The celtic TikZ Library: Documentation

Andrew Stacey
stacey@math.ntnu.no

19th February 2016

1 Introduction

This is a little TikZ library for drawing Celtic style knots. The particular type of Celtic knot (technically, *link*) is very simple and can be specified by listing the "walls" within the region of the knot. From this information, it is possible to build the entire link and thus to tell TikZ how render it. That is what this library does.

2 Usage

The routine is implemented as a TikZ library. Thus to use it, add `celtic` to the list of TikZ libraries that you load.

```
\usetikzlibrary{celtic}
```

The library defines one command which renders a Celtic knot. The knot is specified by passing various *key-value* pairs to this command. The library also defines styles which can be used to modify the rendering.

`\CelticDrawPath` `\CelticDrawPath{<opt>}` is the command to render a knot. It takes one option which is a list of key-value pairs which specify the knot. The allowed key-value pairs are as follows.

- `max steps=N` The process of finding the paths through the knot (needed to ensure that they are coloured correctly) is iterative. Although every care has

been taken to ensure that the iteration is confined (and therefore finite), the iteration has been devised with a built-in limit. This limit can be adjusted using this key. The default is 20. If the limit is reached, a warning will be issued (and the knot will probably look wrong). In that case, use this key to raise the limit.

- `flip` The specification of a Celtic knot in terms of walls does not completely determine it. There is an ambiguity as to which crossings are over and which under (once one crossing is determined, all the others follow). This key flips all of the crossings and so can be used to switch between the two variants.

- `width=W`, `height=H`, `size={W,H}` These set the dimensions on the knot in terms of the number of crossings. The numbers must be even.

- `crossings`, `symmetric crossings` These set the crossings. The general format of a crossing is `x,y,type` where `x` and `y` can be either numbers or ranges, using the format `n:m`[1]. The type of the crossing is either `|` or `-` for (respectively) vertical or horizontal walls[2]. Multiple crossing specifications can be given as a semi-colon-delimited list (a final semi-colon is acceptable, making it easy to comment out items in the list). The `symmetric` variant places walls at four points obtained by applying reflections to the specified crossing.

- `ignore crossings`, `ignore symmetric crossings` The code works out the paths involved by picking a starting point and direction and then following it, bouncing off walls as appropriate, until it comes back to the beginning. It then picks a new starting point and continues until all crossings are used up. These keys designate certain points as *disallowed* as starting points. This can be used to remove certain regions from the knot, for example to create a border around a rectangle. The `symmetric` version works ... symmetrically.

- `style={<style>}` The contents of this are passed on to `\tikzset`. (The key-value pairs for the `\CelticDrawPath` are actually LaTeX3 keys, not TikZ keys, so this is the simplest way to pass them through.)

- `at=<coordinate>` This shifts the knot so that the lower left corner is at the coordinate. (The default location is `(0,0)`.) The coordinate is passed through TikZ's processing so can be any legal TikZ coordinate.

- `inner clip=N`, `outer clip=N`. The crossings are rendered by redrawing the over paths with a clipping region. The size of the clipping region is determined by the line width. These keys add a little to that clipping region. This can be useful if the overdraws are not large enough, but the added amount should not be so much that the overdraws interfere with each other. Some experimentation is needed. The `inner clip` applies to the background part of the path and the `outer clip` to the foreground.

[1] This package uses LaTeX3 internally; using a colon as the range separator was a headache to implement.

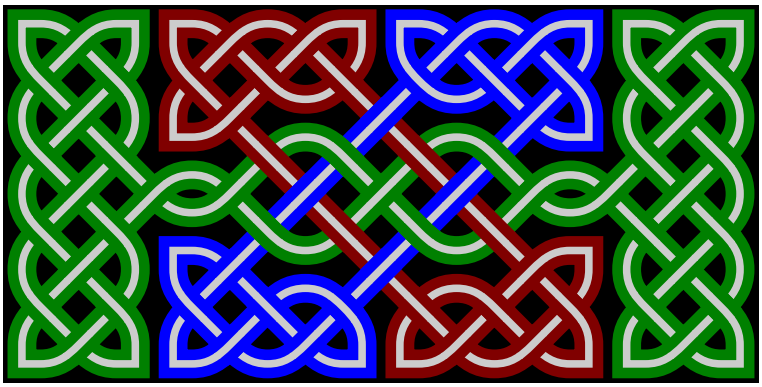
[2] The package attempts to be smart with regard to allowing `|` to be active.

As the pieces are rendered, various TikZ styles are invoked.

- `celtic surround` This style is used for the outer border.
- `celtic bar` This is used for the internal walls. It is also used on the outer border (prior to the `celtic surround` so it can be overridden).
- `celtic path` This is used for the components of the celtic knot (link).
- `celtic path N` The individual components can be styled using their number (there is a logic to the numbering, but experimentation is the best way to work out which is which).

3 Example

```
\begin{tikzpicture}[
  scale=.5,
  celtic path/.style={
    draw,
    double=gray!40,
    red,
    double distance=1mm,
    line width=4pt
  },
  celtic path 1/.style={
    green!50!black,
  },
  celtic path 2/.style={
    blue,
  },
  celtic path 3/.style={
    red!50!black,
  },
  celtic surround/.style={
    ultra thick,
    black,
    fill=black
  },
]
\CelticDrawPath{
  symmetric crossings={
    4,1:3,|;
    10,1,|;
    5,4,-;
    8,3,-;
  },
  size={20,10},
  max steps=50
}
\end{tikzpicture}
```



```
\begin{tikzpicture}[
  scale=.5,
  celtic path/.style={
    draw,
    double=white,
    red,
    double distance=5pt,
    line width=1pt
  },
  celtic bar/.style={
    ultra thick,
    black,
    draw
  },
]
\CelticDrawPath{
  size={20,10},
  symmetric crossings={
    3,4:5,|;
    4:16,3,-
  },
  ignore symmetric crossings={
    4:10,5;
    5:10,4
  },
  max steps=50
}
\end{tikzpicture}
```