Todolist for PGFPlots

`(git show version_1.4-473-gef30385 )`

Christian Feuersänger

July 28, 2010

Contents

1 Tests

last test verifications:

	pgf CVS	pgf 2.00	pgf 2.00+compat=default
pgfplotstests	for 1.4	for 1.4	2009-12-30
manual	for 1.4	for 1.4	
pgfplotstable.pdf	for 1.4	for 1.4	
example latex	for 1.4	2009-12-30	
example context	for 1.4	2009-12-30	
example plain tex	for 1.4	2009-12-30	
tests context	for 1.4		

2 Components planned for version 1.5

- hist 98% (complete UI?)
- quiver 95% (test, shall I create an UI for autoscaling of arrows?)
- contours 95% ready. add "labels=if less than X"

perhaps contour filled 10% ready

- patch plots lib: 95%
  - implement displacement input

  - bug for quad rectangle in middle point
  - perhaps 1d quadratic/cubic patches (simple)?

- polar axes: 70% ready
  - undocumented
  - missing feature: input of cartesian coords
  - special cases (?)
- ternary: 80% ready
  - but there are still quite a lot of feature request concerning them
  - konnodalplots (? implement in new branch to check difficulty)
- smith charts: 0% ready, but implementation should be simple with what I have now

  → implement in new branch
- internal coordmath framework: 80% ready, but not used everywhere and undocumented
- layer graphics support for axes 0% (should be easy, implement in new branch)
- view configuration:
  - ✓document gnuplot import/export
  - view matrix input?
- check 'empty lines' feature -¿ should have compat mode
- new public axis API is 90% complete: documentation is missing log scaling is difficult, still
- Bugfixes
  - what about 'scale' transformations? Are they correct?

pgfplotstodo.tex:104 **Bug** [open, Priority 5]

Test fuer bugtracker:

```
\documentclass{article}

\begin{document}
Hallo Welt
\end{document}
```

3 Documentation todo

There's a tiny typo in pgfplots manual: `\addplot table[x index=0,y index=0,header=false]` on page 18. The second index should be 1, not 1.

pgfplotstodo.tex:112 **Documentation Todo** [open, Priority 5]
release notes: mention improvements of 'shader=interp'

pgfplotstodo.tex:116 **Documentation Todo** [open, Priority 5]
new 'output cs' feature (when it is finished)

pgfplotstodo.tex:125 **Documentation Todo** [open, Priority 5]
quiver: the tests have a further pretty example where quiver is on top of a surf, attached to z =2 or so.

pgfplotstodo.tex:133 **Documentation Todo** [open, Priority 5]
document 'mesh/type'

pgfplotstodo.tex:141 **Documentation Todo** [open, Priority 5]
try a bar plot with individually shaded bars

pgfplotstodo.tex:149 **Documentation Todo** [open, Priority 5]
contour external: Do not forget the `\", \'` etc special handling .

pgfplotstodo.tex:153 **Documentation Todo** [open, Priority 5]
contour: document 'labels over line' style

pgfplotstodo.tex:157 **Documentation Todo** [open, Priority 5]
contour: a change label dist

pgfplotstodo.tex:161 **Documentation Todo** [open, Priority 5]
contour: document the special handling of "point meta".

pgfplotstodo.tex:178 **Documentation Todo** [open, Priority 5]
document frac whole format

pgfplotstodo.tex:182 **Documentation Todo** [open, Priority 5]
document /pgfplots/empty line

pgfplotstodo.tex:186 **Documentation Todo** [open, Priority 5]

document 'clickable coords' and 'clickable coords code' features

pgfplotstodo.tex:190 **Documentation Todo** [open, Priority 5]

document 'execute at begin axis' and its new variants

pgfplotstodo.tex:198 **Documentation Todo** [open, Priority 5]

document how to plot against the coordindex

pgfplotstodo.tex:202 **Documentation Todo** [open, Priority 5]

document the new 'getcolumnbyname=create col/....' feature

pgfplotstodo.tex:211 **Documentation Todo** [open, Priority 5]

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

pgfplotstodo.tex:216 **Documentation Todo** [open, Priority 5]

document how to fix dimension too large problems: restrict to domain for example

pgfplotstodo.tex:220 **Documentation Todo** [open, Priority 5]

colorbar styles are not consistent between docs and code

pgfplotstodo.tex:226 **Documentation Todo** [open, Priority 5]

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:230 **Documentation Todo** [open, Priority 5]

pgfplotstable: show how to use '`\begin{longtable}`'

pgfplotstodo.tex:235 **Documentation Todo** [open, Priority 5]

