A TikZ library for NYU calculus materials

Matthew Leingang*

Released 2017/03/02†

Contents

0 Introduction

This library grew out of a file used for `beamer` presentations for a Calculus I class. Eventually keys and styles were added for Calculus II and III. The file became the basis of a proprietary Ti*k*Z library for *Calculus for the Digital Student*.

After several years the need arose to share with the rest of the NYU Mathematics Clinical Faculty. The `beamer` parts were excised out and the Ti*k*Z parts remain.

This documentation file includes the implementation code and several examples to aid in making nice diagrams and graphs.

We use Ti*k*Z and PGFPLOTS, both built on the PGF package. This code goes into the file `tikzlibrarynyucalc.code.tex`.

```
1 ⟨*library⟩
```

1 TikZ

```
2 \usepackage{tikz}
```

We want the colors defined with the `svgnames` options of the `xcolor` package. Since Ti*k*Z already loads `xcolor`, we have to go one step down and input the color definition file.

```
3 \input{svgnam.def}
```

The following Ti*k*Z libraries are required:

`positioning` for positioning nodes relative to each other

*Email: leingang@nyu.edu

†Version: v0.0-dev

`shapes` for special node shapes

The following Ti*k*Z libraries are required to typeset the manual, but not to use the library.

`fit` for fitting nodes around nodes

Several others are useful, but not required by this library. As the manual grows, these may migrate into the section above.

`hobby` for drawing nice curves through points

`patterns` for filling with patterns (Macintosh '86 style)

`decorations.text` for decorating paths with text

`decorations.shapes` for decorating paths with shapes

`spy` for creating spy-on/spy-in nodes

`intersections` for finding intersections of paths

`arrows` for various arrow shapes

`calc` for calculations on dimensions

`matrix` for creating matrices of TikZ nodes

```
\usetikzlibrary{%
    positioning,
%    fit,
    shapes,
}
```

1.1 Graphs

These two node styles are used still. They could be someday replaced with PGFPLOTS decorations (§4.16.3 of the manual) or `\addplot` with markers and coordinates? `mark=*` and `mark=o` ought be useful here.

```
\tikzstyle{point}=[
    fill,fill opacity=1,
    circle,
    minimum size=3pt,inner sep=0pt,outer sep=0pt]
\tikzstyle{open point}=[draw,fill=white,circle,minimum size=3pt,inner sep=0pt,outer sep=0pt]
```

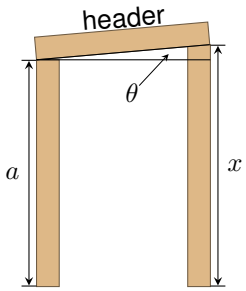
1.2 Layers

With PGF layers we can draw later objects under earlier objects.

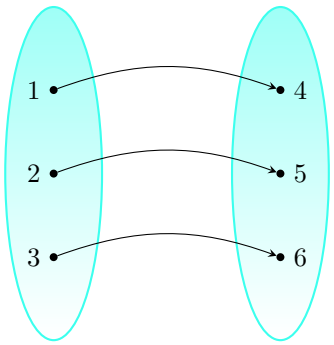
```
\pgfdeclarelayer{background}
\pgfdeclarelayer{foreground}
\pgfsetlayers{background,main,foreground}
```

1.3 Miscellaneous styles

April 14, 2012: Nancy said the `measurement` style is easier to see when it's black.




```
\tikzstyle{measurement}=[arrows={|<->|},Black]
\tikzstyle{angle measurement}=[measurement,arrows={->}]
\tikzstyle{behind}=[black!20]
\tikzstyle{imaginary}=[dashed]
\tikzstyle{set}=[draw=primary,thick,ellipse,shade,top color=primary!50,bottom color=white,inner sep=4mm]
\tikzstyle{chain}=[%
    decoration={%
        shape backgrounds,
        shape=ellipse,
        shape width=5pt,
        shape height=2.5pt,
        shape sep=4pt},
    decorate
]
```



2 Color Schemes

`\definecolor` and `\colorlet` are provided by the `xcolor` package.

