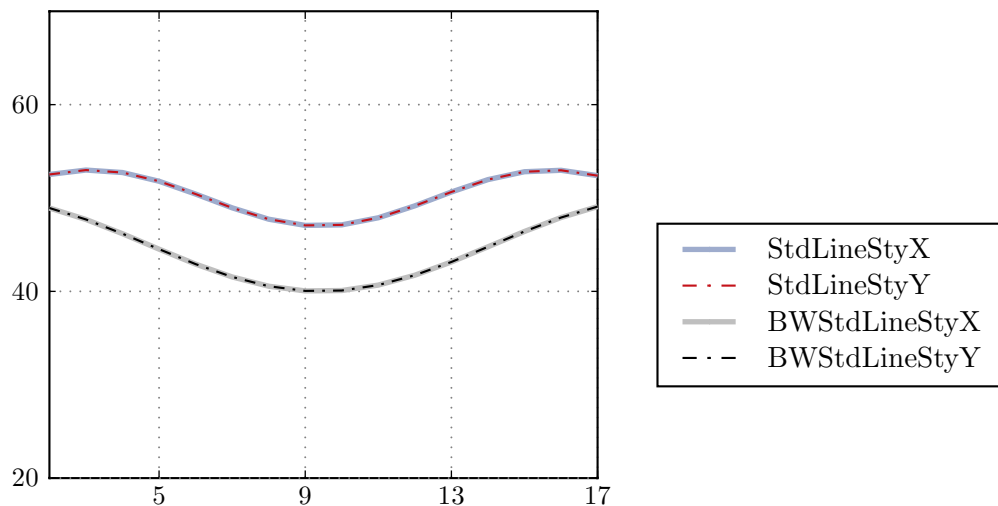
While using the package, there are predefined linestyles which may be used:



When using the package option `BW` the standard line styles will be replaced by their black and white counterparts:



For values which are nearly the same (reference and measurement, e.g.) the following line styles may be used:



It is, of course, possible to redefine the available linestyles or to define new linestyles.

```
\definecolor{MyColor}{cmyk}{0.6 0.21 1.0 0.2}
```

```
\newsstyle{MyLine}
```

```
{linecolor=MyColor, linewidth=2pt,
linestyle=dashed,
dash=1pt 1pt 4pt 1pt 1pt 3pt,
dotstyle=*, showpoints=true,
dotsize=5pt}
```

```
\newsstyle{MyLineA}
```

```
{linecolor=blue, linestyle=dotted,
dotstyle=asterisk, showpoints=true}
```

```
\listplot[style=MyLine]
```

```
{\IdentI}
```

```
\listplot[style=MyLineA]
```

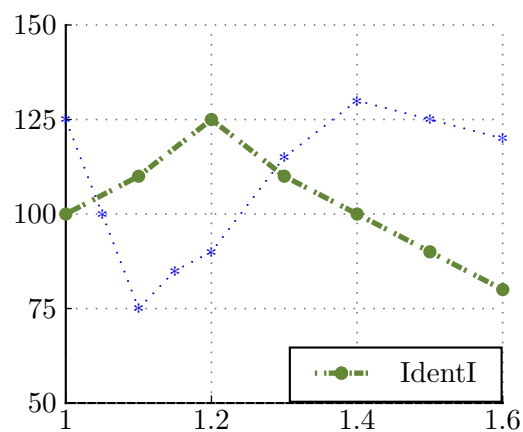
```
{\IdentII}
```

```
\putSE{\LegendDefinition{
```

```
\LegLine{style=MyLine} & IdentI
```

```
\LegLine{style=MyLineA} & IdentII
```

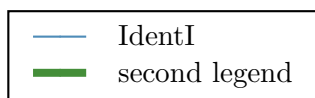
```
}}
```



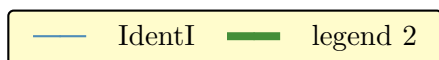
1.5. Legend

The legend may be created with `\LegendDefinition`. The command takes the two optional arguments `nrCols` and `LabelOrientation=[l|c|r]`. The mandatory argument is the definition of a table as demonstrated in the following examples.

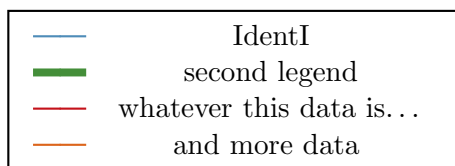
```
\LegendDefinition{
\LegLine{style=StdLineStyA} & IdentI\\
\LegLine{style=StdLineStyB, linewidth=3pt} & second legend
}
```



```
\newpsstyle{LegendBoxStyle}%
{framearc=0.2, fillstyle=solid, fillcolor=yellow, opacity=0.2}
\LegendDefinition[nrCols=2]{
\LegLine{style=StdLineStyA} & IdentI &
\LegLine{style=StdLineStyB, linewidth=3pt} & legend 2
\newpsstyle{LegendBoxStyle}%
{fillstyle=solid, fillcolor=white}
}
```



```
\LegendDefinition[LabelOrientation=c]{
\LegLine{style=StdLineStyA} & IdentI\\
\LegLine{style=StdLineStyB, linewidth=3pt} & legend 2 \\
\LegLine{style=StdLineStyC} & whatever this data is\ldots\\
\LegLine{style=StdLineStyD} & and more data
}
```