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

pgfplotstodo.tex:120 **Documentation Todo** [closed, Priority 5]
There is a typo on section 4.5.12: "As for for dimensional patch plots "

pgfplotstodo.tex:129 **Documentation Todo** [closed, Priority 5]
document 'shader=faceted interp'

pgfplotstodo.tex:137 **Documentation Todo** [closed, Priority 5]
document the 'plot graphics/points' feature.

pgfplotstodo.tex:145 **Documentation Todo** [closed, Priority 5]
document 'contour prepared', 'contour external' and 'contour gnuplot'.

pgfplotstodo.tex:169 **Documentation Todo** [closed, Priority 5]
clickable: document 'popup size' and its variants document 'clickable coords size' document 'richtext' and the formatting things document `\n` and friends

pgfplotstodo.tex:174 **Documentation Todo** [closed, Priority 5]
document ternary lib + do not forget 'cartesian cs' and its applications

pgfplotstodo.tex:206 **Documentation Todo** [closed, Priority 5]
document linear regression

4 Bugs in PGF/TikZ

pgfplotstodo.tex:258 **pgfbug** [open, Priority 5]
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:265 **pgfbug** [open, Priority 5]

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:269 **pgfbug** [open, Priority 5]

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

pgfplotstodo.tex:273 **pgfbug** [open, Priority 5]

implement `\pgfmathfloattocount`

pgfplotstodo.tex:278 **pgfbug** [open, Priority 5]

external lib: think whether it is possible to provide the real jobname without explicit user input. Idea: transport it as TeX code argument to pdflatex

pgfplotstodo.tex:282 **pgfbug** [open, Priority 5]

provide '×' or more general formatting rules to number printer

pgfplotstodo.tex:286 **pgfbug** [open, Priority 5]

code 2 args doesn't work correctly with spaces between the arguments!?

pgfplotstodo.tex:312 **pgfbug** [open, Priority 5]

include addition of Christophe Jorssen for MD5 checksums in external lib

pgfplotstodo.tex:327 **pgfbug** [open, Priority 5]

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:347 **pgfbug** [open, Priority 5]

fix landscape bug (pdflscape) in external lib (PGF)

pgfplotstodo.tex:356 **pgfbug** [open, Priority 5]

pack the default 'system call' for dvips etc into drivers!

pgfplotstodo.tex:360 **pgfbug** [open, Priority 5]

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

pgfplotstodo.tex:364 **pgfbug** [open, Priority 5]

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

pgfplotstodo.tex:370 **pgfbug** [open, Priority 5]

compatiblity code todo: - the example for plot graphics (with view=090) doesn't work. ⇝ that's the '`exp(0-x^2)`' bug which is still in pgf 2.00!

pgfplotstodo.tex:291 **pgfbug** [closed, Priority 5]

external lib: implement `\tikzpicturedependsonfile#1`

pgfplotstodo.tex:296 **pgfbug** [closed, Priority 5]

in pgfplots: invoke `\tikzpicturedependsonfile`. perhaps the plot-from-table-struct should also use it.

pgfplotstodo.tex:300 **pgfbug** [closed, Priority 5]

external lib: 'list and make' does not work together with `\include` (aux files!) or other file writing things – at least not if one tries to do that in parallel.

pgfplotstodo.tex:304 **pgfbug** [closed, Priority 5]

consider the "plot function" patch from Andy Schlaikjer

pgfplotstodo.tex:308 **pgfbug** [closed, Priority 5]

it seems fadings don't work correctly with externalization!?

pgfplotstodo.tex:316 **pgfbug** [closed, Priority 5]

write new sub-package 'pgfmanual.sty' which contains a good user interface to the manual styles, environments and all that.

pgfplotstodo.tex:320 **pgfbug** [closed, Priority 5]

external lib: catcode changes inside of pictures do not work properly.

pgfplotstodo.tex:332 **pgfbug** [closed, Priority 5]

in the manual, the first two arguments of pgfqkeysactivatesinglefamilyandfilteroptions were inverted.

pgfplotstodo.tex:336 **pgfbug** [closed, Priority 5]

some predefined filters do not process unknown options correctly

pgfplotstodo.tex:343 **pgfbug** [closed, Priority 5]

external lib in pgf: think whether 'empty image extension' is a bug or a feature. ⇝ feature of `\pgfimage`! Otherwise it wouldn't be possible to provide an extension! ⇝ bug for external lib which never uses extensions!

pgfplotstodo.tex:352 **pgfbug** [closed, Priority 5]

the pgf math parser has wrong precedence for '-' prefix op: `exp(-x^2)` is wrong.

