Todolist for PGFPlots

`1.11`

Christian Feuersänger

August 6, 2014

Contents

1 Release TODO

The following steps have to be done for every release:

- run tests against pgf CVS
- run tests against latest pgf stable
- finish manual
  - update the description for the `compat` key to the most recent version
  - update `pgfplots.preamble.tex` such that it contains —compat=—newest
  - fix warnings
  - browse through it
- ensure that the "compile" matrix (below this list) is complete for pgf CVS and the latest pgf Stable
- update ChangeLog with "RELEASED VERSION XXX" and update README
- assign git tag for new version

- run `scripts/pgfplots/pgfplotsrevisionfile.sh` to assign the package versions based on tag
- recompile manual (to pick up the correct version)
- run primitive example files
- run context tests (basically compile and browse-through. There are no assertions.)
  - run against pgf CVS
  - run against latest pgf stable
- adjust the release makefile to get correct file names
- assemble TDS.zip and CTAN.zip (using the release makefile)
  - `make -f pgfplots/scripts/pgfplots/Makefile.pgfplots_release_sourceforge` this creates the release files
  - make sure the archives do not contain wrong files.
- upload to CTAN.

```
Submitted to host
dante.ctan.org
Your name and email
    Christian Feuersnger <cfeuersaenger@users.sourceforge.net>
Filename
    pgfplots_1.6.1.ctan.flatdir.zip
Version number
    1.6.1
Location on CTAN
    /graphics/pgf/contrib/pgfplots/
Summary description
    pgfplots - Create normal/logarithmic plots in two and three dimensions for LaTeX.
License type
    gpl
Announcement text
...
Notes to maintainers
    As requested, I created a flat directory structure containing all files (i.e. it
```

- release to sourceforge. Make sure to update the README at top-level.
- upload to sourceforge web space (using the release makefile)
  - `make -f pgfplots/scripts/pgfplots/Makefile.pgfplots_release_sourceforge uploaddis` this copies the manuals
  - change `Makefile.pgfplots_release_sourceforge` back to unstable
  - `make -f pgfplots/scripts/pgfplots/Makefile.pgfplots_release_sourceforge upload` this updates the unstable
  - if necessary, update `scp://cfeuersaenger,pgfplots@web.sourceforge.net/htdocs/index.php`
- send announcement to `pgfplots-features@lists.sourceforge.net`

2 Tests

last test verifications:

	pgf CVS	pgf 3.0.0	pgf 2.10	pgf 2.00	pgf 2.00+compat=default
regressiontests	for 1.11	for 1.11	for 1.10	for 1.8 (7% fail)	2009-12-30
unittests	for 1.11	for 1.11	for 1.9		
manual	for 1.11	for 1.11	for 1.9	for 1.5	
pgfplotstable.pdf	for 1.11	for 1.10	for 1.9	for 1.5	
example latex	for 1.11		for 1.9	2009-12-30	
example context	for 1.11		for 1.9	2009-12-30	
example plain tex	for 1.11		for 1.9	2009-12-30	
tests context	for 1.11		for 1.9		

3 Documentation todo

pgfplotstodo.tex:128 **Documentation Todo** [open, Priority 5, Epic ‘’]

Document standalone vs. external

- standalone does not work with loops (compare `http://tex.stackexchange.com/questions/191108/how-to-avoid-copied-code-with-the-combination-pgfplots-standalone`)
- generating just one single figure without a document is simpler with standalone
- ...

pgfplotstodo.tex:133 **Documentation Todo** [open, Priority 5, Epic ‘’]

The new API for stacked plots + nodes near coords is missing, among it the normalized axis cs but see `\pgfplotspointgetnormalizedcoordinates`

pgfplotstodo.tex:168 **Documentation Todo** [open, Priority 5, Epic ‘’]

document installation requirements when using lualatex (LUAINPUTS should contain pgfplots install dir)

pgfplotstodo.tex:257 **Documentation Todo** [open, Priority 5, Epic ‘’]

bei dem Bsp-Tex zu pgfplotstable scheint eine Zeile im Tex-File zu fehlen: `\usepackage{pgfplotstable}`

Auerdem wre es zum Einstieg fr das aus der Datei lesen schn, wenn es zu den Daten auch ein kurzes Beispiel-File fr einen Plot gbe.

pgfplotstodo.tex:514 **Documentation Todo** [open, Priority 5, Epic ‘’]

try a bar plot with individually shaded bars

FIXME : collect details

pgfplotstodo.tex:530 **Documentation Todo** [open, Priority 5, Epic ‘’]

contour: a change label dist

pgfplotstodo.tex:563 **Documentation Todo** [open, Priority 5, Epic ‘’]

document ’execute at begin axis’ and its new variants

pgfplotstodo.tex:568 **Documentation Todo** [open, Priority 5, Epic ‘’]

document how to plot against the coordindex

pgfplotstodo.tex:581 **Documentation Todo** [open, Priority 5, Epic ‘’]

document how to identify the source of ”dimension too large” errors: tracingstuff.

pgfplotstodo.tex:596 **Documentation Todo** [open, Priority 5, Epic ‘’]

It seems as if the AMS command `$\text{\ref{ref:to:a:plot}}$` instantiates the `\ref` at least four times. Document somehow that it is better to use ’`\hbox`’ instead

pgfplotstodo.tex:605 **Documentation Todo** [open, Priority 5, Epic ‘’]

clickable lib: I have the impression that acroread fires warnings only for the manual - not always when the clickable lib is used. Why!?

4 Bugs/Features in PGF/TikZ

pgfplotstodo.tex:615 **pgfbug** [open, Priority 5, Epic ‘’]

external lib does not properly recompile unresolved references in external files, see `http://texwelt.de/wissen/fragen/7948/markierung-in-pgfplotlegende-fehlerhaft-ubernommen?Seite=1#7973` pgfplotstodo.tex:651 **pgfbug** [open, Priority 5, Epic ‘’]

When reading the manual v2.0 I found a typo 5.1 ”Styling the nodes”. Just after the first block of code, there is a sentence saying ”... can achieve them. Once way is to use ...” which should be ”One way is to use ...”

pgfplotstodo.tex:655 **pgfbug** [open, Priority 5, Epic ‘’]

Beamer + pgf: the default template introduces a white line on top. Interestingly, it happens only for PGF CVS + beamer, but it appears to be dependent on third-party tools as well (see mail conversation with Stefan Tibus)

pgfplotstodo.tex:677 **pgfbug** [open, Priority 5, Epic ‘’]

When using externalize function together with a transform canvas, the result is somehow croped. See this example, compare output with deativated and activated externalize.

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{external}
\tikzexternalize % activate!
\begin{document}
\begin{tikzpicture}[transform canvas={scale=0.7}]
\node {root}
child {node {left}}
child {node {right}
child {node {child}}
child {node {child}}
};
\end{tikzpicture}
A simple image is \tikz \fill (1,0) circle(5pt);.
\end{document}
```

pgfplotstodo.tex:734 **pgfbug** [open, Priority 5, Epic ‘’]

pgf users Vol 50 issue 6:

```
Hi,

Thanks for TikZ.  I'm trying to use the externalization library with
the class file gOMS2e.cls, which is provided for the journal
Optimization Methods and Software.  The class file and related
files/documentation can be found here:
http://www.tandf.co.uk/journals/authors/gomslatex.zip
```

My problem is that the externalized figures are shifted up and to the left significantly, cutting them off. This problem does not occur when not using externalization.
This seems to be related to the problem discussed here:
http://sourceforge.net/tracker/index.php?func=detail&aid=3037831&group_id=142562&atid=7527
and may also be related to this one:
http://sourceforge.net/mailarchive/forum.php?thread_name=4C0F342B.5040008%40ins.uni-bonn.d

In the other cases, the solution was to use \tikzifexternalizing for whatever conflicts with the externalization, but it seems that I can't do this when my class file is the offending bit. Is this true? I would really like to be able to use the correct \documentclass to generate the figures so that the size/fonts/etc. are consistent throughout the resulting document.

A minimal test example is included at the end of this message. It appears that the image is shifted ~1.25cm to the left and ~0.8cm up.
The problem goes away when using \documentclass{article}.
I'm using the CVS version of pgf, and I get the same result when I produce postscript figures by using latex and setting

```
 \tikzset{external/system call={
   latex \tikzexternalcheckshellescape -halt-on-error
-interaction=batchmode -jobname "\image" "\texsource";
   dvips -o "\image".ps "\image".dvi}}
```

```
%-------------------------------------------------------------------------------
\documentclass[printer]{gOMS2e}
\usepackage{tikz}
\usetikzlibrary{external}
\tikzexternalize
\begin{document}
\begin{center}
  \begin{tikzpicture}
    \draw[step=.5cm] (-3,-3) grid (3,3);
    \draw[blue,line width=2mm] (-0.5,-3) -- (-0.5,1.2) -- (3,1.2);
  \end{tikzpicture}
\end{center}
\end{document}
%-------------------------------------------------------------------------------
```

Any help would be appreciated; I'm afraid it's over my head at this point.
Thanks!

pgfplotstodo.tex:772 **pgfbug** [open, Priority 5, Epic ‘’]

```
\documentclass{article}
```

```
\usepackage{german}
\usepackage[utf8]{inputenc} % erlaubt direkte Nutzung von Umlauten

\usepackage{pgfplots} % fuer plots

\usepackage{pgfplotstable} % fuer numeriktabellen
\usepackage{array,colortbl,booktabs}
\usetikzlibrary{external}
\tikzexternalize[force remake]

% DOESN'T WORK. Needs to disable externailization
\usepackage{vmargin}
\setpapersize{A4}
\setmarginsrb{2.5cm}{1cm}{2cm}{2cm}{8mm}{15mm}{5mm}{15mm}

\begin{document}
\begin{tikzpicture}
%\tracingmacros=2 \tracingcommands=2
\begin{axis}
\addplot {x};
\end{axis}
\end{tikzpicture}
\end{document}
```

pgfplotstodo.tex:792 **pgfbug** [open, Priority 5, Epic ‘’]

external bug:

```
\documentclass[
    pagesize=auto,                 % 1
        ]{scrbook}
\usepackage{tikz}
    \usetikzlibrary{external}
        \tikzexternalize
\begin{document}
        \KOMAoption{twoside}{semi}  % 2
            test
    \tikz \draw (0,0) circle (3pt);
\end{document}
```

pgfplotstodo.tex:799 **pgfbug** [open, Priority 5, Epic ‘’]

consider a matrix style which applies only to the outer matrix node style (see feature request
https://sourceforge.net/tracker/?func=detail&atid=1060657&aid=3019259&group_id=224188
)

pgfplotstodo.tex:803 **pgfbug** [open, Priority 5, Epic ‘’]

make assignments to `\pgf@x` and `\pgf@y` always `\global`

pgfplotstodo.tex:807 **pgfbug** [open, Priority 5, Epic ‘’]

implement `\pgfmathfloattocount`

pgfplotstodo.tex:861 **pgfbug** [open, Priority 5, Epic ‘’]

the fpu can't be used inside of paths. That should be fixed. ⇝ the problem is that paths may use `\pgfmath...` routines directly. ⇝ this should work! At least with the public math macros `\pgfmathadd`. The `\pgfmathadd@` might be implemented differently.

pgfplotstodo.tex:914 **pgfbug** [open, Priority 5, Epic ‘’]

active '`|`' characters result in compilation bugs (`\usepackage{program}`)

pgfplotstodo.tex:918 **pgfbug** [open, Priority 5, Epic ‘’]

'text height=1em' realisieren mit [node font units]1em

5 Bugs in PGFPlots

pgfplotstodo.tex:1018 **Bug** [open, Priority 5, Epic ‘’]

REGRESSION: pgf image externalization is broken, see comments in `https://sourceforge.net/p/pgf/bugs/229/`

pgfplotstodo.tex:1075 **Bug** [open, Priority 5, Epic ‘’]

pgfplots fails to handle very small number ranges on logarithmic plots (i.e. if, say, xmin and xmax are very very close to each other).

see `http://www.mrunix.de/forums/showthread.php?76399-tikz-PGFplots-Fehler-dimension-too-large&p=358818#post358818`

pgfplotstodo.tex:1081 **Bug** [open, Priority 5, Epic ‘’]

3d axes and y dir = reverse seems to corrupt the z ordering, see `http://texwelt.de/wissen/fragen/6131/pgfplots-3d-oberflachenproblem`

pgfplotstodo.tex:1090 **Bug** [open, Priority 5, Epic ‘’]

ybar interval does not work well with nodes near coords

ybar interval has an artifical last coordinate. This will also receive nodes near coords.

See `http://tex.stackexchange.com/questions/181061/pgfplots-clipping-everything-outside-a-specific-area`

pgfplotstodo.tex:1096 **Bug** [open, Priority 5, Epic ‘’]

ybar interval + nodes nears coords: nodes are displaced.

The shift does not work here.

pgfplotstodo.tex:1104 **Bug** [open, Priority 5, Epic ‘’]

Coordinate filtering + error bars do not work.

Expected behavior: error coordinates should be computed and coord filters should be aplied to the result. Currently, filters are evaluated first and then error coordinates.

See `http://tex.stackexchange.com/questions/154084/how-to-scale-both-data-and-error-bars-in-pgfplots`

pgfplotstodo.tex:1110 **Bug** [open, Priority 5, Epic ‘’]

Old sampling no-go bug:
`\pgfplotsforeachungrouped \x in {0,1,...,5000}` is numerically instable (because I hesitated to do expensive float comparisons all the time...)

`http://tex.stackexchange.com/questions/183369/tikz-and-pgfplot-problem-with-plotting`

pgfplotstodo.tex:1115 **Bug** [open, Priority 5, Epic ‘’]

fillbetween / intersection lib bug: see minimal at `http://tex.stackexchange.com/questions/180899/fillbetween-library-sometimes-works-and-sometimes-does-not/181224#181224`

pgfplotstodo.tex:1119 **Bug** [open, Priority 5, Epic ‘’]

`|legend`to name— appears to be broken if the legend is empty (empty matrix?) TODO: verify that empty legends do not cause problems here!

pgfplotstodo.tex:1123 **Bug** [open, Priority 5, Epic “]

cube/size x has no effect if used in a 2d axis

pgfplotstodo.tex:1256 **Bug** [open, Priority 5, Epic “]

copy paste from the manual does not copy empty lines — which is quite bad for the 3d surface examples

This appears to be a known issue for pdf, with few available work-arounds. The listings package suffers from it as well (at least regarding trailing spaces/tabs); there are a couple of related questions at tex.sx

A way might be to use `\usepackage{accsupp}` with something like