1.6. Add Lines to the Plot

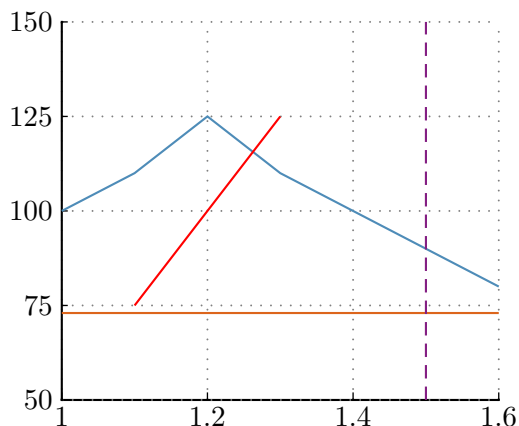
Horizontal and vertical lines may be added to the plot with the commands `\NDPhline{coord}`, `\NDPvline{coord}` and `\NDPline{coord}`. It is also possible to put nodes and draw lines between them, see placing stuff...

```
\begin{NumericDataPlot}
  {\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=1.6, Dx=0.2}
\setyAxis
  {yMin=50, yMax=150, Dy=25}

\plotxAxis[NoLabel]{}
\plotyAxis[NoLabel]{}

\listplot[style=StdLineStyleA]
  {\IdentI}

\NDPhline[linecolor=LineColorD]{73}
\NDPvline[linecolor=LineColorE, linestyle=dashed]{1.5}
\NDPline[linecolor=red]{1.1}{75}{1.3}{125}
\end{NumericDataPlot}
```



1.7. Add Boxes to the Plot

Horizontal and vertical boxes may be added to the plot with the commands `\NDPhbox{coord}`, `\NDPvbox{coord}` and `\NDPbox{coord}`. It is also possible to put nodes and draw lines between them, see placing stuff...

```

\begin{NumericDataPlot}
  {\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=1.6, Dx=0.2}
\setyAxis
  {yMin=50, yMax=150, Dy=25}

\plotxAxis[NoLabel]{}
\plotyAxis[NoLabel]{}

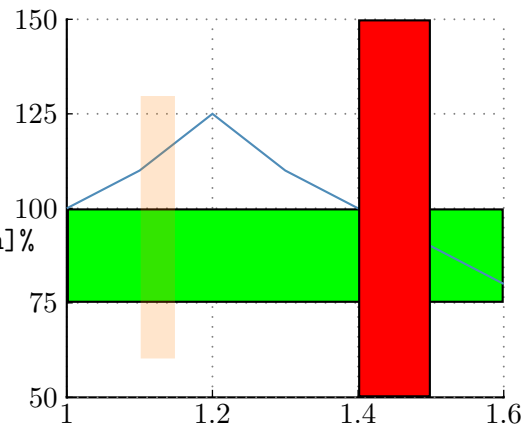
\NDPhbox[fillstyle=solid,fillcolor=green]%
  {75.0}{100.0}%

\listplot[style=StdLineStyA] {\IdentI}

\NDPvbox[fillstyle=solid,fillcolor=red]
  {1.4}{1.5}

  \NDPbox[fillstyle=solid,%
    fillcolor=orange, opacity=0.2, linestyle=none]%
    {1.1}{60}{1.15}{130}
\end{NumericDataPlot}

```



1.8. Grid

One may choose not to plot the grid with the option `NoGrid` for the commands `\plotxAxis` and `\plotyAxis`.

If the grid is plotted with the axis it may happen that the grid is plotted over the axis. To avoid this, plot the grid first and then plot the axis as shown.

```

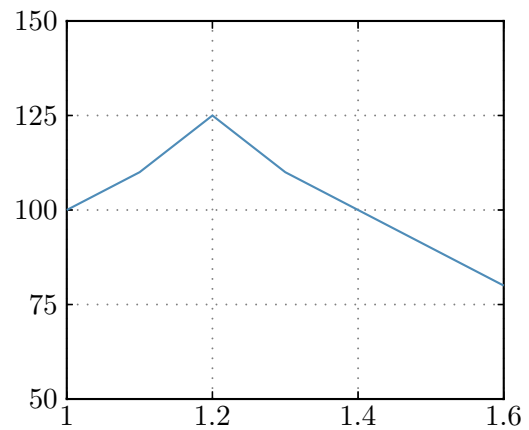
\begin{NumericDataPlot}
  {\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=1.6, Dx=0.2}
\setyAxis
  {yMin=50, yMax=150, Dy=25}

\plotxGrid
\plotyGrid
\plotxAxis
[NoLabel, NoGrid, AxisStyle=Boxed]{}
\plotyAxis
[NoLabel, NoGrid, AxisStyle=Boxed]{}

\listplot[style=StdLineStyA]
  {\IdentI}

\end{NumericDataPlot}