5 Bugs in PGFPlots

pgfplotstodo.tex:381 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:386 **Bug** [open, Priority 5]

external lib + dvi/ps + windows: it seems the ';' doesn't work; use '&' to separate commands

pgfplotstodo.tex:390 **Bug** [open, Priority 5]

check y tick scale label for 2nd y axis

pgfplotstodo.tex:400 **Bug** [open, Priority 5]

foreach variants in pgfplots accept only one parameter

```
%            \foreach \x/\y in {1/a, 2/b, 3/c}
%                {\node at (axis cs:0,\x) {\y};}%   % doesn't work
%            \pgfplotsforeachungrouped \x/\y in {1/a, 2/b, 3/c}
%                {\node at (axis cs:0,\x) {\y};}%   % doesn't work
```

pgfplotstodo.tex:407 **Bug** [open, Priority 5]

groupplots + extra braces or foreach are incompatible.

pgfplotstodo.tex:412 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:416 **Bug** [open, Priority 5]

document INS 671 does not compile with pgfplots 1.4 and pgf 2.00

pgfplotstodo.tex:420 **Bug** [open, Priority 5]

numplotspertype and forget plot and ybar interval yields errors.

pgfplotstodo.tex:426 **Bug** [open, Priority 5]

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:430 **Bug** [open, Priority 5]

view normal vector does not correctly respect plot box ratio and x dir

pgfplotstodo.tex:434 **Bug** [open, Priority 5]

plot box ratio has a strange input format (compare with unit vector ratio).

pgfplotstodo.tex:438 **Bug** [open, Priority 5]

clickable and Windows Acrobat Reader 9 has been reported to fail

pgfplotstodo.tex:444 **Bug** [open, Priority 5]

/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:449 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:454 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:458 **Bug** [open, Priority 5]

'clip=false' does not disable marker clipping!

pgfplotstodo.tex:468 **Bug** [open, Priority 5]

multiple ordinates: grid lines are drawn on top of function plots; that's bad. Check: I think you have to change the process line previousely invoked, and make the axes generation at the end : 1. generating adequate grid -⇝ 2. plotting functions -⇝ 3. creating axes, tick nodes... You can take a minute look at figure 1 @ "The addplot Command: Coordinate Input" section 4.2 p 19. and you can remark that colour filling overlaps x- and y-axis ! So I suggest that you use "excute at end picture=¡axis generation code¿" tikz option or similar to avoid this issue.

pgfplotstodo.tex:472 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:480 **Bug** [open, Priority 5]

3D axes: providing three unit vectors is not sufficient, one also needs to set 'view='. That should be done automatically.

- 3D axes: Providing three unit vectors manually yields incorrect axis initialisation.

pgfplotstodo.tex:488 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:493 **Bug** [open, Priority 5]

Patch plots: directly transform cdata. Should simplify interpolation during refine/triangulation etc. and shouldn't make a difference otherwise.

pgfplotstodo.tex:500 **Bug** [open, Priority 5]

manual errors of given pgfplots_unstable version: 94 2.5.12
`addplot+[patch] --> addplot3+[patch]` 162 "xmode, ymode, zmode" and "x dir, ..." come again on page 177

pgfplotstodo.tex:505 **Bug** [open, Priority 5]

don't loose `\ref`'s when externalizing I'll provide a minimal later

pgfplotstodo.tex:511 **Bug** [open, Priority 5]

incompatibility pdfpages (most recent version), MikTeX and tikz external lib (something with shipout routine)

pgfplotstodo.tex:515 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:521 **Bug** [open, Priority 5]

french babel and colorbars are not fully compatible. The problem is that colorbars use '`\addplot` graphics ;' with a fixed catcode for the ';' – which might lead to problems.

pgfplotstodo.tex:527 **Bug** [open, Priority 5]

markers should not be drawn on top of everything else. Always restore the clipping region for each plot.

pgfplotstodo.tex:532 **Bug** [open, Priority 5]

mesh/patch plots: - jump thing + z buffer=sort probably doesn't work

pgfplotstodo.tex:536 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:546 **Bug** [open, Priority 5]

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:551 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:557 **Bug** [open, Priority 5]

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:564 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:570 **Bug** [open, Priority 5]

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:575 **Bug** [open, Priority 5]

provide remark at end document "Package pgfplots: consider using the preamble command
`\pgfplotsset{compat=1.3} to improve label placement`"

pgfplotstodo.tex:580 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:586 **Bug** [open, Priority 5]

contour external should allow different variations how to deal with end-of-scanline markers. gnuplot requires empty lines; matlab doesn't deal with them as far as I know.

pgfplotstodo.tex:591 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:595 **Bug** [open, Priority 5]

contour external doesn't handle the ordering flag correctly.

pgfplotstodo.tex:600 **Bug** [open, Priority 5]