```
\documentclass{article}

\usepackage{accsupp}

\begin{document}
\long\def\XXX{A
^^J
  ^^J
B}
\BeginAccSupp{method=escape,ActualText=\XXX}
First line

second line
\EndAccSupp{}

\end{document}
```

Expected result: copy-pasting that segment should yield A (newline) B. But it does not work with newlines as it seems

One entry on tex.sx mentioned that Microsoft products can insert empty lines in copy-paste output (suggested solutions there was to generate such a pdf and include it as graphics unless I am mistaken)

I managed to get this up and running so far – but the result is unusable! both acrobat and xpdf get utterly confused unless one hits exactly on some invisible lines [see `bugtracker/minimal_0.pdf`]

```
\pdfcompresslevel=0
\documentclass{article}
\usepackage{pgfplots}
\usepackage{accsupp}
\pgfplotsset{compat=1.7}

\begin{document}
```

Test before

```
\begingroup
\catcode`\ =12\relax%
\catcode`\^^M=13\relax%
\def^^M{^^J}%
\def\begin{\string\begin}%
\def\end{\string\end}%
\def\addplot{\string\addplot}%
\xdef\copypastable{%
\begin{tikzpicture}
    \begin{axis}
        \addplot3[surf] coordinates {
            (0,0,0) (1,0,0) (2,0,0) (3,0,0)

            (0,1,0) (1,1,0.6) (2,1,0.7) (3,1,0.5)

            (0,2,0) (1,2,0.7) (2,2,0.8) (3,2,0.5)
        };
    \end{axis}
\end{tikzpicture}%
}\endgroup

\BeginAccSupp{%
               method=escape,%
               ActualText={\copypastable}}
\begin{verbatim}
\begin{tikzpicture}
    \begin{axis}
        \addplot3[surf] coordinates {
            (0,0,0) (1,0,0) (2,0,0) (3,0,0)

            (0,1,0) (1,1,0.6) (2,1,0.7) (3,1,0.5)

            (0,2,0) (1,2,0.7) (2,2,0.8) (3,2,0.5)
        };
    \end{axis}
\end{tikzpicture}
\end{verbatim}
\EndAccSupp{}%
```

Text middle

```
\begingroup
\catcode`\ =12\relax%
\catcode`\^^M=13\relax%
\def^^M{^^J}%
\def\begin{\string\begin}%
\def\end{\string\end}%
\def\addplot{\string\addplot}%
\xdef\copypastable{%
\begin{tikzpicture}
    \begin{axis}[small,view={0}{90}]
    \addplot3[surf,shader=interp,patch type=bilinear,
        mesh/color input=explicit]
    coordinates {
        (0,0,0) [color=blue] (1,0,0) [color=green]

        (0,1,0) [color=yellow] (1,1,1) [color=red]
    };
    \end{axis}
\end{tikzpicture}
}\endgroup

\BeginAccSupp{%
              method=escape,%
              ActualText={\copypastable}}
\begin{verbatim}
\begin{tikzpicture}
   \begin{axis}[small,view={0}{90}]
   \addplot3[surf,shader=interp,patch type=bilinear,
      mesh/color input=explicit]
   coordinates {
      (0,0,0) [color=blue] (1,0,0) [color=green]

      (0,1,0) [color=yellow] (1,1,1) [color=red]
   };
   \end{axis}
\end{tikzpicture}
\end{verbatim}
\EndAccSupp{}%

Text after
\end{document}
```

Problems here:

- one has to mark the right segments, and these can only be found by trial and error
- hitting the wrong one will copy nothing or only parts
- in the manual, I often had the wrong content in my clipboard (some older state of the clipboard, no idea where that came from)

pgfplotstodo.tex:1260 **Bug** [open, Priority 5, Epic ‘’]

tikz intersections: the “store in macro” methods do not use global macros. Should they!?

pgfplotstodo.tex:1267 **Bug** [open, Priority 5, Epic ‘’]

fill between:

```
addplot [draw=green!80,fill=green!20] fill between [of=lower and axis, soxft clip={(inters
```

produces no “unknown key soxft clip” error

pgfplotstodo.tex:1305 **Bug** [open, Priority 5, Epic ‘’]

fill between / intersections lib bug: spurious intersections
`http://tex.stackexchange.com/questions/180127/stacking-plots-in-animation-using-fill-between-library-with-dynamic-calculation/180299#180299`

analysis in ~/tmp/bug_intersectionlibs.tex: problem seems to be the duplicate detection

problem: almost parallel lines:

```
\begin{tikzpicture}
\makeatletter
 \def\A{%
  \pgfsyssoftpath@movetotoken{15.60165pt}{0.60783pt}%
\pgfsyssoftpath@linetotoken{0pt}{0pt}%
}%
 \def\B{%
  \pgfsyssoftpath@movetotoken{15.60165pt}{0.81604pt}%
\pgfsyssoftpath@linetotoken{0.0pt}{0.33199pt}%
}%
\draw[blue] \pgfextra{\pgfsetpathandBB\A};
\draw[red] \pgfextra{\pgfsetpathandBB\B};
    \pgfintersectionofpaths%
        {%
          \pgfsetpath\A
        }%
        {%
          \pgfsetpath\B
        }%

\pgfmathloop%
\ifnum\pgfmathcounter>\pgfintersectionsolutions\relax%
\else%
\draw[red] \pgfextra{\pgftransformshift{\pgfpointintersectionsolution{\pgfmathcounter}\mes
node[anchor=center] {\pgfmathcounter/\pgfintersectionsolutions};
\repeatpgfmathloop%

 \end{tikzpicture}
```

pgfplotstodo.tex:1309 **Bug** [open, Priority 5, Epic ‘’]

pgfplotstable appears to have problems with # in col names, compare http://tex.stackexchange.com/questions/175395/how-can-csv-files-be-put-into-latex-without-having-to-adjust-much

pgfplotstodo.tex:1632 **Bug** [open, Priority 5, Epic ‘’]

box plots appear to have a bug when the number of coordinates / duplicates is special: http://tex.stackexchange.com/questions/146663/pgfplots-fails-to-generate-boxplots-for-some-data-set?noredirect=1#comment332779_146663

pgfplotstodo.tex:1640 **Bug** [open, Priority 5, Epic ‘’]

`http://tex.stackexchange.com/questions/104010/why-does-loading-pgfplots-after-tikz-break-the-default-layers-in-a-tikzpicture` is still active .

Solution: backgrounds lib must be loaded after pgfplots. Apparently, the backwards layer stuff overwrites hooks of the backgrounds lib

pgfplotstodo.tex:1673 **Bug** [open, Priority 5, Epic ‘’]

default tick labels are placed outside of displayed area if `unit vector ratio*=1 1` is in effect

```
\documentclass{article}
\usepackage{pgfplots}
\usetikzlibrary{decorations.markings}
\pgfplotsset{compat=1.8}

\begin{document}

\begin{tikzpicture}
\begin{axis}[axis lines = middle,smooth,xlabel = $x$, ylabel =$y$, minor tick num =1, grid
\addplot[smooth, thick, -stealth,variable=\t, domain=0:2, ]
({t^2}, {t^4});

\addplot[thick, red,-stealth,samples=8,variable=\t, domain=0:2,quiver={
u=2*t, v=4*t^3, scale arrows=0.05,
}]
({t^2}, {t^4});
\end{axis}
\end{tikzpicture}

\end{document}
```

pgfplotstodo.tex:1686 **Bug** [open, Priority 5, Epic ‘’]

`|ytick=data|` does not work together with stacked plots because, by design, it merely considers coordinates of the *first* plot. That’s nonsense for stacked plots.

pgfplotstodo.tex:1710 **Bug** [open, Priority 5, Epic ‘’]

Tick labels suffer from inexact arithmetics in Tikz foreach:

```
KAPUTT:
```

```
\foreach \x in {1,1.1,...,2} {\x\par}
```

OK:

```
\foreach \x in {1,1.1,...,2.001} {\x\par}
```

pgfplotstodo.tex:1714 **Bug** [open, Priority 5, Epic ‘’]

ytick=data combined with minor y tick num does not work as expected: minor tick lines will be skipped below the lowest limit.

pgfplotstodo.tex:1720 **Bug** [open, Priority 5, Epic ‘’]

pgfplotstable does not apply `postproc cell content` in `every row` styles

see `http://tex.stackexchange.com/questions/111492/postprocess-row-with-pgfplotstable`

pgfplotstodo.tex:1726 **Bug** [open, Priority 5, Epic ‘’]

no markers: it appears to be impossible to switch off markers, only for scatter plots.

See `http://tex.stackexchange.com/questions/105850/pgfplots-points-with-no-marks-but-errorbars`

pgfplotstodo.tex:1768 **Bug** [open, Priority 5, Epic ‘’]

`|...|`and rounding:

Generell erscheint mir die `...`-Syntax nicht robust. Bei kleinen, durchaus nicht ungewhnlichen, Intervallen kommt es zu Rundungsfehlern:

Hauptgitter:

`ytick={0.99,1.00,1.01,1.02,1.03,1.04,1.05,1.06,1.07,1.08}`

Hilfsgitter:

`minor ytick={0.99,0.991,...,1.08}`

Fhrt dazu, dass die Hilfslinien mit zunehmender Gre immer weiter von den Hauptgitterlinien verschoben sind, obwohl diese aufeinanderliegen sollten. Mit Hilfsgitter komplett ausgeschrieben:

`minor ytick={0.99,0.991,0.992,0.993,..............,1.08}` passt es. Die `...`-Syntax ist daher absolut mit Vorsicht zu genieen.

pgfplotstodo.tex:1772 **Bug** [open, Priority 5, Epic ‘’]

box plots with draw direction=y and xtick=data fail to provide the correct xtick locations

pgfplotstodo.tex:1797 **Bug** [open, Priority 5, Epic ‘’]

presets for mark size and tiny / footnotesize are wrong

pgfplotstodo.tex:1815 **Bug** [open, Priority 5, Epic ‘’]

clipping of tick lines does not respect the line width of the axis lines.

`http://tex.stackexchange.com/questions/91517/how-to-make-the-tick-thickness-as-the-axis-line/91645#91645`

pgfplotstodo.tex:1839 **Bug** [open, Priority 5, Epic ‘’]

Inf geht nicht im math parser:

```
\documentclass[a4paper]{article}

\usepackage{pgfplots}


\begin{document}

\pgfmathfloatparsenumber{Inf}
\pgfmathresult

{
\pgfkeys{/pgf/fpu}
\pgfmathparse{Inf}
\pgfmathresult
}

\end{document}
```

pgfplotstodo.tex:1858 **Bug** [open, Priority 5, Epic ‘’]

`|disablelogfilter,`ymax=1e-6, ymode=log— fails. Apparently, the coordinate is not parsed at all.

pgfplotstodo.tex:1870 **Bug** [open, Priority 5, Epic ‘’]

|\closedcycle|does not work together with jumps / interrupted plots

pgfplotstodo.tex:1894 **Bug** [open, Priority 5, Epic ‘’]

auto tick label assignment can sometimes produce strange results:

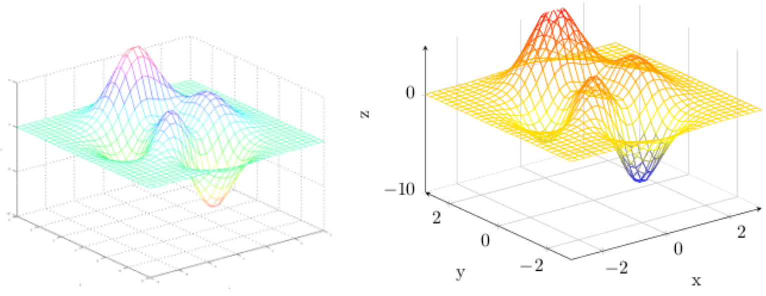
[see `bugtracker/minimal_1.pdf`]

```
\documentclass{article}
\usepackage{pgfplots}
\pgfplotsset{compat=1.6.1}

\begin{document}

\begin{tikzpicture}
\begin{axis}[%
scale only axis,
xmin=0, xmax=0.02,
ymin=-1, ymax=1]
\end{axis}
\end{tikzpicture}%
\end{document}
```

See also the examples for boxplots in the manual

pgfplotstodo.tex:1936 **Bug** [open, Priority 5, Epic ‘’]

the3d clip path is sometimes bad: perhaps it should be the bounding box instead!?



reference rendering

generated

Figure 44: Mesh plot. `meshPlot`

pgfplotstodo.tex:1968 **Bug** [open, Priority 5, Epic ‘’]

One cannot manually load a table inside of a pgfplots axis if it contains empty lines: the scanline callback is active `pgfplotstodo.tex:1972` **Bug** [open, Priority 5, Epic ‘’]

`|ybar`legend— does not contain `ybar` (sourceforge 3482770)

`pgfplotstodo.tex:1976` **Bug** [open, Priority 5, Epic ‘’]

`|refstyle|`does only include partial information of the reference style. (sourceforge 3482770)

`pgfplotstodo.tex:1982` **Bug** [open, Priority 5, Epic ‘’]

Using square brackets as first char inside of `\legend` leads to a failure: `\legend{[\ion{Ne}{2}],...}`.

Cause: the `\legend` command does not properly insert `[]` in front of every entry (as it ought to).

`pgfplotstodo.tex:1988` **Bug** [open, Priority 5, Epic ‘’]

Adding `error bars/.cd` to `\addplot` options causes the `\ref` image to fail.

The problem is the key filtering apparently: it discards the `/pgfplots/.cd` but leaves the `error bars/.cd`.

`pgfplotstodo.tex:1992` **Bug** [open, Priority 5, Epic ‘’]

quiver plots: the clip path does not respect arrow paths

`pgfplotstodo.tex:2049` **Bug** [open, Priority 5, Epic ‘’]

Alignment bug: axis x line=middle combined with a yshift shifts the xlabel incorrectly: [see `bugtracker/minimal_2.pdf`]

```
\documentclass{report}
\usepackage{pgfplots}
\pgfplotsset{compat=1.3}

\begin{document}
  \begin{tikzpicture}
        \draw (0,0) circle (5pt);
    \begin{scope}[yshift=-3cm]
    \begin{axis}[width=10cm,height=3cm,xlabel={$x$},
        axis x line = middle]
      \addplot coordinates {
        (0,1) (1,-1) (2,1)
      };
    \end{axis}
    \end{scope}
  \end{tikzpicture}
\end{document}
```

Using `xlabel style = {yshift=3cm}` in the plot will correctly position the x label (to its default position).

pgfplotstodo.tex:2126 **Bug** [open, Priority 5, Epic ‘’]

cannot provide clip path usage in pgfplots commands because of the nested scopes.

to reproduce, try to give `\addplot+[/tikz/clip]` to some plot.

pgfplotstodo.tex:2134 **Bug** [open, Priority 5, Epic ‘’]

`|\pgfplotsforeachungrouped|`cannot be combined with three or more arguments like `\foreach`

pgfplotstodo.tex:2233 **Bug** [open, Priority 5, Epic ‘’]

the below example of a latex file gives the following error upon the 2nd run of latex. The first run works fine. This happens both when running dvilualatex and just latex, both from TexLive 2011.

The error:

```
...
(/usr/local/texlive/2011/texmf-dist/tex/generic/tex4ht/color.4ht)
(/usr/local/texlive/2011/texmf-dist/tex/generic/tex4ht/html4.4ht)
(/usr/local/texlive/2011/texmf-dist/tex/generic/tex4ht/html4-math.4ht))
(./epub.aux)
! Missing \endcsname inserted.
<to be read again>
                   \protect
l.30 \ref{govconsumptionlegend}

?


