The calligraphy Package: Documentation

Andrew Stacey
loopspace@mathforge.org

v2.4 from 2021/02/21

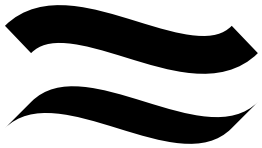
1 Pre-Introduction

This library is built on top of a package for manipulating PGF's *soft paths* called `spath3`. Version 2.0 of `spath3` involved considerable reorganisation of the code. I tried to ensure that this didn't affect this library but it is extremely likely that I wasn't fully successful. If something that used to work no longer does, please do let me know either by opening an issue on github (https://github.com/loopspace/spath3) or at the above email.

2 Introduction

The `calligraphy` TikZ library is designed to enable calligraphic style drawings in TikZ. The idea is to be able to "stroke" a line with a "pen". As a simple example, compare the two lines in the following picture.

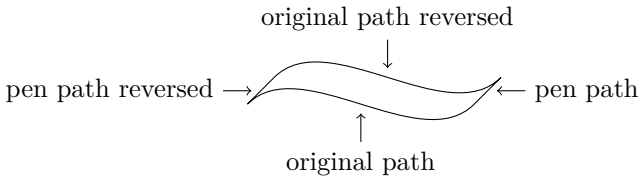
```
\begin{center}
\begin{tikzpicture}
\pen (-135:.25) -- (45:.25);
\draw[line width=.5cm] (0,0) .. controls +(45:1) and +(-135:1) ..
    ++(3,0);
\calligraphy (0,-1) .. controls +(45:1) and +(-135:1) .. ++(3,0);
\end{tikzpicture}
\end{center}
```



The paths are identical in definition but the first is drawn using the standard TikZ path with a line width of .5cm. The second is "stroked" with a calligraphic pen of width .5cm angled at 45 degrees.

3 How It Works

To know how to use this library, it is worth knowing a little about how it works. A "pen" is a path, as is the line that is the template for the pen stroke. The two paths are joined together to form a region which is filled. Thus in constructing the example given in the introduction, the following path is built.



What is important to note about this is that the "pen" isn't *actually* dragged along the path, it is merely a simulation. This can be shown with the following simple example. The first is a continuous path that goes past the angle of the pen and thus the upstroke would involve pushing the pen. The second is how it is meant to be done, the second line is drawn from top to bottom. However, as the direction of the path isn't important, the same effect can be obtained by "lifting the nib" between the lines.

```
\begin{center}
\begin{tikzpicture}
\pen (-135:.125) -- (45:.125);
\calligraphy (0,0) -- (1,0) -- (1,1);
\calligraphy (2,0) -- (3,0) (3,1) -- (3,0);
\calligraphy (4,0) -- (5,0) +(0,0) -- (5,1);
\end{tikzpicture}
\end{center}
```

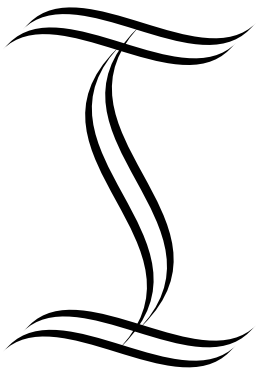
It should work as expected providing the following golden rule is not violated:

> Never *push* a calligraphic pen.

This is good advice for ordinary calligraphy as well, so a path that is realisable as an honest calligraphic path should be fine with this library. Actually since, as remarked above, the direction of the path isn't important, a more accurate golden rule would be that one should never swap from pushing to pulling or vice versa

without lifting the pen off the paper; but that isn't as succinct. The paths for both pens and templates can be reasonably complicated. They can contain gaps, but should not contain closed paths, nor rectangles. The implementation works by breaking a path into its constituent pieces (broken up by "move to"s) and working bit by bit.

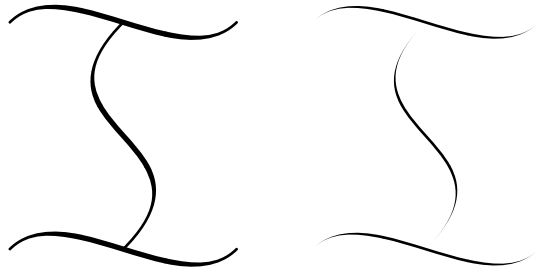
```
\begin{center}
\begin{tikzpicture}
\pen (-135:.25) -- (-135:.125) (45:.125) -- (45:.25);
\calligraphy (0,0) .. controls +(45:1) and +(-135:1) .. +(3,0)
    ++(1.5,0) .. controls +(-135:2) and +(45:2) .. +(0,-4)  (0,-4)
    .. controls +(45:1) and +(-135:1) .. +(3,0);
\end{tikzpicture}
\end{center}
```