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:605 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:610 **Bug** [open, Priority 5]

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:621 **Bug** [open, Priority 5]

avoid dimension too large errors which occur due to a data range restrictions. Example: data range = 0:6000 view range = 0:1 $\rightsquigarrow$ 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?

pgfplotstodo.tex:628 **Bug** [open, Priority 5]

the autodetection of the '\\' list format is buggy: it should return true if and only if the last element is '\\', not if '\\' occurs inside of the argument.

pgfplotstodo.tex:635 **Bug** [open, Priority 5]

`|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:640 **Bug** [open, Priority 5]

'`\addplot[only marks]`' does not assign a plot mark; one needs 'mark=*' explicitly. that's confusing...

pgfplotstodo.tex:647 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:652 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:657 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:661 **Bug** [open, Priority 5]

FPU: atan doesn't check for unbounded inputs.

pgfplotstodo.tex:666 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:672 **Bug** [open, Priority 5]

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... $\rightsquigarrow$ 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:676 **Bug** [open, Priority 5]

the FPU has somewhere spaces; at least it treats them differently from pgfmathparse.

pgfplotstodo.tex:684 **Bug** [open, Priority 5]

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:688 **Bug** [open, Priority 5]

axis lines and 3D: some tick lines are not drawn, see manual examples

pgfplotstodo.tex:695 **Bug** [open, Priority 5]

check for placement of tick scale label for compat=newest ⇝ I improved them for 2d and 3d ⇝ needs some further checks, I guess

pgfplotstodo.tex:699 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:703 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:707 **Bug** [open, Priority 5]

getthisrow still has to be fixed

pgfplotstodo.tex:712 **Bug** [open, Priority 5]

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:716 **Bug** [open, Priority 5]

log samples in plot expression for 3D plots

pgfplotstodo.tex:720 **Bug** [open, Priority 5]

different log bases and gnuplot

pgfplotstodo.tex:734 **Bug** [open, Priority 5]

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:741 **Bug** [open, Priority 5]

the compatibility things should be enabled automatically for some parts: - 3D - color bars - reversed axes

pgfplotstodo.tex:747 **Bug** [open, Priority 5]

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:756 **Bug** [open, Priority 5]

gnuplot and 3D $\rightsquigarrow$ I need a shared interface to prepare the required keys for expression plotting

pgfplotstodo.tex:760 **Bug** [open, Priority 5]

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:765 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:772 **Bug** [open, Priority 5]

check whether /pgfplots/ keys are processed properly in legends. This is certainly not the case for the `\label/\ref` legend! $\rightsquigarrow$ which ones are the problem?

pgfplotstodo.tex:776 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:781 **Bug** [open, Priority 5]

axis equal for semilog plots is not correct (?)

pgfplotstodo.tex:789 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:794 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:802 **Bug** [open, Priority 5]

finish impl of ticklabel pos. I should use the same thing for tickpos as well. And: the default arg processing which uses ticklabel pos = tickpos needs to be fixed. the 2D axes are wrong.

pgfplotstodo.tex:811 **Bug** [open, Priority 5]

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:817 **Bug** [open, Priority 5]

3D: axis equal implementation might not be correct (at least not for view special cases)

pgfplotstodo.tex:824 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:828 **Bug** [open, Priority 5]

the `\thisrow` commands in the table package does not (always) respect aliases!

pgfplotstodo.tex:832 **Bug** [open, Priority 5]