\makeatletter

\def\HCode{\futurelet\HCode\HChar}\def\HChar{\ifx"\HCode\def\HCode"##1"{\Link##1}\expandaf

\makeatother

\HCode "xhtml,png,charset=utf-8".a.b.c.


\documentclass[11pt,a4paper]{book}

\def\pgfsysdriver{pgfsys-tex4ht.def}
```

```
\usepackage{pgfplots}

\pgfplotsset{width=\textwidth,compat=1.3,every axis/.append style={font=\footnotesize},cyc

\begin{document}
\begin{tikzpicture}

\begin{axis}[ylabel=\%,x tick label style={ /pgf/number format/1000 sep=},ymin=0,xmin=1950
\addplot[smooth,solid] coordinates {
(1950,12.98732304) (1951,11.18937899) (1952,10.63447043) (1953,11.25741618) (1954,11.35201
};
\addlegendentry{Country 1}
\addplot[smooth,dotted] coordinates {
(1950,8.90574995) (1951,9.181850378) (1952,9.4040808) (1953,9.790597533) (1954,9.766571438
};
\addlegendentry{Country 2}
\end{axis}\end{tikzpicture}

\ref{govconsumptionlegend}

\end{document}
```

pgfplotstodo.tex:2482 **Bug** [open, Priority 5, Epic ‘’]

http://groups.google.at/group/comp.text.tex/msg/adcb1d071c2cba40

If I use a yshift in a scope to draw two graphs superimposed, the x label in the second plot (the one in the yshift scope) is not positioned correctly. I need to manually add another yshift, with the same value in the opposite direction, to get the label at the correct place. This happens if the `axis x line = middle` option is used. Without that option, the x label is positioned correctly.
Example follows: [see `bugtracker/minimal_3.pdf`]

```
\documentclass{article}

\usepackage{pgfplots}

\pgfplotsset{compat=1.3}

\begin{document}

  \begin{tikzpicture}
    \begin{axis}[width=10cm,height=3cm,xlabel={$x$}]
      \addplot coordinates {
        (0,1) (1,-1) (2,1)
      };
    \end{axis}
    \begin{scope}[yshift=-3cm]
    \begin{axis}[width=10cm,height=3cm,xlabel={$x$},
        axis x line = middle]
      \addplot coordinates {
        (0,1) (1,-1) (2,1)
      };
    \end{axis}
    \end{scope}
  \end{tikzpicture}
\end{document}
```

Using `xlabel style = {yshift=3cm}` in the second plot will correctly position the x label (to its default position).

Gab

pgfplotstodo.tex:2658 **Bug** [open, Priority 5, Epic ‘’]

after using a preset key (milli) with x SI prefix, Next, I want to switch to the normal mode, so I write simply: x SI prefix=none, unfortunately the ’none’ value is undefined and the compilation can not proceed

pgfplotstodo.tex:2724 **Bug** [open, Priority 5, Epic ‘’]

check y tick scale label for 2nd y axis

pgfplotstodo.tex:2741 **Bug** [open, Priority 5, Epic ‘’]

groupplots + extra braces or foreach are incompatible.

pgfplotstodo.tex:2745 **Bug** [open, Priority 5, Epic ‘’]

numplotspertype and forget plot and ybar interval yields errors.

pgfplotstodo.tex:2751 **Bug** [open, Priority 5, Epic ‘’]

expression plotting and empty ’y’ results in errors. Perhaps it would be better to handle that explicitly somehow? (occurs for hist when one input line is empty)

pgfplotstodo.tex:2771 **Bug** [open, Priority 5, Epic ‘’]

/pgfplots/samples at and /tikz/samples at work on the same axe. Tantau says that this key support foreach statement and thus the dotes notation. However, when I want to use two or more different dots notation within pgfplots, latex crashes ! Here is an example which clarify this issue :
`\addplot+[mark=none,variable={\t}, samples at = {\foreach \x in {0,10,...,180,200,...340}`

pgfplotstodo.tex:2781 **Bug** [open, Priority 5, Epic ‘’]

groupplots: mixing 2d/3d in one groupplot doesn’t reset ’zmin,zmax’ ?

pgfplotstodo.tex:2802 **Bug** [open, Priority 5, Epic ‘’]

3D axes: it is difficult to get an 1:1 correspondence to tikz.

pgfplotstodo.tex:2818 **Bug** [open, Priority 5, Epic ‘’]

3D axis: provide support for manual axis configuration, - depth (n vector), - foreground/background, - tick label axes, - ...

pgfplotstodo.tex:2847 **Bug** [open, Priority 5, Epic ‘’]

plot graphics: `\ref` legend doesn’t work properly

pgfplotstodo.tex:2868 **Bug** [open, Priority 5, Epic ‘’]

ternary axes: the ’marker clipping’ doesn’t work (naturally)

pgfplotstodo.tex:2878 **Bug** [open, Priority 5, Epic ‘’]

polar axes:

- ✓is wrong since ’near ticklabel’ anchor uses pointunitx which is not correctly initialised for polar axes.
- axis equal
- ✓data scaling needs to be disabled for X axis.
- ✓auto tick labels work only for the case of disabledatascaling

pgfplotstodo.tex:2883 **Bug** [open, Priority 5, Epic ‘’]

contour: the table/meta=2 default is wrong.

pgfplotstodo.tex:2896 **Bug** [open, Priority 5, Epic ‘’]

dimension too large sanity checking: TeX uses the maximum value instead. Perhaps that can be checked?

pgfplotstodo.tex:2932 **Bug** [open, Priority 5, Epic ‘’]

the quiver/scale arrows thing might need an ”auto” option. If I don’t add it now, it’ll probably never work in the future.

pgfplotstodo.tex:2937 **Bug** [open, Priority 5, Epic ‘’]

‘1.23456e4;’ in a log plot resulted in hard-to-read error messages. Improve sanity checking here.

pgfplotstodo.tex:2956 **Bug** [open, Priority 5, Epic ‘Usability’]

avoid dimension too large errors which occur due to a data range restrictions. Example: data range = 0:6000 view range = 0:1 ⇝ results in error. But that’s easy to detect! Just compute the point coordinate in float (after the scaling is complete). Then, install a filter somewhere. perhaps an ”a posteriori” filter in the pointxyz command?

DUPLICATE: the nasty dimension-to-large message could probably be avoided if pgfplots would simply clip the results to TeX’s range

pgfplotstodo.tex:2970 **Bug** [open, Priority 5, Epic ‘’]

`|yticklabels={<list>},`extra y ticks=...— is incompatible since the extra ticks share the same tick typesetting routine (which, in turn, queries the ¡list¿).

pgfplotstodo.tex:2984 **Bug** [open, Priority 5, Epic ‘’]

The ’text depth’ in legend entries is incompatible with ’text width’. The problem: text width is implemented using `\begin{minipage}[t]` so its contents is all in the depth. Setting text depths overrides the height!

pgfplotstodo.tex:2994 **Bug** [open, Priority 5, Epic ‘’]

one can’t provide ’disable log filter’ to addplot (but it might be interesting)

pgfplotstodo.tex:2998 **Bug** [open, Priority 5, Epic ‘’]

FPU: atan doesn’t check for unbounded inputs.

pgfplotstodo.tex:3003 **Bug** [open, Priority 5, Epic ‘’]

unbounded inputs: improve warning messages: they should not contain low level FPU args.

pgfplotstodo.tex:3009 **Bug** [open, Priority 5, Epic ‘’]

the user interface to set ’tickwidth=0’ for a SINGLE axis is not very good: it seems one needs ’xtick style=/pgfplots/tickwidth=0’ to do so... ⇝ can be solved if tickwidth has a family, I guess. Something like ’draw’ which will not be pulled by pgfplots. But then remains a problem of key paths.

pgfplotstodo.tex:3017 **Bug** [open, Priority 5, Epic ‘’]

I have seen that ’plot table’ with very large files can produce pool size problems – even if the coordinates are all filtered away. In other words: the code can’t simply read a file and throw its contents away. The problem appears to be some math parsing using the table/x expr and friends. ’pool size = names of control sequences and file name’ ⇝ the math parser could be improved with ifcsname

pgfplotstodo.tex:3032 **Bug** [open, Priority 5, Epic ‘’]

providing zmin/xmax to an axis activates 3D mode, ok – but lower dimensional input routines appear to fail.

pgfplotstodo.tex:3036 **Bug** [open, Priority 5, Epic ‘’]

one can’t provide ’scale’ as argument to a (3d) axis

pgfplotstodo.tex:3045 **Bug** [open, Priority 5, Epic ‘’]

it may still happen that log-axes get only *one* tick label (in my case `10^{-0.2}`). That should never happen. The range is about ymin=4.7e-1, ymax=9.5e-1

pgfplotstodo.tex:3049 **Bug** [open, Priority 5, Epic ‘’]

log samples in plot expression for 3D plots

pgfplotstodo.tex:3053 **Bug** [open, Priority 5, Epic ‘’]

different log bases and gnuplot

pgfplotstodo.tex:3067 **Bug** [open, Priority 5, Epic ‘’]

I tried placing a named coordinate inside one axis and using it in another. It failed.

CF: The axis is drawn inside of its own picture which will only be shifted if everything has been drawn. That will be the origin of this problem I guess

Miraculously I can use the coordinate outside axis env. So I have reached the following solution:

pgfplotstodo.tex:3073 **Bug** [open, Priority 5, Epic '']

plot coordinates doesn't check too well if 1. addplot3 is used but only two coords are given 2. addplot is given but three coordinates are provided (also for plot expression)

pgfplotstodo.tex:3086 **Bug** [open, Priority 5, Epic '']

the compat things are not yet complete: I wanted to check when it is really necessary (for example if 'x dir' is used)

pgfplotstodo.tex:3091 **Bug** [open, Priority 5, Epic '']

the nodes near coords feature produces unexpected results when used together with markers ⇝ this is due to the default configuration of scatter plots.

pgfplotstodo.tex:3102 **Bug** [open, Priority 5, Epic '']

the ybar style won't be set inside of `\label{}`

pgfplotstodo.tex:3107 **Bug** [open, Priority 5, Epic '']

axis equal for semilog plots is not correct (?)

pgfplotstodo.tex:3115 **Bug** [open, Priority 5, Epic '']

backwards compatibility problem: axis descriptions can't contain /pgfplots/ styles any longer! This is a key path issue :-(

pgfplotstodo.tex:3120 **Bug** [open, Priority 5, Epic '']

BUG: in empty axes, '`xtick=\empty`' is ignored.

pgfplotstodo.tex:3137 **Bug** [open, Priority 5, Epic '']

The automatic tick labeling sometimes produces inconsistent or confusing labels: 1. engineering and fixed number style are mixed up. 2. If range of an axis is so small that the labels differ only on the third decimal, still only two decimals are used.

pgfplotstodo.tex:3152 **Bug** [open, Priority 5, Epic '']

3D: error bars and stacked plots need to be updated.

Is this still current? There has been a major revision some time ago...

pgfplotstodo.tex:3189 **Bug** [open, Priority 5, Epic ‘’]

```
  3D:  the use of \addplot3 and \addplot is not sanitized properly
Possibilities:
-  used \addplot  when \addplot3 should have been used
-  used \addplot3 where \addplot should have been used.
 What can happen here!? Shouldn't this work in every case?

