PGFPlots Primer

Clinton Curry

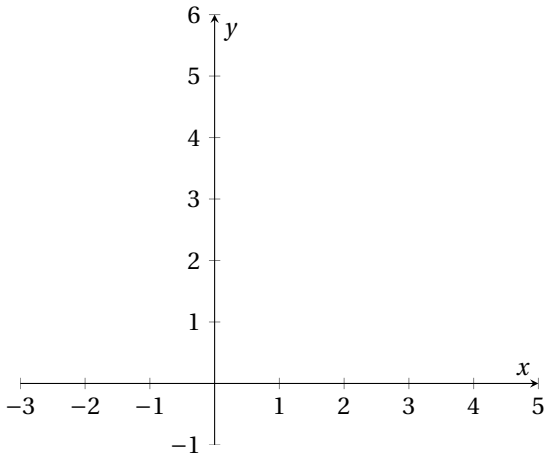
September 27, 2012

PGFPlots[1] is a horrendously intricate LaTeX package for plotting graphs, which is based on PGF/TikZ[2], another horrendously intricate package for drawing figures. These packages have an enormously powerful key/value-based system. The values indicate how the figure is to be drawn, and there are a dizzying number of keys one can configure for a graph. The degree of customization made possible is outlined in the 400-page manual[3] for the package.

This document gives two examples, to be modified for use in such documents as exams and Beamer presentations. Mix and match options as you need; for example, I would use some of the options from the first and second examples to display a graph similar to how it would be drawn by hand. For the myriad other options, refer to the manual; you can do most reasonable things.

1 Plain axes

```
1  \begin{tikzpicture}
2    \begin{axis}[
3      axis x line=middle,
4      axis y line=middle,
5      xlabel=$x$, ylabel=$y$,
6      xmin=-3, xmax=5,
7      ymin=-1, ymax=6,
8      xtick={-3,...,5},
9      ytick={-1,...,6},
10     ]
11   \end{axis}
12 \end{tikzpicture}
```



Line 3–4 As is conventional in mathematics, the x- and y-axes are where the coordinates are labeled. This is **not the default**, presumably because it tends to be a bit ugly on computer-generated plots and induces special cases. (What if the axes are not in the viewing window?) To see the default behavior, examine "A simple plot" below.

Line 5 By default, the axes are not given a label. (Default behavior is displayed in "A simple plot" below.)

Line 6–7 This gives the extents of the graph. If omitted, the x and y ranges are determined automatically.

Line 8–9 Put a tick at every integer value. Ordinarily, the number of major tick marks is determined automatically.

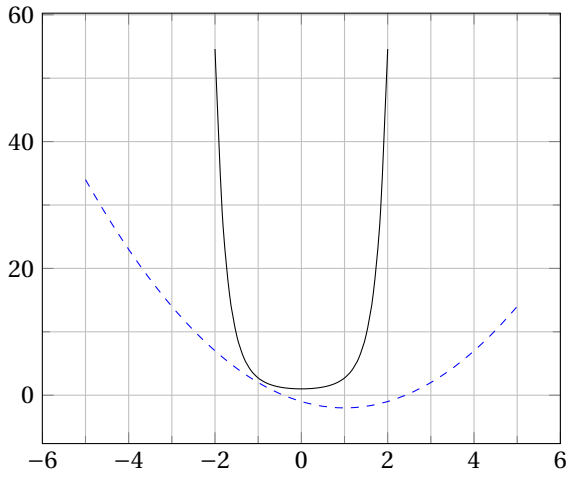
[1] http://pgfplots.sourceforge.net/

[2] http://sourceforge.net/projects/pgf/

[3] http://mirrors.ctan.org/graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf

2 A simple plot

```
1  \begin{tikzpicture}
2    \begin{axis}[
3      minor tick num=1,
4      no markers, smooth,
5      grid=both
6      ]
7      \addplot[dashed,blue]{x^2-2*x-1};
8      \addplot[domain=-2:2]{exp(x^2)};
9    \end{axis}
10 \end{tikzpicture}
```



Line 3 Between every two major ticks, there is 1 minor tick.

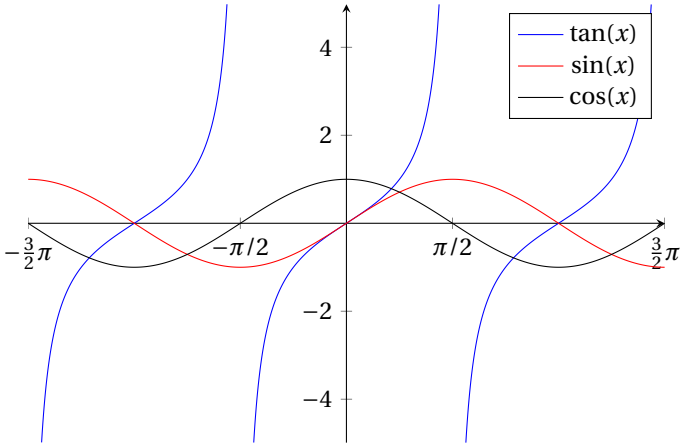
Line 4 Do not place a marker at the computed points. Also, smoothly interpolate the graph between computed points.

Line 5 Hey, let's have a grid!

Line 7 Plot the function $y = x^2 - 2x - 1$. Note the semicolon! Also, "dashed" and "blue" influence the drawing of the graph.

Line 8 Also plot the function $y = e^{x^2}$. The domain is restricted to the interval $[-2,2]$; the default domain is $[-5,5]$, but then the plot would look ridiculous, as the scales are being automatically computed.

3 A final graph



The following example is adapted from the PGFPlots manual.

```
1  \begin{tikzpicture}
2    \begin{axis}[
3      restrict y to domain=-5:5,
4      samples=1000,
5      width=10cm, height=210pt,
6      xmin=-4.7124, xmax=4.7124,
7      xtick={-4.7124,-1.5708,...,10},
8      xticklabels={$-\frac32 \pi$,$-\pi/2$,$\pi/2$,$\frac32 \pi$},
9      axis x line=center,
10     axis y line=center]
11     \addplot[blue] [domain=-1.5*pi:1.5*pi] {tan(deg(x))};
12     \addplot[red] [domain=-1.5*pi:1.5*pi] {sin(deg(x))};
13     \addplot[black] [domain=-1.5*pi:1.5*pi] {cos(deg(x))};
14     \legend{$\tan(x)$, $\sin(x)$, $\cos(x)$}
15   \end{axis}
16 \end{tikzpicture}
```

Line 3 The graph of tangent is unbounded. This option is included to prevent a TEX error.

Line 4 Rather than smooth the graph, this option make TEX compute lots of points. Looks different even from computing 100 points, and then smoothing. A drawback is that this example takes a noticeable length of time to compile.

Line 7 Illustrates regular spacing of tick marks at a distance of something other than 1.

Line 8 Give some custom labels to the tick marks.

Line 11–13 The sine, cosine, and tangent functions of PGF/TikZ expects its arguments in degrees. The deg function does the conversion.

Line 14 Illustrates the creation of a legend; name the curves in a comma-separated list.