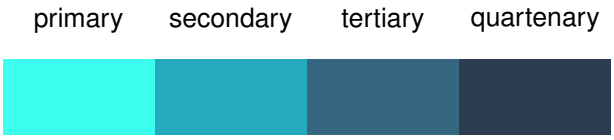
This color scheme is "Raygun Gothic", which I made on Adobe's Kuler application.

```
\definecolor{primary}  {HTML}{3CFFEE}
\definecolor{secondary}{HTML}{24AABC}
\definecolor{tertiary}{HTML}{356781}
\definecolor{quartenary}{HTML}{2C3D51}
\definecolor{quintenary}{HTML}{1C1F24}
```

Here is a blue/green motif:

```
%:TODO make this a key
%\colorlet{primary}{blue!40!black}       % HTML {000066}
%\colorlet{secondary}{blue!20!white}     % HTML {CCCCFF}
```

```
%\colorlet{tertiary}{Green!40!Black}     % HTML {006600}
%\colorlet{quartenary}{Green!20!white}   % HTML {CCFFCC}
%\definecolor{quintenary}{HTML}{1C1F24}
\tikzstyle{curve}=[primary,thick]
\tikzstyle{curve label}=[black,opacity=1]
\tikzstyle{curve direction}=[curve,fill opacity=1]
\tikzstyle{region}=[curve,fill=quartenary]
\tikzstyle{surface}=[fill=quartenary,opacity=0.5]
\tikzstyle{function}=[curve]
```



3 PGFPLOTS

PGFPLOTS is a massive PGF/Ti*k*Z application that does graphs. Very cool.

```
\RequirePackage{pgfplots}
\pgfplotsset{compat=1.10}
\usepgfplotslibrary{groupplots}
\usepgfplotslibrary{fillbetween}
```

We set every plot to draw axes through the origin (The default is a box around the plot with ticks on the edges.) Also the "cycle list" of colors and line styles.

```
\pgfplotsset{
    every linear axis/.append style={%
        axis x line=middle, axis y line=middle,%
        cycle multi list={%
            solid,dashed,dotted\nextlist
            {function,primary},
            {function,secondary},
            {function,tertiary},
            {function,quartenary}
        },
    },
    every axis plot/.append style={every label/.append style={black}}
}
```

This is for marking points on plots. It's a Ti*k*Z key used by PGFPLOTS.

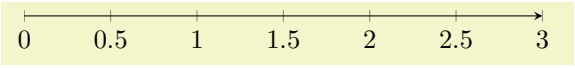
```
\tikzstyle{every mark}=[mark=*,mark size=1pt]
```

3.1 Number lines

We can use pgfplots to make number lines, too. The `number line` key does that. Thanks to TEX.SE user egreg for helping me out here. http://tex.stackexchange.com/q/184191/1402

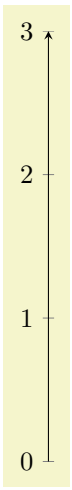
```
\pgfplotsset{
    number line/.style={%
        axis x line=bottom,
        ymin=0,ymax=0.1,
        axis equal image,
```

```
        hide y axis,
        every axis x label/.append style={%
            anchor=west,
            align=left,
        },
        execute at end axis={
            \addplot[draw=none,forget plot] {0};
        }
    }
}
```



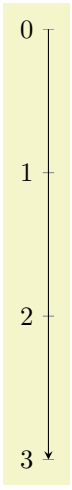

```
% \begin{tikzpicture}
%    \begin{axis}[
%       number line,
%       xmin=0,xmax=3,
%   ]
%    \end{axis}
% \end{tikzpicture}
%
```

```
\pgfplotsset{%
    number line y/.style={%
        axis y line=left,
        axis equal image,
        xmin=0,xmax=0.1,
        hide x axis,
        every axis y label/.append style={
            anchor=south east,
            align=left
        },
%        execute at end axis={
%            \addplot[draw=none,forget plot] {0};
%        }
    }
}
```




```
% \begin{tikzpicture}
%    \begin{axis}[
%       number line y,
%       ymin=0,ymax=3,
%   ]
%    \end{axis}
% \end{tikzpicture}
%
```