- The "xtick" value is not applied unless there is a coordinate in the x range:
$\leadsto$ that's the handling of empty figures...
not working:
  \begin{axis}[xtick=0]
  \end{axis}
not working:
  \begin{axis}[xmin=-5,xmax=5,xtick=0]
  \end{axis}
not working:
  \begin{axis}[xmin=-5,xmax=5,xtick=0]
\addplot coordinates { (-10, 0) };
  \end{axis}
working:
  \begin{axis}[xmin=-5,xmax=5,xtick=0]
\addplot coordinates { (0, 0) };
  \end{axis}
```

pgfplotstodo.tex:1519 **Bug** [open, Priority 4, Epic ‘’]

adding a post action with decoration to a pgfplots style results in an error if markers are active

pgfplotstodo.tex:1010 **Bug** [open, Priority 3, Epic ‘’]

contour plot handler does not allow ’name path’.

pgfplotstodo.tex:1471 **Bug** [open, Priority 3, Epic ‘’]

fill between: `soft clip={domain=1:2}}` might fail if the function in question touches the y boundary within the domain. In this case, the soft clip path produces unwanted side–effects.

Cure: enlarge the soft clip path generated by `domain`. It should be bigger than the function in question (with respect to y)

(I have a not-quite-minimal example in my tmp folder) pgfplotstodo.tex:1580

Bug [open, Priority 3, Epic ‘’]

Usability issues around histograms: http://tex.stackexchange.com/questions/151411/histogram-frequention-polygon/151551#151551

[see bugtracker/minimal_4.pdf]

```
\documentclass{standalone}
\usepackage{tikz}
\usetikzlibrary{calc,intersections,through,backgrounds,snakes}
\usepackage{pgfplots}
\pgfplotsset{compat=1.8}
\usepgfplotslibrary{statistics}
\begin{document}
\begin{tikzpicture}
\begin{axis}[
width=1*\textwidth,
height=8cm,
ymin=0,
ymax=7,
title=Histogram lichaamslengte,
ybar interval,
xticklabel={[\pgfmathprintnumber\tick--\pgfmathprintnumber\nexttick [}],
\addplot+[hist={bins=4, data max=12,data min=0}]
table[row sep=\\,y index=0] {
data\\
1\\ 2\\ 1\\ 5\\ 4\\ 10\\ 4\\ 5\\ 7\\ 10\\ 9\\ 8\\ 9\\ 9\\ 11\\
};

\addplot[
       %
       % there seems to be a bug in pgfplots which would shift the two
       % 'hist' plots as if both were bar plots.
       % This is one possible work-around:
       forget plot,
       %
       mark=*,
       hist={bins=4, data max=12,data min=0,
              % this here should *not* become an ybar interval:
              handler/.style={sharp plot},intervals=false,
       },
       %
       % .. but since the other plot is shown with intervals, we should
       % shift this here accordingly:
       shift={(axis direction cs:1.5,0)},
]
table[row sep=\\,y index=0] {
data\\
1\\ 2\\ 1\\ 5\\ 4\\ 10\\ 4\\ 5\\ 7\\ 10\\ 9\\ 8\\ 9\\ 9\\ 11\\
};
\end{axis}
\end{tikzpicture}
\end{document}
```

Problems:

- point meta uses the wrong values (try out nodes near coords)
- ybar interval is active for the second plot
- interval=false should be deduced automagically

pgfplotstodo.tex:1746 **Bug** [open, Priority 3, Epic ‘’]

units and square brackets

```
Dann habe ich noch eine Anmerkung bzgl. der Verwendung von Einheiten in Achsbeschriftungen

------ Zitat
ANMERKUNG;
Die eckigen Klammern drfen nicht um Einheitenzeichen gesetzt werden. Angaben wie [kg] sind
------ Ende Zitat

'DIN 461: Graphische Darstellung in Koordinatensystemen' gibt Beispiele, wie es gemacht we

Hintergrund: formaler Zusammehang ist X={X}[X]. Das heit: X=5N --> {X}=5 und [X]=N. Unsinn

Wie gesagt: leider eine sehr schlechte Angewohnheit und den meisten unbekannt oder schlich
```

While I understand the problem, I am inclined to keep it as-is... I cannot break backwarts compatibility now, and I hear few complaints. I suppose the units stuff could be reviewed at some time

pgfplotstodo.tex:1862 **Bug** [open, Priority 3, Epic ‘’]

gnuplot interface: unbounded coords are not recognized as such (type=u).

pgfplotstodo.tex:1930 **Bug** [open, Priority 3, Epic ‘’]

x tick scale label for style `tiny` has an unsuitable shift

pgfplotstodo.tex:2521 **Bug** [open, Priority 3, Epic ‘’]

`|\addplot`table[blue]— ignores the color options!

pgfplotstodo.tex:3239 **Bug** [open, Priority 3, Epic ‘’]

3D case : tick/grid lines are on top of the axis lines. This leads to poor quality.

... but nobody has ever complaint so far

pgfplotstodo.tex:1402 **Bug** [open, Priority 2, Epic ‘’]

context + color mix produces spurious shifts

```
I'm using pgfplots to create some bar charts with error bars in ConTeXt,
and I want to use custom colours for them.  This works, but if I mix a
custom colour with another colour, it results in a gap between the
```

error bar line and the error bar mark. If two custom colours are mixed, the gap seems to be doubled. Mixing with predefined colours works normally. MWE:

```
\usemodule[tikz]
\usemodule[pgfplots]

\definecolor [Blue] [h=3465a4]
\definecolor [White] [h=ffffff]

\pgfplotsset{compat=newest}

\starttext
\starttikzpicture
\startaxis[ybar, error bars/y dir=both, error bars/y explicit]
\addplot+[blue!50!white] coordinates {(1,5) +- (0,3)};
\addplot+[Blue] coordinates {(1,5) +- (0,3)};
\addplot+[Blue!50!white] coordinates {(1,5) +- (0,3)};
\addplot+[Blue!50!White] coordinates {(1,5) +- (0,3)};
\stopaxis
\stoptikzpicture
\stoptext
```

Here you can see the result:
http://666kb.com/i/cl86t0mfpin35tht7.png

pgfplotstodo.tex:1515 **Bug** [open, Priority 2, Epic ‘’]

Adding user decorations to axis lines might interfer with the discontinuities which are also decorations, compare

http://tex.stackexchange.com/questions/160936/random-decoration-of-plot-axis-messes-up-direction-of-axis-arrow/161025#comment368659_161025

I already added Jake’s workaround to the code, but might still need more cleanup (and for non-centered axis lines as well)

pgfplotstodo.tex:1584 **Bug** [open, Priority 2, Epic ‘’]

Loading a pgfplots library via `\usetikzlibrary{pgfplots...}` fails unless pgfplots is loaded. This should be reported properly (or perhaps load pgfplots implicitly?)

pgfplotstodo.tex:1628 **Bug** [open, Priority 2, Epic ‘’]

some issue regarding extra ticks in log axes has been reported in http://tex.stackexchange.com/questions/148703/bug-pgfplots-extra-ticks

pgfplotstodo.tex:1697 **Bug** [open, Priority 2, Epic ‘’]

The combination of 3d unit vectors and `scale mode=stretch to fill` is useful, but unsupported.

I made a brief experiment to verify that it does something useful if

- the data scaling is disabled in setunitvector
- the first if branch of `...@prepare@plotbox@limits` is activated.

References: `http://tex.stackexchange.com/questions/139686/controlling-orientation-in-3d-pgf-plots`

pgfplotstodo.tex:1777 **Bug** [open, Priority 2, Epic ‘’]

boxplots: predefined legend entries need to be improved

pgfplotstodo.tex:1866 **Bug** [open, Priority 2, Epic ‘’]

gnuplot interface + raw gnuplot does not handle log scale properly (?)

pgfplotstodo.tex:1906 **Bug** [open, Priority 2, Epic ‘’]

new layered graphics stuff: the style changes of the layer config are read too late; it is impossible to overwrite them within the same axis (for example using `set layers,tick style={on layer=...}`)

pgfplotstodo.tex:2245 **Bug** [open, Priority 2, Epic ‘’]

#3213889 hyperref boxes are in wrong position for vertical labels

see `http://tex.stackexchange.com/questions/13364/how-to-make-pgfplots-vertical-labels-have-proper-hyperref-erence-box` for problem description and potential fixes

pgfplotstodo.tex:2340 **Bug** [open, Priority 2, Epic ‘’]

The clipping of tick lines uses the middle of axis lines; it does not incorporate the line width of the axis lines. [see `bugtracker/minimal_5.pdf`]

```
\documentclass{article}
\usepackage{pgfplots}
\pgfplotsset{compat=1.3,
             every axis/.append style={semithick},
             every tick/.append style={semithick,color=black},
             tick align=outside
            }
\begin{document}
```

```
   \thispagestyle{empty}
   \begin{figure}[p]
     \centering
     \begin{tikzpicture}
       \begin{axis}[xmin=0,
                    xmax=30,
                    ymin=0,
                    ymax=1.2
                   ]
       \end{axis}
     \end{tikzpicture}
   \end{figure}
\end{document}
```

pgfplotstodo.tex:2367 **Bug** [open, Priority 2, Epic ‘’]

can someone confirm the following behavior. The y label of a plot gets truncated in some circumstances if the external library is used. This happens for me if no title is specified for a plot. Consider the following example:

[see bugtracker/minimal_6.pdf]

```
\documentclass[11pt,a4paper]{article}

\usepackage{tikz}
\usepackage{pgfplots}

\pgfplotsset{compat=1.3}
\usepgfplotslibrary{external}
\tikzexternalize[force remake]

\begin{document}
\begin{tikzpicture}
\begin{axis}[y tick scale label style={inner sep=1pt}]
\addplot {x * 10^8};
\end{axis}
\end{tikzpicture}
\end{document}
```

pgfplotstodo.tex:2776 **Bug** [open, Priority 2, Epic ‘’]

potential incompatibility: clickable and external. The clickable lib writes into pgfplots.djs which might cause multithreaded problems.

pgfplotstodo.tex:2889 **Bug** [open, Priority 2, Epic ‘’]

OK : ’every node near coord/.append style=scale=0.7’ NOT OK: ’every node near coord/.append style=scale=0.7,ybar’ -⇝ sequence of shift and scale matters ...

pgfplotstodo.tex:2902 **Bug** [open, Priority 2, Epic ‘’]

view=090 and enlargelimits=auto is not always satisfactory: it disables enlarged limits, but for contours, I'd like to have it. What is to do?

pgfplotstodo.tex:2923 **Bug** [open, Priority 2, Epic ‘’]

contour external doesn't handle explicitly provided matrix data (mesh/rows and mesh/cols) yet.

pgfplotstodo.tex:2927 **Bug** [open, Priority 2, Epic ‘’]

contour external doesn't handle the ordering flag correctly.

pgfplotstodo.tex:2942 **Bug** [open, Priority 2, Epic ‘’]

the title style for 'footnotesize' is not as I want it to: it doesn't respect the depth below the baseline. Or does it need a `\strut`?

pgfplotstodo.tex:3162 **Bug** [open, Priority 2, Epic ‘’]

interp shader is displayed transparently in evince

Bug in evince (for sure). What is to be done?

pgfplotstodo.tex:1014 **Bug** [open, Priority 1, Epic ‘’]

contour plot handler does not allow to specify draw and fill – which might be useful if just one contour is drawn/filled

pgfplotstodo.tex:1594 **Bug** [open, Priority 1, Epic ‘’]

The special feature `patch,patch type=quadratic spline, point meta=none` which produces global paths appears to duplicate the low level path. I suppose this can only be seen in the resulting pdf file or in a debugger, but it makes a big difference when used together with decorations or fill between.

This feature should be documented in more places as well.

FOUND IT: the problem is stated incorrectly: the path is generated once (and only once). BUT: `\tikz@mode` is invoked twice, once by the mesh plot handler and once by tikz. This is idempotent, in general – but not if we have `name path=A` in place! In this case, the softpath will be *appended* every time `\tikz@mode` is being executed.

INACTIVE: at the time of this writing, the offending “append” feature of `name path` is inactive; this bug does not happen anymore

pgfplotstodo.tex:1619 **Bug** [open, Priority 1, Epic ‘’]

|\usepackage[gray]{xcolor}|does not seem to work as good as for cmyk... in particular, the default color map is not being converted correctly!? Probably because xcolor does not convert the colors if they are RGB

This appears to be irrelevant because RGB gray colors can be translated losslessy to scalar gray values!?

pgfplotstodo.tex:1940 **Bug** [open, Priority 1, Epic ‘’]

Internal coordmath framework: it is not used everywhere

pgfplotstodo.tex:1964 **Bug** [open, Priority 1, Epic ‘’]

|lognumber format code— is a global variable and cannot be set for individual axes.

This applies to `log ticks with fixed point` as well.

pgfplotstodo.tex:2266 **Bug** [open, Priority 1, Epic ‘’]

CRASH: [see `bugtracker/minimal_7.pdf`]

```
\begin{tikzpicture}
	\begin{axis}[
		scale mode=scale uniformly,
		x={(1pt,0pt)},
		y={(-0.5pt,0.5pt)},
		z={(0pt,1pt)},
	]

	% addplot3 works (with 3d coords):
	\addplot coordinates {
		(0,0) (1,0) (0,1)
	};

	\end{axis}
\end{tikzpicture}
```

pgfplotstodo.tex:2311 **Bug** [open, Priority 1, Epic ‘’]

Using 0 in pgfplots coordinate systems does not necessarily mean “no offset”. This is misleading. Bug sourceforge #3168030: [see `bugtracker/minimal_8.pdf`]

```
\documentclass[a4paper]{article}
\usepackage{german}
\usepackage[utf8]{inputenc}

\usepackage{pgfplots}

\usepackage{pgfplotstable}
\usepackage{booktabs}
```

```
\usepackage{array}
\usepackage{colortbl}

\begin{document}

\begin{tikzpicture}
       \begin{axis}[enlarge x limits=false, extra description/.code={\draw[very thick] (axis cs:2.5,0) -- ++(re
              \addplot coordinates{
                                                  (0,1)
                                                  (1,2)
                                                  (2,3)
                                                  (3,4)
                                                  (4,5)};
       \end{axis}
\end{tikzpicture}

\begin{tikzpicture}
       \begin{axis}[enlarge x limits=true, extra description/.code={\draw[very thick] (axis cs:2.5,0) -- ++(rel
              \addplot coordinates{
                                                  (0,1)
                                                  (1,2)
                                                  (2,3)
                                                  (3,4)
                                                  (4,5)};
       \end{axis}
\end{tikzpicture}
\end{document}


\end{document}
```

pgfplotstodo.tex:2439 **Bug** [open, Priority 1, Epic ‘’]

The legend has the `text depth=0.15em` initial configuration, which is extremely bad for legend entries with huge depth (large fractionals or formulas?)

pgfplotstodo.tex:2559 **Bug** [open, Priority 1, Epic ‘’]

The `mark list` produces a lot of

```
\XC@edef #1#2->\begingroup \ifnum \catcode ‘\!=13 \edef !{\string !}\fi \ifnum \catcode ‘\
[........]
{\if}

\@@tmp ->.!80!black
{true}
{the character !}
Missing character: There is no ! in font nullfont!
{the character 8}
Missing character: There is no 8 in font nullfont!
{the character 0}
Missing character: There is no 0 in font nullfont!
{the character !}
```