4 Copperplate

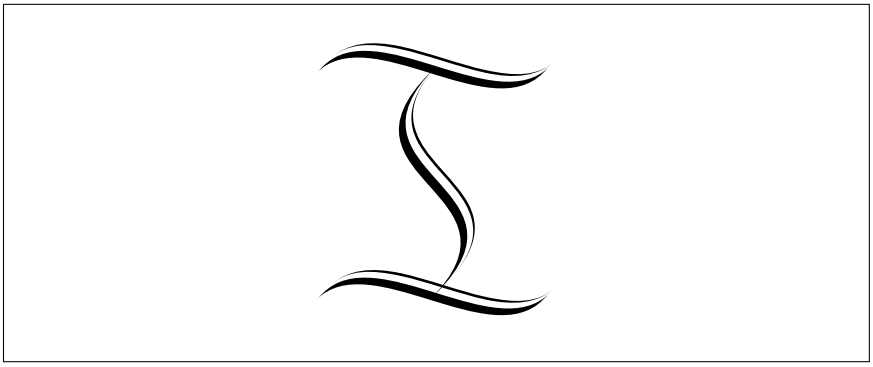
Copperplate pens are somewhat special. They are "thin" so don't need the same treatment as a "thick" pen, but one should be able to vary the pressure with a copperplate pen to get a variation of thickness. Specifying a copperplate pen is straightforward: it is a pen with no thickness.

```
\begin{center}
\begin{tikzpicture}[line width=2pt]
\pen (0,0);
\calligraphy[heavy] (0,0) .. controls +(45:1) and +(-135:1) ..
    +(3,0) ++(1.5,0) .. controls +(-135:2) and +(45:2) .. +(0,-3)
    (0,-3) .. controls +(45:1) and +(-135:1) .. +(3,0);
\calligraphy[light] (4,0) .. controls +(45:1) and +(-135:1) ..
    +(3,0) ++(1.5,0) .. controls +(-135:2) and +(45:2) .. +(0,-3)
    (4,-3) .. controls +(45:1) and +(-135:1) .. +(3,0);
\end{tikzpicture}
\end{center}
```



With a copperplate pen, the segments of a path are tapered. Copperplate and normal pens can be mixed. Any part of the pen specification that has no length is treated as a copperplate pen.

```
\begin{center}
\begin{tikzpicture}[line width=1pt]
\pen (-135:.125) -- (0,0) (45:.125);
\calligraphy (0,0) .. controls +(45:1) and +(-135:1) .. +(3,0)
    ++(1.5,0) .. controls +(-135:2) and +(45:2) .. +(0,-3)  (0,-3)
    .. controls +(45:1) and +(-135:1) .. +(3,0);
\end{tikzpicture}
\end{center}
```



5 Style Options

There are plenty of options for styling the paths and pens.

5.1 Definition Options

Internally, making a pen is a two-step process. First a pen has to be *defined* and then *processed*. To define a pen, the user has to specify a path. That path is stored in a global macro and so can be accessed in throughout the document. However, before being used, the pen has to be processed. At this stage, the pen is converted from a macro in to a special object. These special objects are local and cannot (at present) be made global. Thus whilst a pen can be *defined* inside a group, the *processing* stage has to happen in the outermost group in which the pen is going to be used. There is a shortcut command that (via a bit of suspicious hackery) does all this within a `tikzpicture` group. However, if a pen is to be used in several different pictures, it must be processed outside the group in which it is defined. The following macros and keys are used to set up and use a pen.

`define pen`

- If the `define pen` key is specified on a path then that path will be used to define a pen. It can take one option which will be the pen name, if not specified then `default` is assumed. The resulting path will not be counted for bounding box considerations. When the pen is used, the origin will correspond to the path along which it is dragged.

`pen name`

- The key `pen name=name` sets the name for the current pen. This can be used either when defining or using a pen.

`\pen` `\definepen`

- The macros `\pen` and `\definepen` are analogous to the TikZ commands `\draw` or `\fill` in that they act like a path command but store the path as a pen. The difference between them is that `\definepen` is to be used outside a TikZ picture (it contains its own `\tikz` command) and `\pen` inside.

`use pen`

- The key `use pen=name` on a path means that that path should be "stroked" with the pen (`default` if no name is given, or none specified via the `pen name` key).

5.2 Style Options

There are various options available for styling the calligraphic paths. The style options are as follows.

`pen colour`

- The `pen colour` style defines the default colour to be used. Since calligraphic paths are sometimes filled and sometimes stroked, this ensures that the colour is used correctly.

`nib style`

- It is possible to style particular "nibs" (i.e., segments of the pen path) separately. This is the `nib style` option, which takes two arguments. The first is the index of the part of the nib and the second is the style options to be applied.

`stroke style`

- It is also possible to style particular parts of the template path. One way to do this is to use the `stroke style` key, which takes two arguments. The first is the index of the part of the stroke and the second is the style options to be applied.

`this stroke style`

- It is also possible to style particular parts of the template path as the path is constructed. This is done by putting `[this stroke style={}]` in the template path at the relevant part. The style is saved and applied to that segment of the template path.

`taper`

- The tapering of copperplate paths can be controled by the `taper` option. It takes arguments `none`, `both`, `start`, and `end`.