Use `y number line` with `scale=-1` if you want the number line to increase downwards.




```
% \begin{tikzpicture}
%    \begin{axis}[
%       number line y,
%       ymin=0,ymax=3,
%       yscale=-1
%   ]
%    \end{axis}
% \end{tikzpicture}
%
```

`interval labels` are for increasing/decreasing, concave up/down, etc.

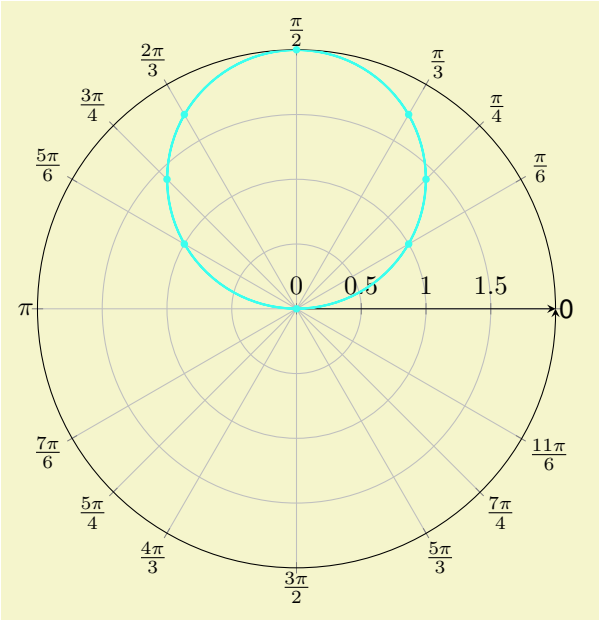
```
\pgfplotsset{%
    interval labels/.style={
        after end axis/.prefix code={
            \foreach \x/\val/\desc in {#1} {
                \ifx\val\desc
                    \edef\temp{\noexpand\draw ({axis cs:\x,0}|-{xticklabel* cs:0})
                        node[number line value] {\unexpanded\expandafter{\val}};}%
                \else
                    \edef\temp{\noexpand\draw ({axis cs:\x,0}|-{xticklabel* cs:0})
                    node[number line value] {\unexpanded\expandafter{\val}}
                    node[number line description] {\unexpanded\expandafter{\desc}};}
                \fi
                \temp
```

```
            }
        }
    },
```

Point labels are for max/min, inflection point, etc.

```
    point labels/.style={
        after end axis/.prefix code={
            \foreach \x/\val/\desc in {#1} {
                \ifx\val\desc
                    \edef\temp{\noexpand\draw ({axis cs:\x,0} |- {xticklabel* cs:0})
                        node[number line value] {\unexpanded\expandafter{\val}};}%
                \else
                    \edef\temp{%
                        \noexpand\draw ({axis cs:\x,0} |- {xticklabel* cs:0})
                            node[number line value] {\unexpanded\expandafter{\val}};
                        \noexpand\draw ({axis cs:\x,0} |- {xticklabel cs:0})
                        node[number line description] {\unexpanded\expandafter{\desc\strut}};
                        }%
                \fi
                \temp
            }
        }
    }
}
```