```
Missing character: There is no ! in font nullfont!
{the character b}
Missing character: There is no b in font nullfont!
{the character l}
Missing character: There is no l in font nullfont!
{the character a}
Missing character: There is no a in font nullfont!
{the character c}
Missing character: There is no c in font nullfont!
{the character k}
Missing character: There is no k in font nullfont!
{\def}
{\else}
```

bugs. Probably fixed with more recent version of xcolor?

pgfplotstodo.tex:2626 **Bug** [open, Priority 1, Epic ‘’]

providing `\legend{}` without any `\addplot` commands causes a problem

pgfplotstodo.tex:2715 **Bug** [open, Priority 1, Epic ‘’]

the axis line combination styles can’t be adjusted for 3D because they are evaluated too early.

pgfplotstodo.tex:2912 **Bug** [open, Priority 1, Epic ‘’]

there are a lot of .code 2 args styles which do not support spaces between their arguments. Fix this.

pgfplotstodo.tex:2989 **Bug** [open, Priority 1, Epic ‘’]

the ’/pgfplots/table/.search also’ is overwritten during `\addplot table` with /.search also=/pgfplots. That’s not so good.

pgfplotstodo.tex:3227 **Bug** [open, Priority 1, Epic ‘’]

In 3D case axis [xyz] line != box, there is just ONE hyperplane. My implementation works only if either ALL are box or ALL are ’middle’.

pgfplotstodo.tex:3262 **Bug** [open, Priority 1, Epic ‘’]

javascript stuff does not work if the complete figure is rotated (sidewaysfigure).

pgfplotstodo.tex:3277 **Bug** [open, Priority 1, Epic ‘’]

javascript: incompatiblity with external library: 1. filenames: `\jobname` contains characters with incompatible catcodes and that funny insdljs package tries to assemble macros with these characters. ⇝ fixed; I simply use pgfplotsJS as temporary file name. 2. the images as such have corrupted forms ⇝ Can be fixed if `\usepackage{eforms}` is used BEFORE loading pgf. The reason: `\begin{Form}` and the shipout-hackery of the pgf externalization bite each other. `\begin{Form}` must come before the shipout hackery of pgf. 3. `\includegraphics` does not preserve PDF forms.

pgfplotstodo.tex:3284 **Bug** [open, Priority 1, Epic ‘’]

the interrupt bounding box feature should still update the data bounding box. Otherwise, transformations may fail.

6 Feature Proposals PGFPlots

pgfplotstodo.tex:3345 **Bug** [open, Priority 5, Epic ‘’]

Unbounded point meta data should be filtered out.

`http://tex.stackexchange.com/questions/164250/pgfplots-surf-plot-dont-draw-nan?lq=1`

However, it seems as if this “filter out” should not be done as for coordinates (which would filter out all adjacent patch segments). It should merely filter out the current one.

pgfplotstodo.tex:5214 **Feature Proposal** [open, Priority 6, Epic ‘’]

It is certainly possible to write some sort of CELL-BASED ’mesh/surf’ shader - a combination of ’flat corner’ and cell based rectangles:

- every coordinate denotes a CELL instead of a corner,
- the ”shader” maps the cdata into the colormap to determine the cell color
- details?
  - to get well-defined cells, I have to enforce either a non-parametric lattice grid or do a LOT of additional operations (?).
  - alternative: define N*M cells by N+1 * M+1 points.
  - perhaps a combination of both? ⇝ that’s more or less the same as ’flat mean’ up to the further row/column pair

- it would be generally useful to have an "interval" or "cell" mode: the idea is that every input coordinate defines an interval (1d) or a cell (2d). To define the last cell, one needs to add one "mesh width" somehow. I just don't know where:
  - the artificial cell should be processed with the normal streams - including limit updates, stacking etc.
  - the artificial cell needs to know when the end-of-stream occurs. For 1d plots, that may be possible. For 2D plots, this information requires a valid 'cols' key.
  - I suppose it would be best to patch @stream@coord.. at least for the 'cell' mode.
  - Idea:
    - the `\pgfplots@coord@stream@coord` implementation realizes the cell-mode: after every 'cols' coordinate, a further one is replicated. This needs the "last mesh width". Furthermore, it needs to accumulate a row vector, the "last row". This last row is need during stream@end to replicate the further row:
    - the `\pgfplots@coord@stream@end` implementation has to realize the last step of cell mode: the replication of a further row. It also has to realize the implementation of 'interval' mode (replication of last coordinate). My idea is to simply use an applist for this row accumulation. The format should be compatible with `\pgfplots@coord@stream@foreach@NORMALIZED`. That doesn't produce problems, even when the end command is invoked within a foreach@NORMALIZED loop - because the loop has already ended.

pgfplotstodo.tex:3302 **Feature Proposal** [open, Priority 5, Epic 'Usability']

parse elements of tick positions with math parser

`http://tex.stackexchange.com/questions/187115/with-pgfplots-how-to-manually-enter-ticks-as-fractions`

pgfplotstodo.tex:3308 **Feature Proposal** [open, Priority 5, Epic '']

table package: Add style "modify content for the following row indices"

can be copy-pasted from `http://tex.stackexchange.com/questions/187067/how-to-have-different-colormaps-for-different-columns-in-the-same-heatmaps-table/187099#187099`

pgfplotstodo.tex:3312 **Feature Proposal** [open, Priority 5, Epic ‘’]

fillbetween: accept negative indices for segments in `intersection segments=A{-1}`

pgfplotstodo.tex:3316 **Feature Proposal** [open, Priority 5, Epic ‘’]

fillbetween: accept `soft path={inner x range}` or something like that, compare `http://tex.stackexchange.com/questions/180127/stacking-plots-in-animation-using-fill-between-library-with-dynamic-calculation/180299?noredirect=1#comment416484_180299`

pgfplotstodo.tex:3332 **Feature Proposal** [open, Priority 5, Epic ‘’]

ich bin es nochmal. Ich habe nochmal eine Anmerkung zu den decorations. Ich bin gerade dabei, meine Plots mit Pfeilen zu versehen wie ich es in dem Bild zuvor bereits getan habe. Leider kann ich fr die decorations nur folgende Angabe machen:

```
mark = at position 0.15 with {\arrow [scale=1]{stealth}},
```

Jedoch kommt es bei Kurven hufiger vor, dass ich nicht gut abschtzen kann, ob es sich dabei um Position 0.1 oder 0.15 oder dergleichen handelt. Ich habe es so verstanden, dass der Compiler die Strecke der Kurve vermisst und bei bspw.

```
mark = at position 0.5 with {\arrow [scale=1]{stealth}},
```

eine Dekoration bei der Hlfte der Kurve macht. Doch wo ist die Hlfte der Kurve, wenn diese gekrmmt und gewunden ist? Nun ist es umstndlich immer diese Dekorationen anzupassen und ich wrde lieber eine x oder y Koordinate verwenden und sagen knnen:

```
mark = at x position 40 with {\arrow [scale=1]{stealth}},
```

So kann ich mir sicher sein, dass der Pfeil dann der x-Koordinate 40 zugeordnet wird und auf der Kurve landet.

pgfplotstodo.tex:3336 **Feature Proposal** [open, Priority 5, Epic ‘’]

Implement something like `pos={x=40}`

pgfplotstodo.tex:3352 **Feature Proposal** [open, Priority 5, Epic ‘’]

addplot graphics: support “scale to natural dimensions” of the input graphics

there is a half-ready solution in `http://tex.stackexchange.com/questions/175618/`

addplot-graphics-maintaining-images-aspect-ratio-despite-different-scaling-of/175726#175726

pgfplotstodo.tex:3419 **Feature Proposal** [open, Priority 5, Epic 'Usability']

The trigonometric functions rely on degrees which is one of the most unanticipated properties of pgfplots.

It would be good to change them to radians while keeping compatibility.

Thoughts:

- ensure that only "plot–related" items are replaced (not tikz basic stuff)
  - coord math stuff
  - calls to pgfmathparse while working with coordinate math (which should be coord math but isn't)

  this is hard to collect! Open questions
  - what if someone writes `\addplot table {x.dat} (0:90) node {x};` !?
  - axis cs should make use of the key (polar axis only probably)
  - survey phase should have it (for all calculations probably)
  - what about view key!? Probably not unless reconfigured globally
  - adopt key filtering: should be possible in `\begin{axis}[trig format=rad]` (it is not currently)
  - what about `data cs=polarrad`? would apply rad twice
  - what about `data cs=polar`? is defined to use degrees!?
  - an idea would be to apply it to plot expression – and only to plot expression
- modify all related functions `/pgf/trig format=deg,rad`:

Function	basic	FPU
sin	✓	uses basic ✓
cos	✓	uses basic ✓
tan	uses sin/cos ✓	uses basic ✓
sec	uses cos ✓	uses basic ✓
cosec	uses sin ✓	uses basic ✓
cot	uses sin/cos ✓	uses basic ✓
asin	✓	uses basic✓
acos	✓	uses basic ✓
atan	✓	(special impl) ✓
atan2	✓	unavailable

  ATTENTION: `pgfmathfloatTRIG@` needs to be patched (compute modulo 2π instead of 360)! ✓

  ATTENTION: the switch replaces all math functions. This includes all path instructions and libraries and and and... set it only in local scopes!

- think about upgrade procedure
  - add switch of sorts `trig coordinates=rad|deg`
  - force users to add this switch manually into the preamble and advertise in every example in manual
  - after two releases: add it to `compat=<version>`
  - alternatively: add directly to `compat=<version>`
- ensure that pgfplots 1.11 comes with a copy of the affected math functions

pgfplotstodo.tex:3425 **Feature Proposal** [open, Priority 5, Epic 'Usability']

The distinction into survey phase and visualization phase is technically reasonable, but causes confusion.

Is there a way to simplify loop operations during the survey phase as in `http://tex.stackexchange.com/questions/172917/pgfplots-and-using-axis-cs-to-add-points` ?

pgfplotstodo.tex:3435 **Feature Proposal** [open, Priority 5, Epic 'Usability']

Pgfplots requires user input to distinghuish between parameterized 3d plots and matrix-like 3d plots.

Why can't PGFPlots determine automatically if `z buffer=sort` is adequate?

`http://tex.stackexchange.com/questions/172753/pgfplots-and-gnuplot-with-addplot3`

ATTENTION: what about 3d sampled line plots!? this would break with z buffer sort.

pgfplotstodo.tex:3465 **Feature Proposal** [open, Priority 5, Epic 'Usability']

pgfplots math expressions always yield internal FPU formats which confuses users.

Return SCI representation instead.

TO BE CLARIFIED: which operations should be replaced? REplacing 'x filter' and its friends might actually make things worse because FPU functions will no longer accept the argument.

It might be useful to improve the FPU such that declarefunction for a function which does *not* expect FPU arguments accept it.

This could be done by patching `\pgfmath@stack@push@operation` - and check if the argument is a function and that function has a known FPU implementation (its float-backup exists). If not: generate a dummy which converts to fixed points. Use the function's arity!

see
`http://tex.stackexchange.com/questions/171263/plotting-a-function-defined-with-pgfmathdecl`

see
`http://texwelt.de/wissen/fragen/3960/fraktale-mit-pgfplots?Seite=1#3993`

I started to work on a solution for the second one, see the uncommented call to `\pgfplots@expression@normalize@floats`

pgfplotstodo.tex:3487 **Feature Proposal** [open, Priority 5, Epic ‘’]

bullet graphs:

`http://tex.stackexchange.com/questions/146538/how-to-create-vertical-bullet-graphs-with-pstricks`

Inspiration:

`http://img35.imageshack.us/img35/1061/snap2134.png`

I started a prototype. To do items:

- make sure that color schemes can be exchanged in a simple way
- prepare the package for up to 3 discriminative markers
- regarding data files: TO CLARIFY

pgfplotstodo.tex:3493 **Feature Proposal** [open, Priority 5, Epic ‘’]

Given some PGF point, allow to access the high-level coordinates.

see `http://tex.stackexchange.com/questions/174404/converting-unit-to-coordinate-works-for-x-coordinate-but-not-for-y-coordnate/174443#174443` for a use-case and note that there is a work-around which works for ‘pin‘ in the pgfplots manual - and that is awkward.

pgfplotstodo.tex:3499 **Feature Proposal** [open, Priority 5, Epic ‘’]

add “stackable tick labels” (tick labels with vertical shifts if they are too close).

compare `http://tex.stackexchange.com/questions/164673/positioning-even-or-odd-x-axis-tick-labels-in-pgfplots`

pgfplotstodo.tex:3503 **Feature Proposal** [open, Priority 5, Epic ‘’]

bar plots: Introduce significance stars (see GROUP BARS on page 69)

pgfplotstodo.tex:3777 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

GROUP BARS

Bar plots: simplify grouped bars

```
Hi Joshua,

as already mentioned, your old mails from January have slipped my discipline... I am sorry

You asked for a simpler way to
a) position groups of bars without having to compute 'bar shift' manually
b) a simpler way to get "significance stars".

First, grouped bar charts are an open feature request in pgfplots, and your input will eve

The first request is actually possible, assuming that you always provide them in the seque

The default is
    /pgfplots/ybar/.style={
        /pgf/bar shift={%
                % total width = n*w + (n-1)*skip
                % -> subtract half for centering
                -0.5*(\numplotsofactualtype*\pgfplotbarwidth + (\numplotsofactualtype-1)*#
                % the '0.5*w' is for centering
                (.5+\plotnumofactualtype)*\pgfplotbarwidth + \plotnumofactualtype*#1},%
    }

and my modification substitutes every index and 'n' by half of it:

\pgfplotsset{
    % #1 = separation between bars
    bar shift for half number plots/.style={%
        /pgf/bar shift={%
            % total width = n*w + (n-1)*skip
                % -> subtract half for centering
                -0.5*(\numplotsofactualtype/2*\pgfplotbarwidth + (\numplotsofactualtype/2-
                % the '0.5*w' is for centering
                (.5+div(\plotnumofactualtype,2))*\pgfplotbarwidth + div(\plotnumofactualty
        },%
    },%
    bar shift for half number plots/.default=2pt,
}

With this definition, you do not have to write 'bar shift' in your styles.
```

The whole figure becomes

```
\usemodule[pgfplots]
\pgfplotsset{compat=newest}

\pgfplotsset{HeartControl/.style=
{
    red, fill=red!33!white,
    %bar shift=-0.1667
}}
\pgfplotsset{HeartDiabetes/.style=
{
    red!33!black, fill=red!66!white,
    %bar shift=+0.1667
}}
\pgfplotsset{KidneyControl/.style=
{
    orange, fill=orange!33!white,
    %bar shift=-0.1667
}}
\pgfplotsset{KidneyDiabetes/.style=
{
    orange!33!black, fill=orange!66!white,
    %bar shift=+0.1667
}}

\pgfplotsset{/pgfplots/ybar legend/.style=
{
    /pgfplots/legend image code/.code={%
        \draw[
            ##1, /tikz/.cd,
            bar width=0.25em,
            yshift=-0.27em,
            bar shift=0pt
        ]
        plot coordinates {(0pt,0.8em)};
    }
}}

\pgfplotsset{
    % #1 = separation between bars
    bar shift for half number plots/.style={%
        /pgf/bar shift={%
            % total width = n*w + (n-1)*skip
                % -> subtract half for centering
                -0.5*(\numplotsofactualtype/2*\pgfplotbarwidth + (\numplotsofactualtype/2-
                % the ’0.5*w’ is for centering
                (.5+div(\plotnumofactualtype,2))*\pgfplotbarwidth + div(\plotnumofactualty
        },%
    },%
    bar shift for half number plots/.default=2pt,
}