interp shader is displayed transparently in evince

pgfplotstodo.tex:859 **Bug** [open, Priority 5]

```
  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:864 **Bug** [open, Priority 5]

think about basic level commands for the axis lines – this should also allow !

pgfplotstodo.tex:871 **Bug** [open, Priority 5]

the arguments to `plot file[#1] and plot table[#1]` are not consistent with rest. They need to be treated as behavior options (maybe in a different key path).

pgfplotstodo.tex:876 **Bug** [open, Priority 5]

check behavior options provided to table[] and file[] and so - is that correct?

pgfplotstodo.tex:897 **Bug** [open, Priority 5]

verify that 'draw=none' works! Is something broken here? $\rightsquigarrow$ write tests! + it appears to be desired that (at least some) markers invoke `\pgfusepathqfillstroke` $\rightsquigarrow$ they always 'draw', regardless of tikz color settings. $\rightsquigarrow$ ok, I patched that in my marker code... (hackery :-( ) - no, it works only partially: draw=none or fill=none works as expected. But 'blue'

disables filling!? - Possible fix: Overwrite
`\filltrue \fillfalse, \drawtrue, \drawfalse`: they should set a further boolean '`\drawbooleanhasbeenset`' and '`\fillbooleanhasbeenset`'. ⇝ Replace the `\pgfusepathqfillstroke` if and only if the respective booleans have been set *explicitly*. If they are unchanged, fall back to a "reasonable" default.

pgfplotstodo.tex:902 **Bug** [open, Priority 5]

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:907 **Bug** [open, Priority 5]

3D case : grid lines work correctly, but they are not satisfactory. I'd like grid lines in the background only.

pgfplotstodo.tex:912 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:935 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:950 **Bug** [open, Priority 5]

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:957 **Bug** [open, Priority 5]

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

pgfplotstodo.tex:962 **Bug** [open, Priority 5]

extra ticks can be disabled by the tick special cases for axis lines (when two axis lines cross each other)

pgfplotstodo.tex:724 **Bug** [closed, Priority 5]

3D gnuplot: z buffer fails (see tests)

pgfplotstodo.tex:751 **Bug** [closed, Priority 5]

gnuplot: set terminal table seems to be deprecated.

pgfplotstodo.tex:931 **Bug** [closed, Priority 5]

the clickable library does *not* work inside of figure environments ⇝ yes. That's fixed; was a bug in hyperref. - I could try to re-implement it without insdljs. Ideas: - the document catalog's names dictionary needs to '/JavaScript [(¡arbitrary script name¿) ¡dictionary with JS¿]' entry. The ¡dictionary with JS¿ contains document level javascript. - it is very simple to generate these entries for my case. Unfortunately, this may be incompatible with 'insdljs' or other tools which write DLJS. - I am not sure why the floating figures of TeX produce an incompatibility here. It appears the 'hidden' flag in the form fields are the problem - if that is the case, I'd need to reimplement the form annotations (which could be more difficult).

6 Feature Proposals PGFPlots

pgfplotstodo.tex:972 **Feature Proposal** [open, Priority 5]

bar plots: provide constant zero level?

pgfplotstodo.tex:977 **Feature Proposal** [open, Priority 5]

implement properly layered graphics — especially for grid lines should probably also respect multiple ordinates

pgfplotstodo.tex:981 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:986 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:990 **Feature Proposal** [open, Priority 5]

hist does not allow modifications to the data range

pgfplotstodo.tex:1015 **Feature Proposal** [open, Priority 5]

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:1020 **Feature Proposal** [open, Priority 5]

plot graphics for 3D axes.

pgfplotstodo.tex:1024 **Feature Proposal** [open, Priority 5]

feature to replicate axis descriptions on both sides

pgfplotstodo.tex:1029 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1036 **Feature Proposal** [open, Priority 5]

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:1042 **Feature Proposal** [open, Priority 5]

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`)

pgfplotstodo.tex:1046 **Feature Proposal** [open, Priority 5]

ternary diagram for extractions (more details will come)

pgfplotstodo.tex:1052 **Feature Proposal** [open, Priority 5]

filled area between 2 addplot's (already requested in mailing list) perhaps style 'fill plot' which is applied in vis phase. There, one can access the postprocessed information of the previous plot.

pgfplotstodo.tex:1057 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1075 **Feature Proposal** [open, Priority 5]

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:1079 **Feature Proposal** [open, Priority 5]

support for "spy"glass into particular parts of an axis

pgfplotstodo.tex:1083 **Feature Proposal** [open, Priority 5]

groupplots: group-wide axis labels

pgfplotstodo.tex:1087 **Feature Proposal** [open, Priority 5]

It would be really great to have the possibility to attach a style to every nth row of a data table. For example, I would like to have a `\midrule` not after every line or after odd/even lines but after every fifth (or whatever) line.

pgfplotstodo.tex:1101 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1127 **Feature Proposal** [open, Priority 5]

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:1135 **Feature Proposal** [open, Priority 5]

¿ Is it possible to shade the area between two curves, using pgfplots, such as ¿ in this example:
`http://www.mathworks.com/matlabcentral/fileexchange/13188` ¿ The only shading I could find is between one curve and the x axis... Shading ¿ between curves seems to be possible, but only with stacked curves. Is is ¿ possible to disable stacking somehow, but keep the closedcycle behavior?

pgfplotstodo.tex:1143 **Feature Proposal** [open, Priority 5]

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:1156 **Feature Proposal** [open, Priority 5]

feature request for line styles in tikz/pgf or pgfplots respectively: add dash-dotted line which is quite commen in engineering field for example something like