```



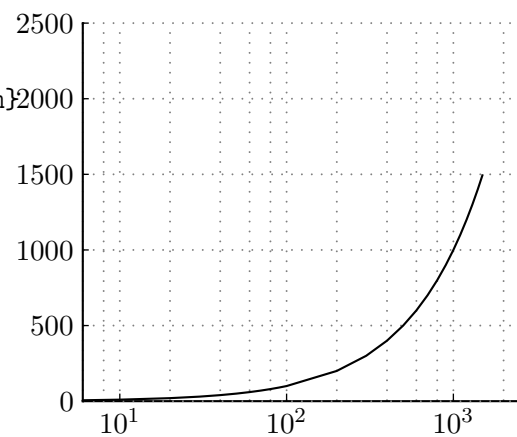
1.9. Logarithmic axes

```

\begin{NumericDataPlot}{\textwidth}{5cm}
\setxAxis
  {xMin=6, xMax=2500, Dx=10, xLog}
\setyAxis
  {yMin=0, yMax=2500, Dy=500}
\plotxAxis{}
\plotyAxis{}

\listplot{\LogData}
\end{NumericDataPlot}

```

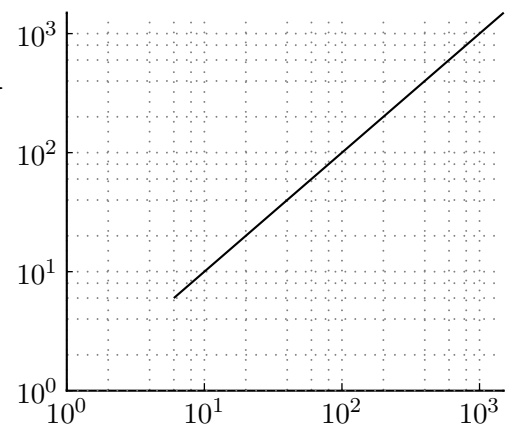


```

\begin{NumericDataPlot}{\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=1500, Dx=10, xLog}
\setyAxis
  {yMin=1, yMax=1500, Dy=10, yLog}
\plotxAxis{}
\plotyAxis{}

\listplot{\LogData}
\end{NumericDataPlot}

```



2. Multiple plots in one picture

xPicMin, xPicMax, yPicMin and yPicMax are the inner coordinates of one picture. The position of the axes are defined in this coordinate system via xCoordMin, xCoordMax, yCoordMin and yCoordMax.