\starttext

\starttikzpicture
    \startaxis
    [
        bar width=0.3,
        ybar,
        bar shift for half number plots=5pt,
        xtick=data,
        ylabel={mRNA level},
        ymin=0,
        xmin=0.333, xmax=2.667,
        xtick={1,2}, xticklabels={Heart,Kidney},
        error bars/y dir=both,
        error bars/y explicit,
        legend columns=2,
        legend pos=outer north east,
        legend cell align=left
    ]

    \addplot+[HeartControl] coordinates {(1,1) +- (0,0.1)};
    \addplot+[KidneyControl] coordinates {(2,1) +- (0,0.12)};
    \addplot+[HeartDiabetes] coordinates {(1,1.1) +- (0,0.2)};
    \addplot+[KidneyDiabetes] coordinates {(2,0.8) +- (0,0.05)};

    \draw (axis cs:2,1.15) +(-2.2em,0) -- +(2.2em,0);
    \node[anchor=south, yshift=-1ex] at (axis cs:2,1.15) {*};

    \legend{{\kern-0.1em}, Control, {\kern-0.1em}, Diabetes}

    \stopaxis
\stoptikzpicture

\stoptext
```

I suppose you could even simplify the styles by means of a cycle list or whatever.

I also experimented with symbolic x coords, but how would you write "xmin=0.3333" or "bar

You can use

```
\def\heartUnit{1}
\def\kidneyUnit{2}
```

to introduce constants - this might make it more readable.

I am aware of the fact that this solution comes way too late. Perhaps it proves to be usef

Kind regards

Christian

```
Am 21.01.2014 13:29, schrieb Joshua Krmer:
> Dear pgfplots developers!
>
> First, thanks a lot for your great package.  I'm using it to create
> diagrams in ConTeXt.  Please consider the following (M)WE.  I hope you
> can run it, otherwise, you can see the output here:
> http://666kb.com/i/cl5sdm34i4ig69mkk.png
>
> \usemodule[pgfplots]
> \pgfplotsset{compat=newest}
>
> \pgfplotsset{HeartControl/.style=
> {
>  red, fill=red!33!white,
>  bar shift=-0.1667
> }}
> \pgfplotsset{HeartDiabetes/.style=
> {
>  red!33!black, fill=red!66!white,
>  bar shift=+0.1667
> }}
> \pgfplotsset{KidneyControl/.style=
> {
>  orange, fill=orange!33!white,
>  bar shift=-0.1667
> }}
> \pgfplotsset{KidneyDiabetes/.style=
> {
>  orange!33!black, fill=orange!66!white,
>  bar shift=+0.1667
> }}
>
> \pgfplotsset{/pgfplots/ybar legend/.style=
> {
>  /pgfplots/legend image code/.code={%
>  \draw[
>  ##1, /tikz/.cd,
>  bar width=0.25em,
>  yshift=-0.27em,
>  bar shift=0pt
>  ]
>  plot coordinates {(0pt,0.8em)};
>  }
> }}
>
> \starttext
>
> \starttikzpicture
>  \startaxis
>  [
>  ybar,
>  xtick=data,
>  ylabel={mRNA level},
>  ymin=0,
>  xmin=0.333, xmax=2.667,
>  xtick={1,2}, xticklabels={Heart,Kidney},
>  error bars/y dir=both,
>  error bars/y explicit,
>  bar width=0.3,
>  legend columns=2,
>  legend pos=outer north east,
>  legend cell align=left
>  ]
>
>  \addplot+[HeartControl] coordinates {(1,1) +- (0,0.1)};
>  \addplot+[KidneyControl] coordinates {(2,1) +- (0,0.12)};
>  \addplot+[HeartDiabetes] coordinates {(1,1.1) +- (0,0.2)};
>  \addplot+[KidneyDiabetes] coordinates {(2,0.8) +- (0,0.05)};
>
>  \draw (axis cs:2,1.15) +(-2.2em,0) -- +(2.2em,0);
>  \node[anchor=south, yshift=-1ex] at (axis cs:2,1.15) {*};
>
>  \legend{{\kern-0.1em}, Control, {\kern-0.1em}, Diabetes}
>
>  \stopaxis
> \stoptikzpicture
>
> \stoptext
>
> As you can see, I have two groups (organs: hearts, kidneys), sometimes
> more, and two conditions (control, Diabetes).  To make it easier to
> compare the diagrams (there are many), I want to use consistent colours
> for the same organs, and two brightnesses for the two conditions.  The
> code above works, but automatic positioning would be much nicer, so I
> could just set something like "bar width=..." and "bar seperation=..."
> and let the bars be positioned automatically.  This would also allow me
> to use symbolic coordinates, avoiding the pseudo coordinates (1 and
> 2).  Is there a better way to get the desired result than what I've
> done?
>
> I also hope there is a better way to create significance stars.  If two
> values are significantly different, there is a horizontal line to be
> added which spans the two bars, and a symbol above it.  The symbols
> usually are one till three stars (depending on the degree of
> significance), sometimes other symbols are used in the literature.  At
> least it would be nice if I could define the coordinates for the
> horizontal line with something like "max(errormark1, errormark2) +a",
> ie, the higher of the two error marks involved plus some separation.
>
> Kind regards,
> Joshua Krmer
>
>
>
> ------------------------------------------------------------------------------
> CenturyLink Cloud: The Leader in Enterprise Cloud Services.
> Learn Why More Businesses Are Choosing CenturyLink Cloud For
> Critical Workloads, Development Environments & Everything In Between.
> Get a Quote or Start a Free Trial Today.
> http://pubads.g.doubleclick.net/gampad/clk?id=119420431&iu=/4140/ostg.clktrk
> -----------------------------------------------
> Pgfplots-features mailing list
> Pgfplots-features@lists.sourceforge.net
> https://lists.sourceforge.net/lists/listinfo/pgfplots-features
```

pgfplotstodo.tex:3811 **Feature Proposal** [open, Priority 5, Epic ‘’]

suppress tick-line cross in 3d for view directions in which the cross degenerates to a very thick line

compare

```
\documentclass[tikz,12pt]{standalone}
\usepackage{tikz,pgfplots,pgfplotstable}

\pgfplotsset{compat=1.10}

\pgfplotsset{every axis/.append style={tick style={line width=0.7pt}}}

\pgfplotstableread{
a b
-0.1 0.2
0.1 0.5
}\testdata

\begin{document}
```

```
\foreach \h in {5,10,...,360} {
\begin{tikzpicture}

\begin{axis}[title=\h,axis lines=center, ymin=-0.22,ymax=0.22, xmin=-0.2,xmax=0.2,xlabel=x
    \addplot3 [color=blue,] table[x expr=0,y=a,z=b] {\testdata};
\end{axis}

\end{tikzpicture}
\par
}

\end{document}
```

pgfplotstodo.tex:3828 **Feature Proposal** [open, Priority 5, Epic ‘’]

Tufte-style range frames for 3d `http://tex.stackexchange.com/questions/165766/tufte-style-range-frames-for-three-dimensional-plots`

pgfplotstodo.tex:3834 **Feature Proposal** [open, Priority 5, Epic ‘’]

Add option of sorts “shift along outer normal vector of current axis” while generating paths for axes, ticks, tick labels, and perhaps even grid lines.

`http://tex.stackexchange.com/questions/165766/tufte-style-range-frames-for-three-dimensional-plots`

pgfplotstodo.tex:3844 **Feature Proposal** [open, Priority 5, Epic ‘’]

Allow to shift (x) tick labels vertically if they are too dense.

Sometimes one needs lots of tick labels. It would be cool if every second would be shifted vertically, perhaps with an edge to the original location

pgfplotstodo.tex:3882 **Feature Proposal** [open, Priority 5, Epic ‘’]

the `...` syntax in tick label position lists is inherently limited to $[-16384, 16384]$ (it inherits the limitations of `\foreach`). Think about alternatives

pgfplotstodo.tex:3918 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

It is surprisingly difficult to have JUST axis ticks and tick labels and labels, but NO axis line. This is because I accidentally made `axis x line=none` equivalent to `hide x axis`. Too bad.

Idea: implement keys `axis x line hidden=true,false`. Perhaps with options `axis x line=bottom hidden` which is the same as `axis x line hidden,axis x line=bottom`?

pgfplotstodo.tex:3939 **Feature Proposal** [open, Priority 5, Epic '']

Implement document-level javascript for the clickable lib WITHOUT the eforms/insdljs package

should be quite straight-forward. Unless resource-acquisition problems occur (i.e. interoperability issues with other packages)

See

`http://tex.stackexchange.com/questions/3080/what-is-the-best-way-to-insert-document-level-javascript-in-latex-documents?rq=1`

pgfplotstodo.tex:3945 **Feature Proposal** [open, Priority 5, Epic '']

Implement a custom legend environment such that one doesn't need to collect all options manually

`http://tex.stackexchange.com/questions/54794/using-a-pgfplots-style-legend-in-a-plain-old-`

pgfplotstodo.tex:3950 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

Allow 3d bar plots (see also `http://tex.stackexchange.com/questions/176347/2d-bar-chart-in-3d-space/176363#176363`)

pgfplotstodo.tex:3962 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

Individual bars: allow to modify / adjust the bar plot handler(s) such that each bar can have its individual appearance

- create individual `\path` instructions for every bar
- discard the outer `\path` at the end
- allow simple styles of sorts `bar 1/.style={...}` or `bar value 1.23/.style={...}` perhaps using prefix search? similar to the request for nodes near coords
- should be dependend on point meta (like scatter plots)

There is some preparation key `at begin bar` combined with `at end bar` in the bar plot handlers. It can be used as low–level backend, but it still needs to be worked out (see `unittest_bar_shade_atbeginbar.tex`) pgfplotstodo.tex:3966 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

nodes near coords: allow styles of sorts `node near coord 1/.style={...}` or `node near coord value 1.23/.style={...}` (similar to the feature request for bar plots)

pgfplotstodo.tex:3972 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

bar plots: auto-select axis limits, unit size, bar width, and bar shift.

Perhaps it is sufficient to auto-select bar width.

pgfplotstodo.tex:3976 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

Is it possible to have bar plots which do not start from the x or y axis?. For example a bar plot from (0,2) to (0,3).

pgfplotstodo.tex:3980 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

bar plots: provide constant zero level?

pgfplotstodo.tex:3988 **Feature Proposal** [open, Priority 5, Epic '']

allow to rotate polar plots and fix rotation of tick labels.

see

http://tex.stackexchange.com/questions/116830/polar-plot-x-and-y-ticks-and-units

pgfplotstodo.tex:3993 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

polar axes: polar bar plots (see sourceforge feature request and `http://matplotlib.sourceforge.net/examples/pylab_examples/polar_bar.html`)

pgfplotstodo.tex:4003 **Feature Proposal** [open, Priority 5, Epic 'Bar plots']

discontinuity in the middle of a plot (as an example see the phase diagram of water `http://pruffle.mit.edu/3.00/Lecture_29_web/img20.gif`)

`http://peltiertech.com/images/2011-11/Ybroken.png`

`http://tex.stackexchange.com/questions/46422/axis-break-in-pgfplots`

pgfplotstodo.tex:4007 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

chunked bars: interrupt the bars at predefined coordinates (like white grid lines)

pgfplotstodo.tex:4018 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

bar plots:

- bar interval plot handler which *assumes* uniform distances. This allows to eliminate the last, superfluos grid point (because it can be generated automatically as replication xlast + h for known h)
- in fact, I could also implement xlast + hlast and introduce a new name like ’bar interval*’ or something like that

pgfplotstodo.tex:4030 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

Mails from Stefan Ruhstorfer:

- Gruppierte Säulendiagramme sind nach meinem Wissenstand nur dann möglich wenn man in der Axis-Definiton die Bedindung ybar angibt. Ich finde diese Ausrichtung sehr unflexible, da ich sehr oft über das Problem stolpere, dass ich in meinem gruppierten Säulendiagramm noch eine waagrechte Linie oder ähnlichs einzeichnen möchte um z.B. meine obere Toleranzgrenze einzuzeichnen. Bis jetzt mache ich das über den normalen draw Modus, was auch ausgezeichnet funktioniert. Jedoch habe ich dann das Problem, dass ich keinen schönen Legendeintrag mehr bekomme. Hier häte ich 2 Vorschläge. Zum einen die Legende ”freier” zu gestalten. Also so, dass man beliebig (ggf. auch ohne Plot) ein Legendenelement hinzufüen kann und vllt. noch das zugehörige Symbol festlegen kann. (Bis jetzt habe ich das Problem, das ich mit tricksen zwar meine Obere Tolerangrenze in die Legende bekomme, dann jedoch mit einem Säulenzeichnen davor). Der andere Vorschlag ist, dass Säulendiagramm anders zu definiern. So das ich auch noch einen Plot hinzufügen kann, der mir eine waagrechte Linie ohne zu tricksen einzeichnen lässt.
- Eine Gruppierung von stacked bars ist nach meinem Wissen nicht möglich. Es ist zwar schwer sich ein Anwendungsgebiet dafür vorzustellen, aber wenn sie danach mal suchen (speziell im Excelbereich) werden sie sehen, dass viele Leute so eine Funktion benutzen. ⇝ siehe auch folgemails mit Beispielskizzen ⇝ beachte: Fall 2.) erfordert mehr arbeit als lediglich ’line legend’, weil ybar ja den koordinatenindex verarbeitet!

pgfplotstodo.tex:4034 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

let stacked bar plots respect the line width when stacking them on top of each other

pgfplotstodo.tex:4053 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

On input: provide the desired distance between adjacent bars. The bar width should be automatically determined in a way which shows all bars.

An extreme case would degenerate to ycomb.

If the plot area does is insufficient, bar distances should be restricted (?)

Note: if (and only if) the bars have unit distance 1, a relative distance is already supported by providing bar width=0.9 or something like that...

Note: if we knew the distance between adjacent ticks, and we assume that each tick has its associated bar plot, we could define the bar width in terms of the tick distance... perhaps I should expose read-only access to the list of (computed) tick positions; with a convenience method to access the common distance? Or just a common distance?

pgfplotstodo.tex:4059 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

bar plots: if you provide both, the desired distance between adjacent bars and the bar width, the axis limits should be increased or decreased automatically.

This is actually a more generic concept: it would be ’scale mode=limits only’ (?)

pgfplotstodo.tex:4063 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

Default bar plot styles should always include the (correct) origin in the visible axis range.

pgfplotstodo.tex:4069 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

It should be simpler to customize the position of nodes near coords in a way to position them in the middle or below each bar.

This is particularly difficult for stacked bar plots

pgfplotstodo.tex:4087 **Feature Proposal** [open, Priority 5, Epic ‘’]

modify the `node[at=<pos>]` feature such that `node[at value=42.4]` or `at max value` or `at min value` or `node[at coordindex=3]` pgfplotstodo.tex:4094 **Feature Proposal** [open, Priority 5, Epic '']

Layered graphics: consider drawing tick lines which are on the "outer part" of the axis on the foreground layer.

See
`http://tex.stackexchange.com/questions/31708/draw-a-bivariate-normal-distribution-in-tikz/`
for a motivation (the tick lines are hidden by the surface)

pgfplotstodo.tex:4100 **Feature Proposal** [open, Priority 5, Epic '']

filled contour plots (prototype is 10% ready)

works by means of gnuplot if the outer region is extended artificially. However, color data needs to be compensated etc.

pgfplotstodo.tex:4104 **Feature Proposal** [open, Priority 5, Epic '']

allow support for units in `bar width` and `bar shift` (compare the implementation for circles/ellipses)

pgfplotstodo.tex:4121 **Feature Proposal** [open, Priority 5, Epic '']

it would be nice to have automatic PNG export for huge graphics. Such an approach, combined with plot graphics, could result in considerably smaller pdfs and faster rendering. At the same time, it would not suffer the limitation which arises if one uses the external lib and converts the complete figure to png (including axis descritpions)

pgfplotstodo.tex:4131 **Feature Proposal** [open, Priority 5, Epic '']

There is no simple way to provide LOG colorbars:

1. ymode=log is not supported in 'every colorbar' due to key filtering problems
2. disablelogfilter appears to be useless and does not respect 'log basis'

If those two this would be fixed, one could provide
`colorbar style={ymode=log,disablelogfilter}` and would get a proper logarithmic colorbar. Perhaps even combined with `log basis` ... ?

pgfplotstodo.tex:4135 **Feature Proposal** [open, Priority 5, Epic '']

Cases-statement in math parser

`pgfplotstodo.tex:4143` **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

provide a way to provide more customization to stacked plots as in

`http://tex.stackexchange.com/questions/13627/pgfplots-multiple-shifted-stacked-plots-in-on`

(stacked and clustered bar charts)

`pgfplotstodo.tex:4147` **Feature Proposal** [open, Priority 5, Epic ‘’]

the `empty line` feature should produce a log notice when it finds an empty line in compat mode.

`pgfplotstodo.tex:4155` **Feature Proposal** [open, Priority 5, Epic ‘’]

Support something like
’`\addplot table[x symbolic expr={\thisrow{year}-\thisrow{month}-\thisrow{day}}]`’.

`pgfplotstodo.tex:4159` **Feature Proposal** [open, Priority 5, Epic ‘’]

What about a ‘draft’ mode which does nothing but typeset an empty axis without descriptions?

`pgfplotstodo.tex:4167` **Feature Proposal** [open, Priority 5, Epic ‘’]

Provide features of an axis *outside* of the axis environment. For a start, this could use the `axis cs` (or an alias to it).

Details and examples:

`https://sourceforge.net/tracker/?func=detail&atid=1060659&aid=3086794&group_id=224188`

`pgfplotstodo.tex:4171` **Feature Proposal** [open, Priority 5, Epic ‘’]

add ’force 2d axis’ key (or similar)

`pgfplotstodo.tex:4181` **Feature Proposal** [open, Priority 5, Epic ‘’]

could you extend the /tikz/prefix key so it also works as a prefix for imported files/tables? So far one has to type for example