```
\tikzset{
dash-dot/.style={
dash pattern=on 4pt off 3pt on 1pt off 3pt,
},
}
```

pgfplotstodo.tex:1228 **Feature Proposal** [open, Priority 5]

Konnodalplots fuer Ternary Axes

given: pairs of points (A_i, B_i) with $A_i, B_i \in R^3$ for the connodals

aim: connect $A_i - -B_i$ for each i *and* create the binodal line
$A_1 - -A_2 - - \cdots A_n - -B_n - -B_{n-1} - - \cdots B_1$

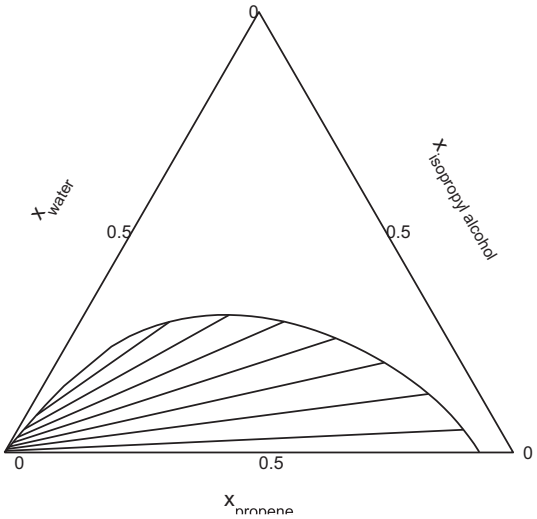
Remarks of stefan:

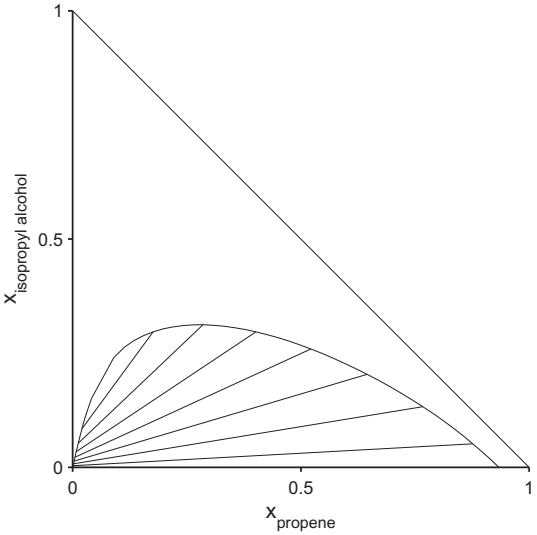
Im Anhang ist ein Beispiel gezeigt, wie es gehen könnte.

Noch einmal zur Klärung der Begriffe, mit denen ich gleich argumentieren werde:

- Binodale: Kurve
- Konode(n): Gerade(n) [engl.: tie line]
- Kritischer Entmischungspunkt: Ist der Punkt, an dem die beiden Punkte der Konode zusammenfallen. (nicht eingezeichnet)
- Mischungslücke: Das Gebiet, was von der Binodalen eingeschlossen wird. [engl.: miscibility gap]

Im Anhang findest du zum Einen die Daten-Datei und zwei mögliche Darstellungsformen. Das "`gibbs_phase_diagram`" ist die Darstellung im Dreieckdiagram (was auch Gibbs'sches Phasendiagramm oder Gibbs'sches Phasendreieck genannt wird); "`cartesian_phase_diagram`" entsprechend im Kartesischen Phasendiagramm.





Wenn man die Daten generiert, bekommt man üblicherweise 2 Matrizen mit den jeweiligen Zusammensetzungen an den Enden der Konoden (A_y bzw. B_y, wobei y die jeweilige Komponente ist). Diese kann man dann einfach nebeneinander setzen und erhält z.B. das mitgelieferte Textfile. Jetzt könnte man schon einmal die Binodale zeichnen. Dazu generiert mein Kollege in Matlab eine neue Matrix, indem er die UpDownGeflippte-Matrix B unter die Matrix A hängt und diese dann zeichnen lässt. Damit die Binodale "schön rund" ist, erzeugt man häufig mehr Punktepaare, als man nachher als Konoden anzeigen lassen möchte. In den mitgelieferten Plots ist so nur jede 5. Konode eingezeichnet.

Die Frage ist nun, wie man das Abfragen der Konoden gestalten kann. Dafür gäbe es jetzt die Möglichkeit einen Key zu erstellen, der sowas sagt wie "plot every Xth tie line".

Ich denke mal, du brauchst auch noch einen schönen Namen den Aufruf dieses Spezialfalls. Da diese zum Zeichnen von Mischungslücken dient, wäre der Englische Name dafür (s.o.) eine Möglichkeit.

was mir noch eingefallen ist:

- Zuweisung der Spalten Es sollte weiterhin möglich sein, Spalten zuzuweisen. Die Frage ist jetzt nur, wie man das macht. Am Einfachsten dürfte es sein, in den ersten 3 Spalten nach den Namen zu suchen. Sollte sie dort nicht gefunden werden, sollte eine Fehlermeldung erscheinen. Zum Zuweisen der "zweiten" dazugehörigen Spalte sollte zu der gefundenen Spaltennummer 3 hinzuaddiert werden. Metadaten können somit erst ab der 7. Spalte auftauchen.

- kartesische Darstellung hier hatte ich vergessen zu erwähnen, wie dies überhaupt funktioniert (vielleicht hast du es aber auch schon alleine herausbekommen).

Da sich die 3. Komponente immer als Differenz zu den gezeigten beiden ergibt, ist diese nicht zwingend zum Darstellen erforderlich. Ausgehend von der gleichen gegebenen table-Datei muss nun nur noch angegeben werden, welche beiden Komponenten dargestellt werden sollen. Dies sollte wie schon oben beschrieben wurde möglich sein.

Das Plotten sollte dann out-of-the-box möglich sein.

pgfplotstodo.tex:1233 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1241 **Feature Proposal** [open, Priority 5]

output cs: - implement automatic limit computation ⇝ I prepared something like that; use it. I guess I'll need to convert the streamed data to the accepted format of the axis, at least in order to update limits.

pgfplotstodo.tex:1258 **Feature Proposal** [open, Priority 5]

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:1262 **Feature Proposal** [open, Priority 5]
patch visualization: provide displacement input format

pgfplotstodo.tex:1293 **Feature Proposal** [open, Priority 5]
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!

pgfplotstodo.tex:1297 **Feature Proposal** [open, Priority 5]
add polar coordinates

pgfplotstodo.tex:1303 **Feature Proposal** [open, Priority 5]
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:1320 **Feature Proposal** [open, Priority 5]
Idea: implement an automatic /pgf/number format setting which determines a suitable representation for a *set* of numbers. For example, 1e-17 0.2 0.4 0.8 should be printed as 0 0.2 0.4 0.8 whereas 1e-17 2e-17 3e-17 should be printed using the scientific range (perhaps even using some sort of scaling as for ticks). This would be useful for contour plot labels as well. $\rightsquigarrow$ a realization should check the data range (especially its exponent). Thus, I want a *relative* number printing style.

pgfplotstodo.tex:1325 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1373 **Feature Proposal** [open, Priority 5]

new structure for math operations:

- aim: interface for math operations which works independent of lowlevel repr
- ¿ FPU vs basic pgf vs LUA vs 'fp.sty' vs ....
- ¿ log axes can be done in pgf (faster)
- necessary: high level `\pgfmathparse` *and* mid level invocation of operations
- necessary: parsenumber, tofixed, tostring
- datascaling needs access to exponents and base 10 shifts
- necessary: check for nan and inf
- necessary: the max/min routines which are no longer supported by pgf (the `\pgfplotsmath...` routines)

interface:

- transparent exchange of math mode routines
- fast (enough)
- for each axis separately (optimized for log)
- variable number of arguments
- expansion of arguments should be possible
- the interface is necessary for *coordinate* arithmetics, not necessarily for the pgf interaction (can keep register math)

realization ideas:

- command suffix for each axis '@basic' versus 'float'
- central interface to invoke math ops: `\pgfplotscoordmath{x}{multiply}{{<arga>}{<argb>}}` Idea: use `\edef` on the arguments.