Example:

```
% input data and define linestyles
\input{DataTestRealData}
\newsstyle{Database}{linecolor=LineColorA, linestyle=none, dotstyle=*,
showpoints=true, dotsize=5pt}
\newsstyle{Result}{linecolor=LineColorB, linestyle=none, dotstyle=+,
showpoints=true, dotsize=10pt}

\begin{center}
\begin{NumericDataPlot}[xPicMin=0, xPicMax=1050,
yPicMin=0, yPicMax=1450]{\textwidth}{0.75\textheight}

% --- definition of the axis and the grid ---
% set the axis of the lower left corner
\setxAxis{xMin=2, xMax=17, Dx=4, xCoordMin=0, xCoordMax=500}
\setyAxis{yMin=20, yMax=70, Dy=20, yCoordMin=0, yCoordMax=500}

% plot the axis of the lower left corner
\plotxAxis{Stichnummer}
\plotyAxis{$F_{\text{roll}}$ in $\text{mega}\text{newton}$}

\listplot[style=Database]{\DataA}
\listplot[style=Result]{\DataB}
\listplot[style=StdLineStyleC]{\DataC}
\listplot[style=StdLineStyleD]{\DataD}
\listplot[style=StdLineStyleE]{\DataE}
\listplot[style=StdLineStyleF]{\DataF}
\listplot[style=StdLineStyleG]{\DataG}

% set the y-axis for the plot in the middle of the left side
% x-axis remains the same
\setyAxis{yMin=20, yMax=70, Dy=20, yCoordMin=550, yCoordMax=1050}
% plot the axis (x-axis without ticklabels and label
\plotxAxis[NoTickLabel, NoLabel]{ }
\plotyAxis{$F_{\text{roll}}$ in $\text{mega}\text{newton}$}
```

```

\rput[t]{0}(250,550){a) Ein plot}

\listplot[style=Database]{\DataA}
\listplot[style=Result]{\DataB}

% set axis for the plot at the right side
\setxAxis{xMin=2, xMax=17, Dx=4, xCoordMin=600, xCoordMax=1050}
\setyAxis{yMin=20, yMax=70, Dy=10, yCoordMin=0, yCoordMax=1050}

% plot the axis at the right side (y-axis without label)
\plotxAxis{Stichnummer}
\plotyAxis[NoLabel, TickSep=10]{ $F_{\text{roll}}$  in  $\text{mega}\backslash\text{newton}$ }

\listplot[style=Database]{\DataA}
\listplot[style=Result]{\DataB}
\listplot[style=StdLineStyleC]{\DataC}
\listplot[style=StdLineStyleD]{\DataD}
\listplot[style=StdLineStyleE]{\DataE}
\listplot[style=StdLineStyleF]{\DataF}
\listplot[style=StdLineStyleG]{\DataG}

% set the axis for the plot at the top
\setxAxis{xMin=8, xMax=17, Dx=1, xCoordMin=0, xCoordMax=1050}
\setyAxis{yMin=35, yMax=65, Dy=10, y0=40, yCoordMin=1150, yCoordMax=1450}

% plot the axis for the plot at the top
\plotxAxis[NoLabel]{Stichnummer}
\plotyAxis[NoLabel]{ $F_{\text{roll}}$  in  $\text{mega}\backslash\text{newton}$ }

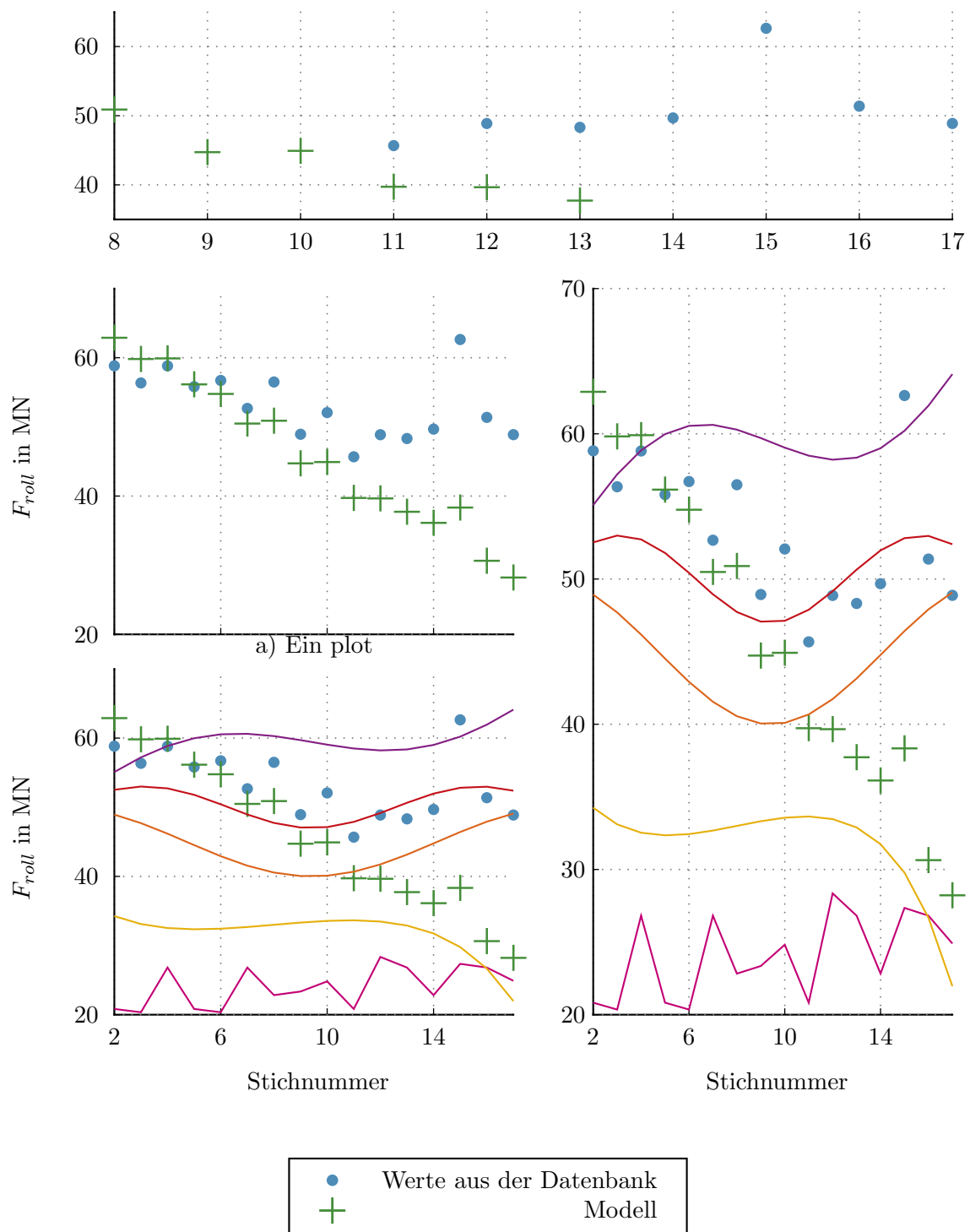
% plot only part of the data
\listplot[style=Database, xStart=11, xEnd=17]{\DataA}
\listplot[style=Result, xStart=8, xEnd=13]{\DataB}

\end{NumericDataPlot}

% put legend outside of the plot
\LegendDefinition[nrCols=1, LabelOrientation=r]{
\LegLine{Database} & Werte aus der Datenbank \\\
\LegLine{Result} & Modell}

\end{center}

```

3. Matlab Support

3.1. export2latex.m

The function `export2latex(data, filename, [options])` may be used to export data from Matlab to be used with `NumericPlots.sty`. The function takes the two parameters `data` and `filename`, where `filename` is the name of the file where the data should be stored with full path but without extension. `data` is a structure with the three entries `x`, `y`, and `ident`, where `x` are the x-data, `y` the y-data and `ident` is an identifier to have access to the data in Latex. To access the data in Latex the command `\Data<ident>` is used.

The parameter `data` may be provided as an `1xK` array if each entry of the array is a structure with the entries `x`, `y` and `ident`. There may be an additional entry `descr`.

- `data(1,i).x = X_i`
- `data(1,i).y = Y_i`
- `data(1,i).ident = Identifier`
- `(data(1,i).descr = Description)`

Identifier must be a valid Latex command name, which basically means it must be a string without numbers.

For every entry of `data`, `X` and `Y` must be arrays of the same size $M \times N$. If $M > 1$, each column of the array is exported to the file and the identifier is expanded by the roman representation of the column number.

It might be useful to do a `interp1` before exporting the data to get smaller files and to use less of \TeX 's memory. (\rightarrow script Patrick!?)