```
 \addplot table {plots/data/test.txt};
```

If there would be a search path like `\graphicspath` for graphics it would be really nice.

See also `https://sourceforge.net/tracker/?func=detail&atid=1060659&aid=3020246&group_id=224188`

pgfplotstodo.tex:4189 **Feature Proposal** [open, Priority 5, Epic ‘’]

Support standard filters for `hist` and its variants.

Improve filtering for `hist` and similar plot handlers.

I already added the `hist/data filter` and `pre filter` keys (undocumented!). Use them.

pgfplotstodo.tex:4194 **Feature Proposal** [open, Priority 5, Epic ‘’]

the ’xtick’ syntax accepts only numbers, not even constant expressions are possible (and ’pi’ is even more complicated).

pgfplotstodo.tex:4199 **Feature Proposal** [open, Priority 5, Epic ‘’]

Table Package: support context–based `row predicate`s (some kind of WHERE clauses)

pgfplotstodo.tex:4203 **Feature Proposal** [open, Priority 5, Epic ‘’]

support the `/data point/x` method for all key filters and in all contexts (i.e. in the same context where `\thisrow` is accepted)

pgfplotstodo.tex:4207 **Feature Proposal** [open, Priority 5, Epic ‘’]

Support selection of individual 3D axis lines which shall be drawn (or “floor”)

pgfplotstodo.tex:4220 **Feature Proposal** [open, Priority 5, Epic ‘’]

linear regression which passes through (0,0) (see mail of Stefan Pinnow)

pgfplotstodo.tex:4225 **Feature Proposal** [open, Priority 5, Epic ‘’]

plot graphics 3D: handle the case when the first two points share the same x (or y) coordinate

pgfplotstodo.tex:4229 **Feature Proposal** [open, Priority 5, Epic ‘’]

hist does not allow modifications to the data range

pgfplotstodo.tex:4254 **Feature Proposal** [open, Priority 5, Epic ‘Bar plots’]

see the interesting things at `http://peltiertech.com/Excel/Charts/axes.html#Broken` broken (y) axis: remove interval [a,b] idea: if y¡a : visualize as usual if a¡y¡b : use coordinate y=a if b¡y : use coordinate y=y-(b-a) axis:

- compute two sets of axis descriptions. Perhaps one can try to compute the step size just once, and discard only [a,b] afterwards? This would require to use a canvas axis length corresponding to the unremoved axis range. BTW: I need access to the unremoved axis range; both for tick computation and for 'nodes near coords' or the clickable lib.
- draw a decoration at the break.
- perhaps also a decoration near affected coords.
- perhaps I should apply the thing during the visualization phase, not before. Then, I have all limits and the correct coordinates; only canvas coords are affected.

pgfplotstodo.tex:4263 **Feature Proposal** [open, Priority 5, Epic ‘’]

feature to replicate axis descriptions on both sides

pgfplotstodo.tex:4270 **Feature Proposal** [open, Priority 5, Epic ‘’]

couldn't you add something like `\providecommand*\pgfplotsset[1]{}` to the "tikzexternal.sty" so one doesn't have to do it by hand when switching from tikz/pgfplots?

pgfplotstodo.tex:4340 **Feature Proposal** [open, Priority 5, Epic ‘’]

Das pos= feature funktioniert nur ungenau fuer smooth. Implementiere es genauer.

pgfplotstodo.tex:4345 **Feature Proposal** [open, Priority 5, Epic ‘’]

make work `\matrix in \matrix` so one can use groupplots or "Allignment in Array Form" (section 4.18.4) with legends

pgfplotstodo.tex:4373 **Feature Proposal** [open, Priority 5, Epic ‘’]

groupplots: group-wide axis labels

pgfplotstodo.tex:4391 **Feature Proposal** [open, Priority 5, Epic ‘’]

is there a way to get the current row/col index during addplot?

pgfplotstodo.tex:4417 **Feature Proposal** [open, Priority 5, Epic ‘’]

plot
shell: - It would be nice if the standard shell interpreter could be replaced. Idea:
`\pgfkeys{/pgfplots/plot shell/interpreter/.code 2 args={sh #1 > #2}}`
then in the code
`\pgfkeysvalueof{/pgfplots/plot shell/interpreter/.@cmd}{#1.sh}{#1.out}\pgfeov`
- the pgfshell macro is quite general and could be added to pgf (as suggested by you, Stefan). However, this would also need modifications in tikz.code.tex to get some sort of high-level user interface. I find plot shell very useful, and it could be added easily. My suggestion: Either write a high level user interface for tikz or rename the command to pgfplotsshell and put it into pgfplotscoordprocessing.code.tex. In the meantime, I added it to pgfplotscoordprocessing.code.tex (bottom). - there is a potential difficulty with the 'addplot table shell' command (which is a good solution!): the semicolon in this routine will have a fixed catcode. But packages like babel with french language will change it to active, so french people can't use addplot table shell. The solution is technical and I am not proude of my own anyway... we'll just have to think about one. - documentation for the 'table shell' feature is missing yet. - I am not sure if the replication of /tikz/prefix and /tikz/id is helpful or confusing....

pgfplotstodo.tex:4435 **Feature Proposal** [open, Priority 5, Epic ‘’]

new `\plotnumofactualtype` thing: if you set /tikz/ plot handlers in `\begin{axis}`, they won't be set before the visualization phase. consequently, I can't count them! Idea: add a 'family' to each of them. Or wright a coord filter which checks for `\tikz@plot@handler` . Or write pgfplots styles which set them.

pgfplotstodo.tex:4560 **Feature Proposal** [open, Priority 5, Epic ‘’]

polar:

- is my current datascaling approach correct? I mean, is the linear trafo feasible at all?
- the *affine* radius datascaletrafo could be enabled, if only parts of the circle are drawn at all, for example xmin=0,xmax=45, ymin=1e-4,ymax=1.003e-4 Idea: check arc size and disable the radius *affine* data scaling only if the arc has more than 90 (?) degrees Is that mathematically correct? And: is it useful at all?
- handle "empty axis". It should reset to a circle, not a box.

pgfplotstodo.tex:4564 **Feature Proposal** [open, Priority 5, Epic ‘’]

patch visualization: provide displacement input format

pgfplotstodo.tex:4596 **Feature Proposal** [open, Priority 5, Epic ‘’]

the following keys should process their argument with pgfmathparse:

xyz tick,

- min/max
- tickmin/max
- meta min/max
- domain/ y domain,
- error bar arguments,
- without FPU: width/height/ view
- check optimizations of the math parser!
- check if I can activate the FPU during the survey phase!

TODO: check which ones are missing. Some of them have been added in the meantime

pgfplotstodo.tex:4606 **Feature Proposal** [open, Priority 5, Epic ‘’]

Idea for input stuff: implement high level user interface for coordinate input, similar to the pgf basic level framework. Then, add styles on top of it (try to be compatible with DV engine)

pgfplotstodo.tex:4628 **Feature Proposal** [open, Priority 5, Epic ‘’]

new plot structure : use the ‘/data point’ key interface coming with pgf CVS

pgfplotstodo.tex:4706 **Feature Proposal** [open, Priority 5, Epic ‘’]

quiver plots:

- allow to disable update of axis limits
- provide rescaling of arrows such that they don’t overlap. manual rescaling is simple, auto is more difficult. auto: if I have a matrix, I could rescale such that its mesh width is larger than the largest vector. Same fo a vector of input data. But what if I don’t know whether it’s a vector or matrix? ⇝ second run. ⇝ after the first, it should be possible

to autocomplete the mesh rows/cols. Try it. If that works, we have a matrix. ⇝ could be done from within the scanlinelength routines: auto-detect mesh/rows mesh/cols mesh/ordering mesh/width but that fails if there is no scanline marker.

- what with log plots? What with other axis features like symbolic trafos? ⇝ need difference type!
- that is: quiver plots in log coords are *multiplicative* and invoke the same routines. make special handling for '0'.
- allow feature where (u,v) are *coords*, not vectors. this could allow additive log quiver plots.

pgfplotstodo.tex:4710 **Feature Proposal** [open, Priority 5, Epic '']

plot expression: make the sampling parameters available within survey phase

pgfplotstodo.tex:4714 **Feature Proposal** [open, Priority 5, Epic '']

the table package uses a lot of logs – but it can't change the log basis.

pgfplotstodo.tex:4719 **Feature Proposal** [open, Priority 5, Epic '']

3D + axis line variants: someone might prefer GRID LINES as for the boxed case combined with axis line=left...

pgfplotstodo.tex:4763 **Feature Proposal** [open, Priority 5, Epic '']

Mail by Hubertus Bromberger:

- ✓Period in legend, without the need of using the math environment? `\legend{ML spcm$.$, CW spcm$.$, ML AC};`
- Maybe a more straight forward way for legend to implement something like shown in the graph. (see his mail .tex) ⇝ plot marks only at specific points. thus, the legend image should contain both lines and marks, but there are effectively two addplot commands.
- As a physicist, I often have the problem to fit curves. A job gnuplot can do very well. It should be possible using "raw gnuplot" but maybe you can either provide an example or even implement a more straight forward way for this purpose.
- The color scheme is not really my taste. In CONTEXT:

```
 cycle list={%
{Col1,mark=*},
{Col2,mark=square*},
{Col3,mark=diamond*},
{Col4,mark=star},
{Col5,mark=pentagon*},
{Col6,mark=square*},
{Col7,mark=diamond*},
{Col8,mark=triangle*} }}
  \definecolor[Col1][r=0.24106,g=0.05490,b=0.90588]  % blau
  \definecolor[Col2][r=1,g=0.05490,b=0.06667]        % rot
  \definecolor[Col3][r=0.65490,g=0.73333,b=0.01176]  % grn
  \definecolor[Col4][r=0.08627,g=0.92549,b=0.91373]  % tyrkis
  \definecolor[Col5][r=1,g=0.5,b=0]              % orange
  \definecolor[Col6][r=0.54118,g=0.51765,b=0.51765]  % grau
  \definecolor[Col7][r=0.80784,g=0.49804,b=0.06275]  % okker
  \definecolor[Col8][r=0.74902,g=0.07451,b=0.91765]  % lila
```

- Sometimes it would be good to have a bit more of a programming language, but still that's not what tex is made for. The python-script looks promising, it's just, that I think it doesn't work with context.

pgfplotstodo.tex:4772 **Feature Proposal** [open, Priority 5, Epic ‘’]

add something like