- provide `\pgfplotssetmathmode{x}{<suffix>}` should assert that the desired interface is complete

- `\pgfmathparse` may need to be adjusted if it uses a different output format than ¡suffix¿

TODO:

- the log routines -⇝ also use it for table package. BUGGY! compare examples in manual. Minor log ticks don't work at all, default log tick labels are simply wrong.
- disablelogfilter case
- error bars work with both, float and log
- plothandlers.code.tex
- prepare@ZERO@coords

pgfplotstodo.tex:1378 **Feature Proposal** [open, Priority 5]

rewrite the read number routines. They should allow 'disabledatafilter' thing during addplot.

pgfplotstodo.tex:1406 **Feature Proposal** [open, Priority 5]

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:1410 **Feature Proposal** [open, Priority 5]

plot expression: make the sampling parameters available within survey phase

pgfplotstodo.tex:1414 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1419 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1430 **Feature Proposal** [open, Priority 5]

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:1443 **Feature Proposal** [open, Priority 5]

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:1485 **Feature Proposal** [open, Priority 5]

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:1494 **Feature Proposal** [open, Priority 5]

add something like

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

pgfplotstodo.tex:1508 **Feature Proposal** [open, Priority 5]

I got several feature requests for non-cartesian axes. Perhaps there is a way to generalize the complete procedure... as far as I remember, I use the pointxyz routines anyway to place tick marks and so on. Perhaps it can be reconfigured to do something "advanced". Idea: nonlinear transformation into the axis combined with special drawing rotuines for the axis? ternary diagrams `http://staff.aist.go.jp/a.noda/programs/ternary/ternary-en.html`. smith charts `http://www.mathworks.com/access/helpdesk/help/toolbox/rf/f2-999699.html` `http://www.siart.de/lehre/smithdgr.pdf`

pgfplotstodo.tex:1531 **Feature Proposal** [open, Priority 5]

smith charts

`http://www.siart.de/lehre/tutorien.xhtml#smishort`
`http://www.siart.de/lehre/smithdgr.pdf`

`www.amanogawa.com/archive/docs/G-tutorial.pdf`

`http://www.mathworks.com/access/helpdesk/help/toolbox/rf/f2-999699.html`

ok, basic things work todo still:

- write support for ticklabel lines which are not axis lines (or for centered axis lines)

  $\rightarrow$ place ticklabels on the $y \equiv 0$ instead of $y_{\min}$!
- provide support for partial grid lines
- default tick positions should have nonlinear distance (?)
- fine tuning
- check anchors

pgfplotstodo.tex:1543 **Feature Proposal** [open, Priority 5]

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:1560 **Feature Proposal** [open, Priority 5]

contour:

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

pgfplotstodo.tex:1568 **Feature Proposal** [open, Priority 5]

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:1573 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1592 **Feature Proposal** [open, Priority 5]

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:1598 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1605 **Feature Proposal** [open, Priority 5]

it might be interesting to fill the area between two paths. Perhaps there is such a feature in pgf; or perhaps I can generalize the `\closedcycle` implementation written for stacked plots.

pgfplotstodo.tex:1610 **Feature Proposal** [open, Priority 5]

provide a `\numplotsperplothandler` or something like that. This would improve things for bar plots!

pgfplotstodo.tex:1614 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1625 **Feature Proposal** [open, Priority 5]

table package and axes should improve their communication. Namely:

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

pgfplotstodo.tex:1630 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1637 **Feature Proposal** [open, Priority 5]

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:1641 **Feature Proposal** [open, Priority 5]

improve support for multiple ordinates

pgfplotstodo.tex:1654 **Feature Proposal** [open, Priority 5]

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:1659 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1666 **Feature Proposal** [open, Priority 5]

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:1709 **Feature Proposal** [open, Priority 5]

optimization ideas:

- replace `\pgfpointscale` with a 'q' version $\rightsquigarrow$ 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:1714 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1718 **Feature Proposal** [open, Priority 5]

asymmetric error bars

pgfplotstodo.tex:1723 **Feature Proposal** [open, Priority 5]

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:1728 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1735 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1771 **Feature Proposal** [open, Priority 5]

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:1777 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1781 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1787 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1791 **Feature Proposal** [open, Priority 5]

interpolate missing coordinates for stacked plots.

pgfplotstodo.tex:1797 **Feature Proposal** [open, Priority 5]

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:1827 **Feature Proposal** [open, Priority 5]

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:1831 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1864 **Feature Proposal** [open, Priority 5]

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()

$\rightsquigarrow$ 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
```