3.2 Polar plots

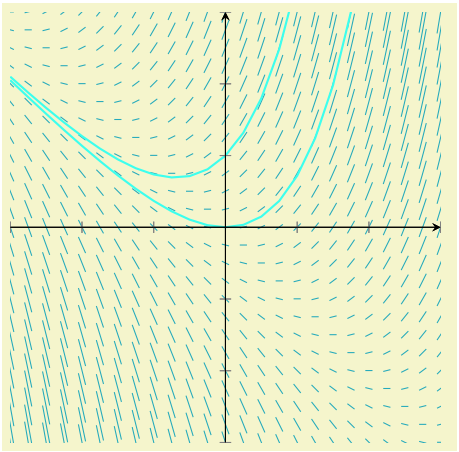


```
%    \begin{tikzpicture}
%    \tikzstyle{every plot}=[] % usually smooth, but that gets in the way of only marks below
%        \begin{polaraxis}[
%            axis x line=right,
%            xtick={0,30,45,60,90,120,135,150,180,210,225,240,270,300,315,330},
%            xticklabels={
%                0,$\frac{\pi}{6}$,$\frac{\pi}{4}$,$\frac{\pi}{3}$,$\frac{\pi}{2}$,$\frac{2\pi}{3}$,$\frac{3\pi}{4}$,$
%                $\pi$,$\frac{7\pi}{6}$,$\frac{5\pi}{4}$,$\frac{4\pi}{3}$,$\frac{3\pi}{2}$,$\frac{5\pi}{3}$,$\frac{7\p
%            },
%            ymin=0,ymax=2,
%        ]
%            \addplot+[mark=none,data cs=polarrad,domain=0:6.28,forget plot,samples=100] {2*sin(deg(x))};
%            \addplot+[only marks,data cs=polarrad,mark=*] coordinates{
%                (0.5236,1.)
%                (0.7854,1.4142)
%                (1.0472,1.7321)
%                (1.5708,2.)
%                (2.0944,1.7321)
%                (2.3562,1.4142)
%                (2.618,1.)
%                (3.1416,0.)
%            };
%        \end{polaraxis}
%    \end{tikzpicture}
%
```

```
131 \usepgfplotslibrary{polar}
```

3.3 Slope fields

Slope fields are implemented as PGFPLOTS “quivers”—that is, vector fields. But since no arrow style is set the arrows are headless. We use a coordinate filter to shift the line segments so that the base point is at the *center* of the segment instead of the tail.

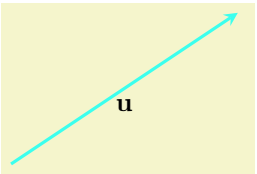


```
%    \begin{tikzpicture}[
%        declare function={
%            F(\y,\t) = \t+\y;
%            phi(\t,\yo) = (\yo+1)*exp(\t) - \t - 1;
%        }
%    ]
%        \begin{axis}[
%            xmin=-3,xmax=3,
%            ymin=-3,ymax=3,
%            axis equal image,
%            view={0}{90},
%            xtick={-3,-2,...,3},
%            ytick={-3,-2,...,3},
%        ]
%            \addplot3+[
%                slope field={F(x,y)},
%                domain=-3:3,
%                domain y=-3:3,
%            ]{x};
%            \addplot+[ode solution,variable=\t,domain=-3:3] {phi(t,1)};
%            \addplot+[ode solution,variable=\t,domain=-3:3] {phi(t,0)};
%        \end{axis}
%    \end{tikzpicture}
%
```

```
\pgfplotsset{
slope field/.style={
    secondary,thin,
    /pgfplots/quiver={u=1,v={#1},scale arrows=0.1},
    /pgfplots/x filter/.expression={x-1*0.5*\pgfkeysvalueof{/pgfplots/quiver/scale arrows}},
    /pgfplots/y filter/.expression={y-(#1)*.5*\pgfkeysvalueof{/pgfplots/quiver/scale arrows}}
    },
    ode solution/.style={primary},
}
```

4 Vectors

`vector` The `vector` key is for vectors.




```
%    \begin{tikzpicture}
%        \draw[vector] (0,0) --++ (3,2) node[vector label,below] {$\mathbf{u}$};
%    \end{tikzpicture}
%
```

```
\tikzset{>=stealth}
\tikzstyle{vector}=[primary,very thick,arrows={->}]
```

`vector label`

```
\tikzstyle{vector label}=[pos=0.5,black]
```

This is the end of `tikzlibrarynyucalc.code.tex`.

⟨/library⟩