The optional parameter `options` might be used to control which additional information is provided in the output file.

- `options.DataBoundaries [true]` If set to true, the output file contains the commands `\DataXmin`, `\DataXmax`, `\DataYmin` and `\DataYmax` which may be used to define the axis.
- `options.AxisBoundaries [false]` If set to true, the output file contains the commands `\DataXminAxis`, `\DataXmaxAxis`, `\DataYminAxis` and `\DataYmaxAxis` which may be used to define the axis. In contrary to the `DataBoundaries` the `AxisBound-`

aries add an additional gap of `options.AxisBoundariesGap` [10] percent of the full scale to the data boundaries such that the plot doesn't touch the axis.

3.2. struct2latex.m

The function `struct2latex(data,[xname],[downsample],[filename],[postfix],[options])` may be used to export a Matlab struct to be used with `NumericPlots.sty`. The function is based on `export2latex` and the `options` argument is given to `export2latex`. The optional argument `xname` specifies the fieldname of the x-data. If no `xname` is given, you have to chose one via popup. All other fields are treated like y-data. Furthermore, the identifier of the y-data are equal to the fieldnames and have to be valid Latex commands (fieldnames with numbers are not supported). The `struct2latex` command only supports 1 dimensional arrays (row vectors). Column vectors are mapped for proper export without warning. With the optional argument `downsample` (integer \geq 0) it is possible to downsample the data for smaller files. A warning is given when data points exceed 5000. Usually 1000 data points are enough for a proper plot in latex. The optional argument `filename` specifies the outputname with path of the texfile. If no filename is given, it has to be chosen via popup. The optional argument `postfix` can be used to put a postfix after the fieldname in the identifier of the y-data. The `options` argument is also optional and equal to the `options` argument of the function `export2latex`.

Matlab example.

```
% data for struct2latex should be row-vectors. anyway, column-vectors are
% mapped to row-vectors. matrices are not supported

%% export row-vectors
t = [1:0.1:10];

data.time = t;
data.sin = sin(t);
data.cos = cos(t);
data.exp = exp(t);
data.tan = tan(t);

%struct2latex(data,xname,downsample,filename,postfix,options)
%downsample,filename,postfix,options are optional arguments
struct2latex(data,'time',1,'texdataA','row');
```

3.3. `dspace2struct`

The function `dspace2struct(dataname,version)` may be used to import `.mat` created by DSPACE to MATLAB. It returns a MATLAB struct out of the provided variables `dataname` and `version`. Thereby `dataname` is the name of the DSPACE `.mat` file and the optional string argument `version` has to be equal `version = 'NG'` if DSPACE CONTROL DESK NEXT GENERATION is used. It is convenient to chose a unique signal name in MATLAB/SIMULINK because they will be the same in DSPACE. The function `dspace2struct(dataname,version)` just adopts the names provided by DSPACE and use them as fieldnames. The time vector will always be called `.time`.

Matlab example.

```
dataOG = dspace2struct(measurements) %Old Control Desk  
dataNG = dspace2struct(measurements,'NG') %Control Desk Next Generation
```

3.4. `dspace2latex`

The function `dspace2latex` is just a combination of `dspace2struct` and `struct2latex` with commonly used arguments. It is a simple and quick way to export measurements from DSPACE to L^AT_EX without any revision of the data.

4. package options

5. list of commands and options

6. Options

6.1. General

These are the options for the environment `NumericDataPlot`.

- `xPicMin` [0], `yPicMin`[0], `xPicMax`[1000], `yPicMax`[1000]
- `TickLength` [2] defines the length of the ticks in mm
- `llx`[-1.5cm], `lly`[-1.25cm], `urx`[2pt], `ury`[2pt]

6.2. Automatic list with `xkvview`

The following list of keys defined by the package was created with `xkvview`.

Key	Prefix	Family	Type	Default
<code>xMax</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>xMin</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>Dx</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>dx</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>xCoordMax</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>xCoordMin</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>yMax</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>yMin</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>Dy</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>dy</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>yCoordMax</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>yCoordMin</code>	<code>NumericDataPlot</code>	Axis	command	[none]
<code>LabelRotation</code>	<code>NumericDataPlot</code>	<code>xAxis</code>	command	[none]
<code>LabelPos</code>	<code>NumericDataPlot</code>	<code>xAxis</code>	command	[none]
<code>TickSep</code>	<code>NumericDataPlot</code>	<code>xAxis</code>	command	[none]
<code>LabelOption</code>	<code>NumericDataPlot</code>	<code>xAxis</code>	command	[none]
<code>TickLabelOption</code>	<code>NumericDataPlot</code>	<code>xAxis</code>	command	[none]
<code>LabelRotation</code>	<code>NumericDataPlot</code>	<code>yAxis</code>	command	[none]
<code>LabelPos</code>	<code>NumericDataPlot</code>	<code>yAxis</code>	command	[none]
<code>TickSep</code>	<code>NumericDataPlot</code>	<code>yAxis</code>	command	[none]
<code>LabelOption</code>	<code>NumericDataPlot</code>	<code>yAxis</code>	command	[none]
<code>TickLabelOption</code>	<code>NumericDataPlot</code>	<code>yAxis</code>	command	[none]
<code>xLog</code>	<code>NumericDataPlot</code>	Axis	boolean	true

