TikZ tips and tricks

Nathan M. Dunfield

This document assumes that `nmd/graphics` has been loaded with the `tikz` option enabled. The `nmdstd` style includes my preferred line styles and arrows.

```
\begin{tikzpicture}[line width=3pt]
  \draw[->] (0,0) -- (1, 1) -- (2, 0);
  \begin{scope}[nmdstd, xshift=3cm]
    \draw[->] (0,0) -- (1, 1) -- (2, 0);
  \end{scope}
\end{tikzpicture}
```

Note that `nmd/graphics` also loads the following TikZ libraries:

```
\usetikzlibrary{calc}
\usetikzlibrary{positioning}
\usetikzlibrary{intersections}
\usetikzlibrary{arrows.meta}
\usetikzlibrary{decorations.markings}
\usetikzlibrary{decorations.pathreplacing}
```

1 Recommended workflow

(a) On paper, draw sketches and the finalize overall design as much as possible. The more that is fixed now, the easier this will be.

(b) Scan the final drawing and crop in e.g. Preview. Go to the master document and do something like the following to determine the rough size of the figure.

```
\begin{figure}
  \begin{center}
    \includegraphics[width=4.5cm]{scan}
  \end{center}
  \caption{A caption}
  \label{fig:test}
\end{figure}
```

(c) Create a file `plots/afig.tex` where the TikZ code will live containing

```
\begin{tikzpicture}[nmdstd]
  \node[above right, opacity=0.3]
        at (0, 0) {\includegraphics[width=4.5cm]{scan}};
  \measurementgrid{5}{5};
\end{tikzpicture}
```

and setup a minimal document to use whilst programming the figure

```
\documentclass[margin=0.5cm]{standalone}
\usepackage{nmd/math}
\usepackage[tikz]{nmd/graphics}
\begin{document}
  \input plots/afig.tex
\end{document}
```

(d) Now hack away at `afig.tex` while compiling the minimal document. A good approach is first to define `\coordinates` for key points using the grid and plot them

```
  \coordinate (A) at (0.8, 0.4);
  \coordinate (B) at (4.5, 2.5);
  \coordinate (C) at (0.6, 4.6);

  \foreach \p in {A,B,C}{
    \node[opacity=0.2] at (\p) {\p};
  }
```

Now use convex combinations and barycentric coordinates to define even more points

```
    \coordinate (D) at ($(A)!0.25!(B)$);
    \coordinate (center) at (barycentric cs:A=0.5,B=0.5,C=0.5);
```

2 Drawing Bézier curves efficiently

Visualize the control points via the `show controls` style and specify them relative to the endpoints using polar coordinates as in the second example.

```
\begin{tikzpicture}[nmdstd, line width=1pt]
  \draw[show controls] (0, 0) .. controls (1, 1) and (2, -1) .. (4, 0);
  \draw[show controls] (7, 0) .. controls +(45:1) and +(-150:2) .. (10, 0);
\end{tikzpicture}
```

There is a whole package `spath` for working with paths as objects, but overall it seems like the best approach if you want to reuse a path is just to declare a macro.

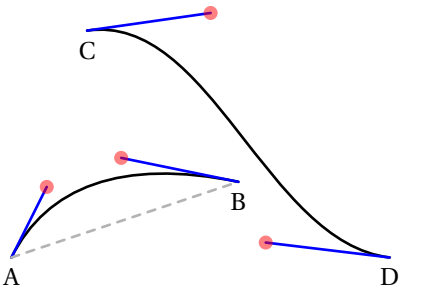
```
\begin{tikzpicture}
  \def\somepath{(0, 0) .. controls (1, 1) and (2, -1) .. (4, 0)}
  \draw[line width=2pt, color=blue] \somepath;
  \begin{scope}[shift={(5, 0)}]
    \draw[line width=1pt, color=red] \somepath;
  \end{scope}
\end{tikzpicture}
```

Even better than in the previous examples, you can use my `easybezier` macro to specify the control points using polar coordinates oriented with respect to the line between the end points so that the distance between the endpoints is scaled to be one unit.

```
\begin{tikzpicture}[nmdstd, line width=1pt]
  \coordinate (A) at (0, 0);
  \coordinate (B) at (3, 1);
  \draw[show controls]
        \easybezier{(A)}{(B)}{45}{0.33}{-30}{0.5};
  \draw[dashed, color=black!30] (A) -- (B);

  \coordinate (C) at (1, 3);
  \coordinate (D) at (5, 0);
  \draw[show controls]
       \easybezier{(C)}{(D)}{45}{0.33}{30}{0.33};

  \foreach \p in {A,B,C,D}
    {\node[below] at (\p) {\p};}
\end{tikzpicture}
```



3 Doing things along paths

Often we want to do something along a path, like add an arrow or a coordinate or add a tangent or normal vector. This is done with the decorations/markings functionality of TikZ, which is a bit verbose, so I made shortcuts for some of these.

```
\begin{tikzpicture}[line width=1pt]
  \draw[mid arrow=0.1,
        mid arrow=0.2,
        add coordinate={0.5}{A},
        add coordinate={0.9}{B},
        tangent vector={0.7}{2cm},
        normal vector={0.7}{-2cm},
        vector along={0.7}{(-45:2)},
        ]
     \easybezier{(0, 0)}{(10, 0)}{70}{0.2}{50}{0.5};

  \node[below] at (A) {A};
  \node[below] at (B) {B};
  \draw[dashed] (A) -- (B);
  \fill ($(A)!0.5!(B)$) circle (2pt);
\end{tikzpicture}
```