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

pgfplotstodo.tex:1912 **Feature Proposal** [open, Priority 5]

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? $\rightsquigarrow$ 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:1920 **Feature Proposal** [open, Priority 5]

support `\multicolumn` for legends

pgfplotstodo.tex:1925 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1933 **Feature Proposal** [open, Priority 5]

pgfplotstable file open protocol: provide public listener interface

pgfplotstodo.tex:1937 **Feature Proposal** [open, Priority 5]

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

pgfplotstodo.tex:1961 **Feature Proposal** [open, Priority 5]

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:1097 **Feature Proposal** [open, Priority 1]

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:1548 **Feature Proposal** [cancelled, Priority 5]

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

pgfplotstodo.tex:1267 **Feature Proposal** [closed, Priority 5]

write better on-the-fly table generation support like
`\addplot table[y=create col/linear regression{x=Basis,y=L2/ref_h,xmode=log,ymode=log},]`

pgfplotstodo.tex:1271 **Feature Proposal** [closed, Priority 5]

improve access to 'create on use' things in addplot table.

pgfplotstodo.tex:1276 **Feature Proposal** [closed, Priority 5]

linear regression: at least when used inside of addplot table, the initial values of x,y,xmode,ymode should be acquired from pgfplots!

pgfplotstodo.tex:1916 **Feature Proposal** [closed, Priority 5]

external lib + makefile support: provide data files automatically as prereqs

pgfplotstodo.tex:1929 **Feature Proposal** [closed, Priority 5]

external lib + makefile support: provide data files automatically as prereqs