Continued on next page

Continued from previous page

Key	Prefix	Family	Type	Default
yLog	NumericDataPlot	Axis	boolean	true
yNearlyTight	NumericDataPlot	Axis	boolean	false
x0	NumericDataPlot	AxisWait	command	[none]
y0	NumericDataPlot	AxisWait	command	[none]
NoGrid	NumericDataPlot	xAxis	boolean	true
NoTicks	NumericDataPlot	xAxis	boolean	true
NoLabel	NumericDataPlot	xAxis	boolean	true
NoTickLabel	NumericDataPlot	xAxis	boolean	true
NoGrid	NumericDataPlot	yAxis	boolean	true
NoTicks	NumericDataPlot	yAxis	boolean	true
NoLabel	NumericDataPlot	yAxis	boolean	true
NoTickLabel	NumericDataPlot	yAxis	boolean	true
yLabelSep	NumericDataPlot	yAxis	ordinary	\origXLabelSep
xLabelSep	NumericDataPlot	xAxis	ordinary	\origYLabelSep
AxisStyle	NumericDataPlot	xAxis	choice	[none]
AxisStyle	NumericDataPlot	yAxis	choice	[none]
xPicMin	NumericDataPlot	General	command	[none]
yPicMin	NumericDataPlot	General	command	[none]
xPicMax	NumericDataPlot	General	command	[none]
yPicMax	NumericDataPlot	General	command	[none]
TickLength	NumericDataPlot	General	command	[none]
llx	NumericDataPlot	General	ordinary	\StdLLX
urx	NumericDataPlot	General	ordinary	\StdURX
lly	NumericDataPlot	General	ordinary	\StdLLY
ury	NumericDataPlot	General	ordinary	\StdURY
nrCols	NumericDataPlot	Legend	ordinary	1
LegLineWidth	NumericDataPlot	Legend	ordinary	20pt
LabelOrientation	NumericDataPlot	Legend	choice	[none]
x	NumericDataPlot	put	ordinary	[none]
y	NumericDataPlot	put	ordinary	[none]
RefPoint	NumericDataPlot	put	ordinary	c

7. Details

7.1. Coordinate Systems

This section should just give some hints how to use the different coordinate systems.

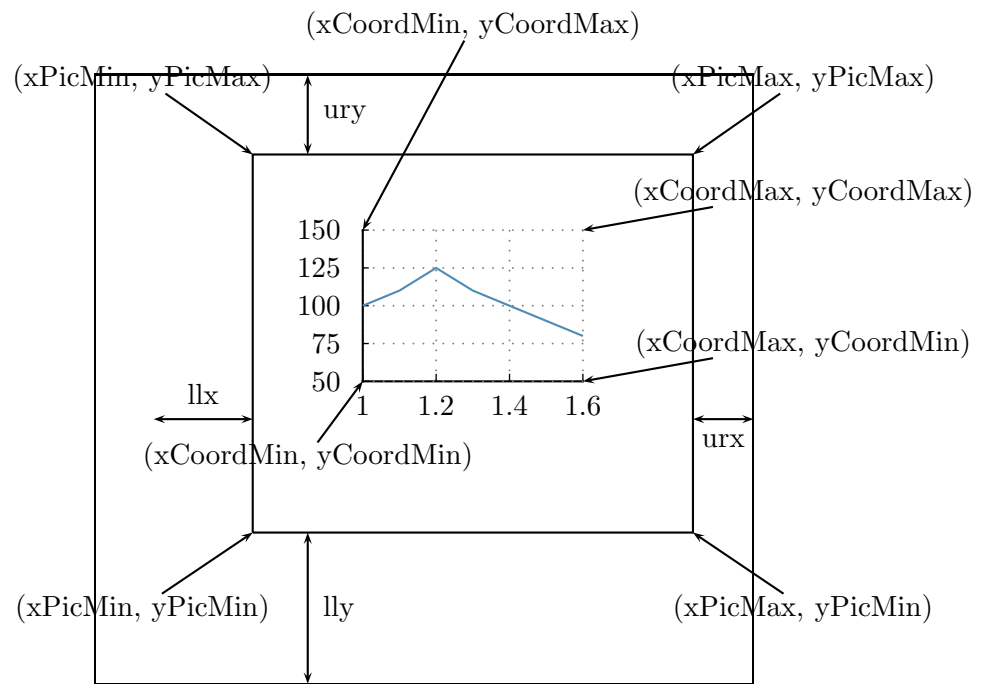
A new plot is created with the environment `NumericDataPlot`. The options are illustrated in the following example.

```
\begin{NumericDataPlot}
[xPicMin=300, xPicMax=700]
  {\textwidth}{5cm}
\setxAxis
  {xMin=1, xMax=1.6, Dx=0.2}
\setyAxis
  {yMin=50, yMax=150, Dy=25}

\plotxAxis[NoLabel]{}
\plotyAxis[NoLabel]{}

\listplot[style=StdLineStyleA]
  {\IdentI}

\NDPhline[linecolor=LineColorD]{73}
\NDPvline[linecolor=LineColorE, linestyle=dashed]{1.5}
\end{NumericDataPlot}
```