```
\pgfplotstabletypeset[
  cell { 1 }{ 2 }={\multirow{*}{3}{text}}
]
```

pgfplotstodo.tex:4816 **Feature Proposal** [open, Priority 5, Epic ‘’]

ternary diagrams todo:

- the `\pgfplotsqpointoutsideofaxis` work only for position 1, nothing in-between (since it doesn't compute the other axis components correctly)
- data ranges are currently only correct if in [0,1] or if one provides the [xyz]min and [xyz]max keys (and the ternary limits relative=false). How should it work!?

pgfplotstodo.tex:4833 **Feature Proposal** [open, Priority 5, Epic ‘’]

contour:

- labels=true,false,auto ⇝ auto should deactivate labels if there are too many contour lines.
- labels should not be clipped...
- add label position shifting facilities. ⇝ identify by contour label *and* an optional index. There may be more than one line.

pgfplotstodo.tex:4842 **Feature Proposal** [open, Priority 5, Epic ‘’]

DUPLICATE contourf: I guess filled contour plots could be possible if always two adjacent color levels are combined into a single path which is then filled with the simplified even/odd rule (not the winding fill rule). With the underlying smoothness assumption C^0, there can't be any level between two adjacent ones, and there can't be self-intersections.

pgfplotstodo.tex:4847 **Feature Proposal** [open, Priority 5, Epic ‘’]

it would be very interesting to allow more flexible handling of empty lines in input data, especially files.

pgfplotstodo.tex:4866 **Feature Proposal** [open, Priority 5, Epic ‘’]

contour draft TODO:

- color of text nodes
- make sure there is at least one label node
- implement contourf
  - often: use 'even odd rule' to fill adjacent contours.
  - but this works only if adjacent contours are contained in each other.
  - if that's not the case, perhaps I need to add an artifical path from the data limits.
  - idea: in case I know the corner values, I'd know which contour plateau requires the artifical path.
  - other idea: I could implement some sort of even-odd rule in TeX. This should also yield the information.

pgfplotstodo.tex:4872 **Feature Proposal** [open, Priority 5, Epic ‘’]

implement simplified constructions to access DIFFERENCE coordinates. For example, `\draw` ellipse needs x radius and y radius.

pgfplotstodo.tex:4893 **Feature Proposal** [open, Priority 5, Epic ‘’]

the ’table/y index’ should be changed. It should be min(numcols,1) instead of 1.

pgfplotstodo.tex:4904 **Feature Proposal** [open, Priority 5, Epic ‘’]

table package and axes should improve their communication. Namely:

-
- communicate table names.
- communicate xmode/ymode
- communicate log basis [xy]

pgfplotstodo.tex:4909 **Feature Proposal** [open, Priority 5, Epic ‘’]

provide and document access to (sanitized?) mesh/rows and mesh/cols fields during the survey phase. This might allow 2d key filters

pgfplotstodo.tex:4916 **Feature Proposal** [open, Priority 5, Epic ‘’]

Praktisch fände ich, wenn man folgende Dinge spezifizieren kann: 1. Welche Zeilen aus der Datei ausgelesen sollen (häufig gibt es nicht nur 1, sondern mehrere Header-Zeilen, oder auch am Ende noch sonstige Zeilen)

pgfplotstodo.tex:4943 **Feature Proposal** [open, Priority 5, Epic ‘’]

improve support for multiple ordinates

1.
   ```
   * \pgfplotsset{set layers}
   * scale only axis
   * xmin=..., xmax=...,
   * axis y line*=left
   * axis y line*=right
   * axis x line=none
   ```
   would be hidden in the doubleaxis definition,
2. the first addplot would be the left one and the second, the right one, (???)
3. the comma separated list in the legend command’s argument applies successively to the two addplot.
4. the colours of the two plots are given by the color cycle list.

pgfplotstodo.tex:4956 **Feature Proposal** [open, Priority 5, Epic ‘’]

it would be useful if the clipping could be disabled for certain parts of the axis. Is that possible?

- yes. Idea: start clipping for every axis element separately! Shouldn't be much more expensive than a single marker path.
- should work in the same way as before, there is no difference!
- scopes should introduce no further problems
- I could eliminate the nasty marker list

pgfplotstodo.tex:4961 **Feature Proposal** [open, Priority 5, Epic ‘’]

provide a `\pgfplotspathcube` command as generalization from the cube marker. The cube command should work similar to pathrectangle or rectanglecorners.

pgfplotstodo.tex:4968 **Feature Proposal** [open, Priority 5, Epic ‘’]

re-implement sampling loops. I should discard the compatibility with foreach internally in order to gain accuracy! Maybe it is necessary to invoke different loops - one for tikz foreach (samples at) and one ”standard” sampling routine.

pgfplotstodo.tex:5011 **Feature Proposal** [open, Priority 5, Epic ‘’]

optimization ideas:

- replace `\pgfpointscale` with a ’q’ version ⇝ it invokes the expensive math parser.
- pgfmultipartnode evaluates every anchor twice
- implement a cache for expensive, repeated math operations like ’view’ directions or common results of $1/||e_i||$.
- search for unnecessary math parser invocations; replace with ’q’ versions if possible.
- implement a hierarchical generalization of the ’applist’ container (a tree applist of arbitrary length)
- eliminate the deprecated ’non-legend-option’ processing.
- remove the different (empty) paths of the axis node – it appears they are not necessary and waste only time and mem.

- try implementing an abstract 'serialize' and 'unserialize' method - it might be faster to re-process input streams instead of generating preprocessed coordinate lists.
- try to reduce invocations of pgfkeys
- optimize the filtered pgfkeys invocations - the filter is slower than necessary!
- the plot mark code invokes a lot of math parsing routines - which is a waste of time in my opinion. All expressions etc. have already been parsed.
- the point meta transform is set up twice for scatter plots.
- my elementary data structures always use `\string` to support macros as data structure names. I fear this might be ineffective. Perhaps its better to check if the argument is a macro (at creation time, thus only once) and call `\edef#1{\string#1}` to assign some sort of name to it. This will invoke `\string` only once. Is this faster?
- eliminate the 'veclength' invocations for single axes - they can be replaced with "inverse unit length * (max-min)"
- the key setting things can be optimized with pgfkeysdef
- create the /pgfplots/.unknown handler (.search also=/tikz) once and remember it.
- the (new) tick label code might be very expensive:
  - check for (unnecessary) calls to `\pgfpointnormalised` – the normal vectors are already normalised!
  - check the cost for bounding box size control of the tick labels – maybe this can be optimized away if it is not used. But this decision is not easy.

`pgfplotstodo.tex:5016` **Feature Proposal** [open, Priority 5, Epic ‘’]

perhaps math style
`{grid=major, axis x line=middle, axis y line=center, tick align=outside}`

`pgfplotstodo.tex:5025` **Feature Proposal** [open, Priority 5, Epic ‘’]

provide access to axis limits and data bounding box. It would be useful to get access to axis coordinates, for example in 'circle (XXX)'

`pgfplotstodo.tex:5030` **Feature Proposal** [open, Priority 5, Epic ‘’]

allow math expressions for axis limits etc. Idea: try float parsing routine; if it fails: use math parser first.

pgfplotstodo.tex:5037 **Feature Proposal** [open, Priority 5, Epic ‘’]

write a public math interface which provides access to axis internals like limits, the ’dimen-to-coordinate’ method and so on. ⇝ it might be useful to use pgfmathparse for any numerical input argument as well.

pgfplotstodo.tex:5073 **Feature Proposal** [open, Priority 5, Epic ‘’]

Store the axis limits into the axis’ node as saved macros. This would allow

- ’use [xy] limits of=¡axis name¿’
- access to axis limits from other macros.
- provide a command `\pgfplotslimits{current axis}{x}{min}` which expands to the ’xmin’ limit. PROBLEM: to WHICH limit: the untransformed one? The transformed one? The logarithmized one?
    - ¿ I can’t compute exp(xmin) in log plots!
    - Ideas:
    - provide both, if possible. It is NOT possible for log axes.
    - use log-limits (possibly combined with ’logxmin=’ option ?)
    - The operation requires several operations because floats need to be converted. Idea: do that only for NAMED AXES.
    - all user-interface macros must be expandable!
    - I don’t want to spent time for number format conversions unnecessarily here!
    - provide `\pgfplotslimits` and `\pgfplotstransformedlimits` combined with simpler key-value interfaces
    - I could also provide access to the unit lengths (they are available as macro anyway)
    - ALTERNATIVE: implement access to axis limits as a math function which simply defines `\pgfmathresult`.
    - that is probably the most efficient way to do it. I only need to register the new function(s) to PGF MATH.
    - PGF 2.00: use `\csname pgfmath@parsefunction@\pgfmath@parsedfunctionname\endcsname`
    - PGF ¿ 2.00: use `\pgfmathdeclarefunction` Is it possible to provide ’string’ arguments which are not parsed? No.

pgfplotstodo.tex:5079 **Feature Proposal** [open, Priority 5, Epic ‘’]

I could provide public macros for the data transformations (and inverse transformations). This would also allow relatively simple access to axis limits.

pgfplotstodo.tex:5083 **Feature Proposal** [open, Priority 5, Epic ‘’]

cycle list should be implemented using an array structure. That’s faster.

pgfplotstodo.tex:5089 **Feature Proposal** [open, Priority 5, Epic ‘’]

what about a feature like ’draw[xmin=...,xmax=...] fitline between points (a) (b)’?

pgfplotstodo.tex:5093 **Feature Proposal** [open, Priority 5, Epic ‘’]

interpolate missing coordinates for stacked plots.

pgfplotstodo.tex:5099 **Feature Proposal** [open, Priority 5, Epic ‘’]

the error bar implementation is relatively inefficient. Think about something like ’/pgfplots/error bars/prepare drawing’ which sets common style keys for every error bar

pgfplotstodo.tex:5129 **Feature Proposal** [open, Priority 5, Epic ‘’]

think about using a combination of the visualization engine of pgf CVS and my prepared-list-structure. Maybe I can adjust the list format for the current plot type? I need

- scatter/line plots 2D
- meta coords
- quiver may need extra vectors
- matrix plots may need twodimensional structure
- error bars could be handled more consistently
- ...
- ¿ implement a visualization class which provides methods
  - prepare()
  - visualize()
  - serialize()
  - visualizestream() and provide protected pgfplots methods
  - axis⇝preprocesscoordinate (filters, logs)
  - visualizer⇝prepare()
  - axis⇝processcoordinate()
  - visualizer⇝serialize()

  - axis⇝postprocesscoordinate() The markers as they are implememted now don’t really fit into this framework. The clipping region is not really what I want here... Idea: enable/disable clipping separately for each drawing command!

pgfplotstodo.tex:5133 **Feature Proposal** [open, Priority 5, Epic ‘’]

the coordindex shouldn’t be changed by z buffer=sort

pgfplotstodo.tex:5166 **Feature Proposal** [open, Priority 5, Epic ‘’]

table package: provide abstract layer for low level storage interface. Idea: the interface should allow the container interface

- push_back()
- get(i)
- set(i)
- foreach()
- pop_front()
- newempty()
- clone()
- unscope()
- startPushBackSequence()
- stopPushBackSequence()

⇝ this could allow to use arrays for fast algorithms. At least it would make things easier to read. Problem as always: the ’unscope()’ operation. Currently, I have two different structures: the applists which have fast construction properties and the standard lists which implement the rest. Can I combine both? Yes, by means of the incremental construction pattern:

```
\startPushBackSequence
\push_back
\push_back
\push_back
\stopPushBackSequence
```

⇝ inside of the construction, only `\push_back` is allowed and the structure is in ”locked state” (low level: applist repr) ⇝ Idea: the creation is fast, afterwards, it has flexibility.

pgfplotstodo.tex:5222 **Feature Proposal** [open, Priority 5, Epic ‘’]

support \multicolumn for legends

pgfplotstodo.tex:5227 **Feature Proposal** [open, Priority 5, Epic ‘’]

it appears line breaks in legend descriptions are a problem (?) ⇝ bug in pgf: \\ is overwritten and won’t be restored.

pgfplotstodo.tex:5235 **Feature Proposal** [open, Priority 5, Epic ‘’]

pgfplotstable file open protocol: provide public listener interface

pgfplotstodo.tex:5239 **Feature Proposal** [open, Priority 5, Epic ‘’]

```
\addplot coordinates {\macro};
```

pgfplotstodo.tex:5263 **Feature Proposal** [open, Priority 5, Epic ‘’]

precise width calculation idea:

- Problem: total width depends on width of axis descriptions
- width of axis descriptions depends on position of axis descriptions
- position of axis descriptions depends on width of axis
- width of axis depends on width of axis descriptions
- non-linearly coupled system.
- Idea: introduce a loop.
  - details:
    1. place axis descriptions + the axis rectangle into a box.
    2. Measure box’es width, throw it away if it is too bad. Keep it and stop iteration otherwise.
    3. recompute the complete scaling.
    4. go back to step 1.) and iterate
  - one or two iterations should be enough .
  - it’s not necessary to recompute the prepared and stored plots. Just keep them in main memory until the scaling is fixed.

pgfplotstodo.tex:3838 **Feature Proposal** [open, Priority 4, Epic ‘’]

Optimization: if the same plot contains `name intersections` and `fill between` or `intersection segments`: compute the intersection points just once

pgfplotstodo.tex:3852 **Feature Proposal** [open, Priority 4, Epic ‘’]

regression line computation: also generate macros containing the coefficient of determination

`http://en.wikipedia.org/wiki/Coefficient_of_determination`

`https://tex.stackexchange.com/questions/147249/pgfplots-linear-regression-mean-square-error`

pgfplotstodo.tex:3823 **Feature Proposal** [open, Priority 3, Epic ‘’]

add support for symlogs `http://stackoverflow.com/questions/3305865/what-is-the-difference-between-log-and-symlog`

pgfplotstodo.tex:4525 **Feature Proposal** [open, Priority 3, Epic ‘Performance’]

disable bounding box updated during addplot – it makes no sense and wastes time (unless the axis is hidden)

pgfplotstodo.tex:3819 **Feature Proposal** [open, Priority 2, Epic ‘’]

faceted allows to draw the entire rectangle.

It would allow cool effects if one could draw only those lines along the x direction (or y direction).

Compare `http://tex.stackexchange.com/questions/166768/draw-a-surface-from-scattered-curves`

pgfplotstodo.tex:4363 **Feature Proposal** [open, Priority 1, Epic ‘’]

nested axes would be a nice feature. TODO: - update the list of global state variables - ”interrupt” these variables somehow. - make sure local redefinitions of TikZ commands (like point commands) work; the `\let...@orig=` assignments should be handled somehow. - What about keys? They will be inherited from the outer axis... perhaps the best would be an

```
\endgroup
<nested axis>
\begingroup
<restore state>
```

which includes the keys of the outer axis!?

pgfplotstodo.tex:4387 **Feature Proposal** [open, Priority 1, Epic ‘’]

log plots: minor tick num would be useful here! If tick labels are placed at ’1e-5, 1e0’, minor tick num= 4 would lead to the minor tick lines at ’1e-4,1e-3,1e-2,1e-1’ which is useful. So:allow minor tick num for log axes. ⇝ need to adjust the check for ”uniform log ticks”

pgfplotstodo.tex:4821 **Feature Proposal** [cancelled, Priority 5, Epic ‘’]

idea: ’mesh/ordering=auto’. Just check for ’x varies’ and ’y varies’! The two first points inside of a scanline are enough.