`weight` `heavy` `light`

- Copperplate paths come in two "weights": *heavy* and *light*. The weight also affects the tapering: by default a light path is tapered to nothing whilst a heavy path is tapered to the width of a light path. Weights can be specified by either `weight=weight` or just `heavy` and `light`. It is possible to change the weight for different components of a path using the `stroke style` key. With tapering, this means that one can easily vary from a light stroke to a heavy one. The relevant widths are controlled by the keys `heavy line width` and the `light line width`. The `taper line width`, is set automatically by the weight but can be altered afterwards using the `taper line width` key.

`heavy line width` `light line width` `taper line width`

```
\begin{center}
\begin{tikzpicture}
\calligraphy[pen colour=green,nib style={2}{color=red}] (0,0) ..
    controls +(45:1) and +(-135:1) .. +(3,0) ++(1.5,0) .. controls
    +(-135:2) and +(45:2) .. +(0,-3)  (0,-3) .. controls +(45:1)
    and +(-135:1) .. +(3,0);
\calligraphy[line width=1pt] (0,-4) .. controls +(45:1) and
    +(-135:1) .. +(3,0) ++(1.5,0) .. controls +(-135:2) and
    +(45:2) .. +(0,-3)  (0,-7) .. controls +(45:1) and +(-135:1)
    .. +(3,0);
\end{tikzpicture}
\end{center}
```

6 Decorations

If a TikZ/PGF decorations library is loaded prior to this library, then the `calligraphy` library defines some decorations that use the calligraphic paths, specifically with the copperplate nib. The current decorations are:

`calligraphic brace`

- `calligraphic brace` for a brace.

`calligraphic straight parenthesis`

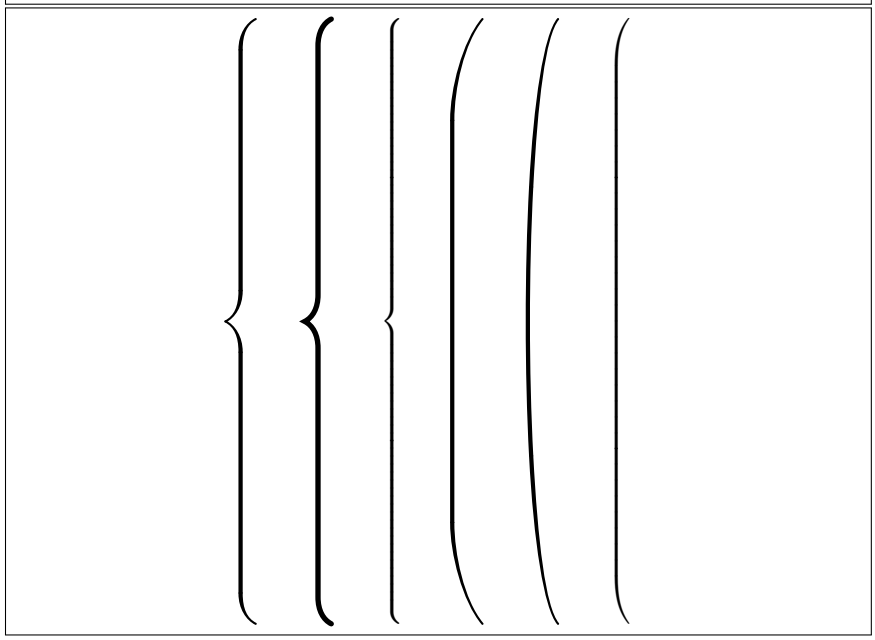
- `calligraphic straight parenthesis` for a parenthesis with straight middle component.

`calligraphic curved parenthesis`

- `calligraphic curved parenthesis` for a parenthesis with a curved middle component.

All the above use the `amplitude` option to specify their size. The following is an example of their use, together with the standard `brace` and the `delimiter` key from the `matrix` library for comparison.

```
\begin{center}
\begin{tikzpicture}
\draw[decorate,decoration={calligraphic brace,amplitude=4mm},ultra
    thick] (0,0) -- (0,8);
\draw[line width=2pt,decorate,decoration={brace,amplitude=10},line
    cap=round] (1,0) -- ++(0,8);
\node[anchor=south west,minimum height=8cm,outer sep=0pt,left
    delimiter=\{] (a) at (2,0) {};
\draw[decorate,decoration={calligraphic straight
    parenthesis,amplitude=4mm},ultra thick] (3,0) -- ++(0,8);
\draw[decorate,decoration={calligraphic curved
    parenthesis,amplitude=4mm},ultra thick] (4,0) -- ++(0,8);
\node[anchor=south west,minimum height=8cm,outer sep=0pt,left
    delimiter=(] (a) at (5,0) {};
\end{tikzpicture}
\end{center}
```



7 Pre-Defined Pens

The following pens are predefined:

`copperplate`

- `copperplate`:

```
\begin{center}
\tikz \calligraphy[copperplate] (0,0) .. controls +(1,-1) and
    +(-1,1) .. ++(3,0) [this stroke
    style={light,taper=start}] +(0,0) .. controls +(1,-1) and
    +(-1,1) .. ++(3,0) [this stroke style={heavy}] +(0,0) ..
    controls +(1,-1) and +(-1,1) .. ++(3,0) [this stroke
    style={light,taper=end}];
\end{center}
```