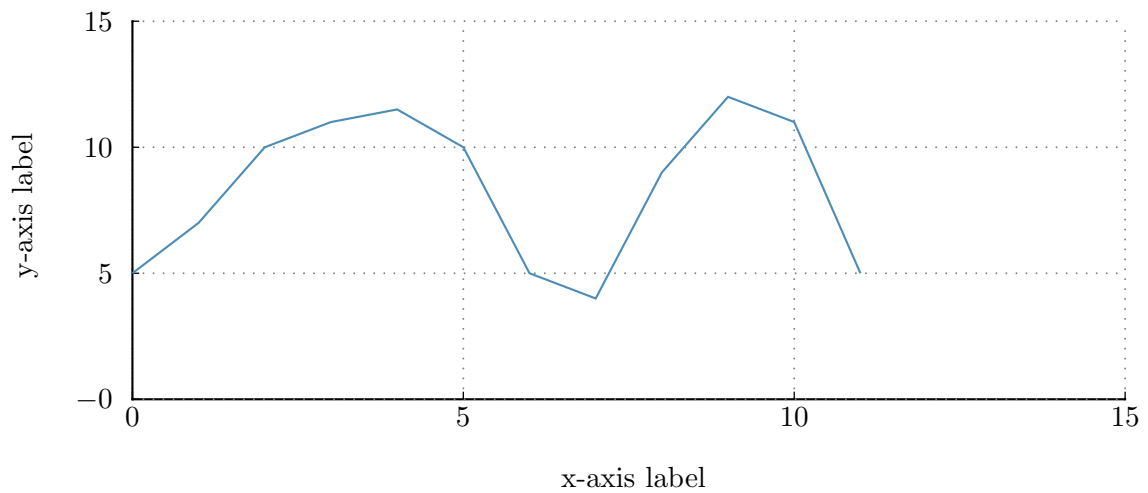
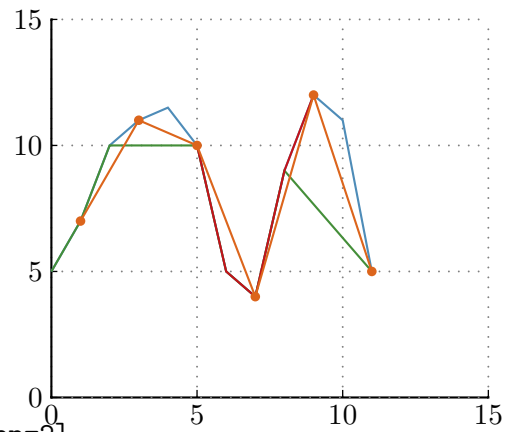


8. Further Examples

```

\listplot
[style=StdLineStyleA]
{\MoreData}
\listplot
[style=StdLineStyleB, yEnd=10]
{\MoreData}
\listplot
[style=StdLineStyleC, xStart=5, xEnd=9]
{\MoreData}
\listplot
[style=StdLineStyleD, showpoints=true, nStep=2]
{\MoreData}

```

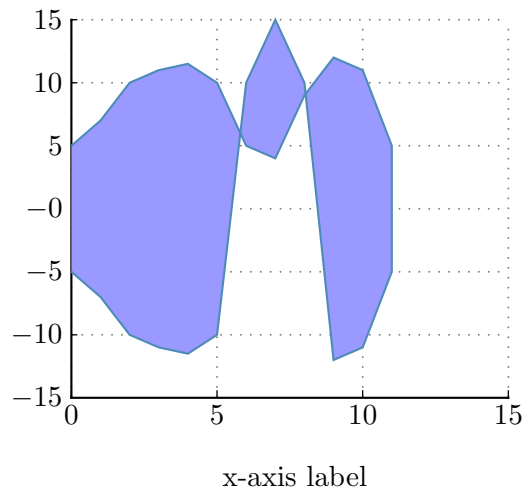


8.1. Fill area between plots

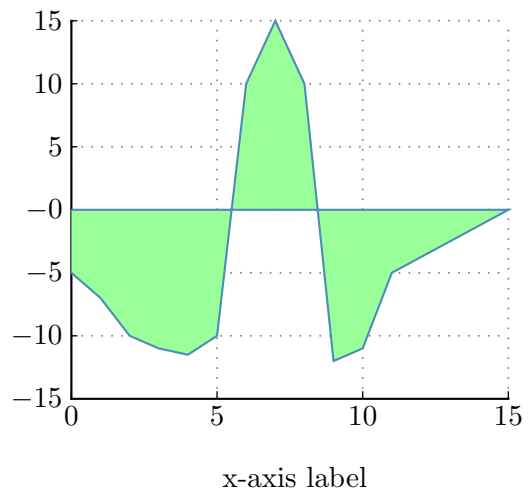
```
\begin{NumericDataPlot}%
{\textwidth}{5cm}
\setxAxis
{xMin=0, xMax=15, Dx=5, x0=0}
\setyAxis
{yMin=-15, yMax=15, Dy=5, y0=10}

\plotxAxis{x-axis label}
\plotyAxis[NoLabel]{}

\pscustom%
[style=StdLineStyleA, fillstyle=solid, %
fillcolor=blue!40]{%
\listplot{\MoreData}%
\listplot[ChangeOrder]{\MoreDataI}%
}
\end{NumericDataPlot}
```



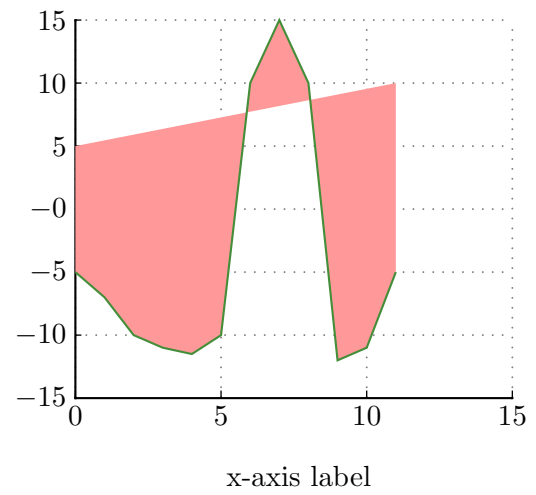
```
...
\pscustom%
[style=StdLineStyleA, fillstyle=solid, %
fillcolor=green!40]{%
\NDPhline{0}
\listplot[ChangeOrder]{\MoreDataI}%
}
...
```



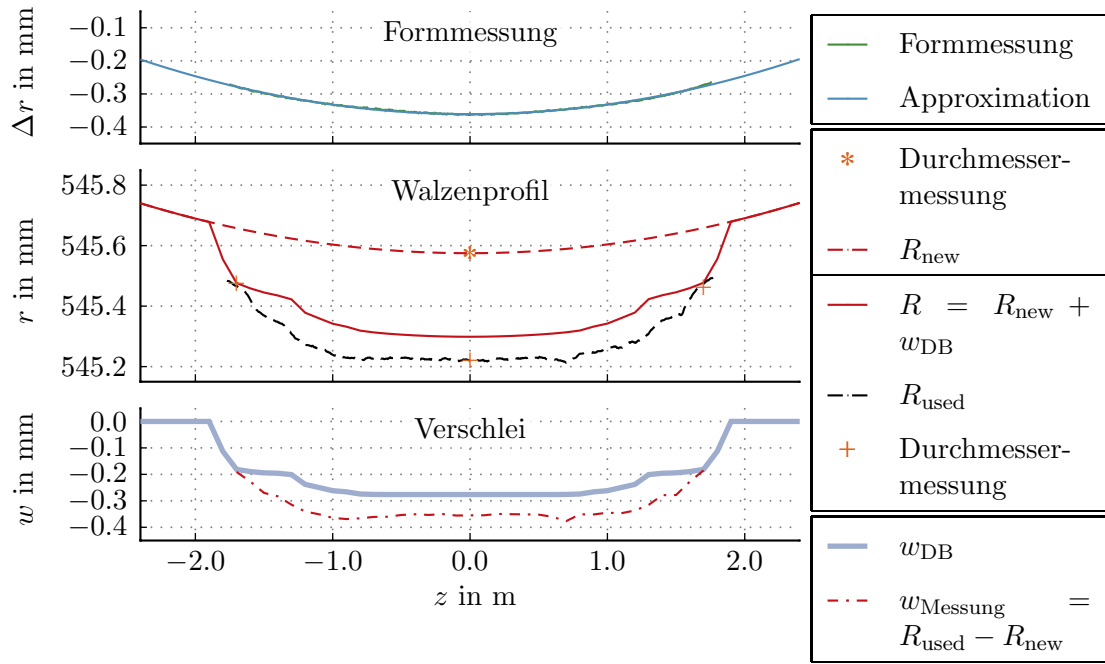
```

...
\pscustom%
[style=StdLineStyA, fillstyle=solid, %
fillcolor=red!40]{%
\NDPline{0}{5}{11}{10}
\listplot[ChangeOrder]{\MoreDataI}%
}
...

```



9. Some test plots



Part III.

Version

10. History

- 18.05.2010
 - added `LabelOption` to `plotAxis`
 - added `TickLabelOption` to `plotAxis`
 - added possibility to change `StdLabelOption`
 - added possibility to change `StdTickLabelOption`
- 11.06.2010
 - expanded documentation
 - changed the command `LegLine`.
 - added the commands `plotxGrid` and `plotyGrid`. The grids may now be plotted before the axis to avoid overlapping of the grid and the axis.
- 17.06.2010
 - changed `export2latex`: check for the necessary precision before exporting the data
- 21.06.2010
 - added option for logarithmic axes
- 05.07.2010
 - added command `NDPline`
 - added instructions to fill areas between plots to documentation
- 03.08.2010

- added options to `export2latex.m`
- 05.08.2010
 - new calculation of tick lengths
 - length of the ticks may now be given in mm, see 6.1
- 14.09.2010
 - added `struct2latex` to the package
 - added `dspace2struct` to the package
 - added `dspace2latex` to the package
 - changed `struct2latex`: mapping column vectors
 - changed `struct2latex`: downsampling option added
- 15.11.2010
 - changed the lineends of the axis
- 07.01.2011
 - adjusted linestyle
 - example with the available linestyle
- 04.05.2011
 - added commands to place exponents at the axes
 - added documentation for `put North, \dots, SouthWest`
- 14.05.2011
 - added command to put a box in the graph
- 17.05.2011
 - added ps-style `LegendBoxStyle` to adjust the style of the box underlying the legend

- 12.07.2011
 - changes in documentation
 - adopted `dspace2struct` for DSPACE NEXT GENERATION
 - added `\NDPvline`, `\NDPhline`, `\NDPvbox` and `\NDPhbox`