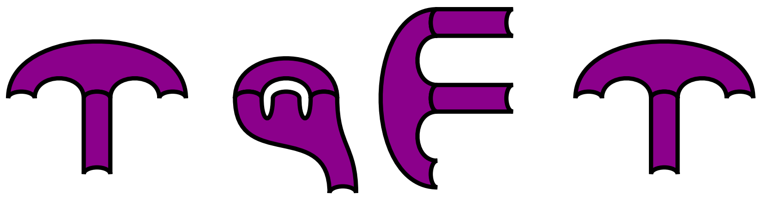
# The tqft Package: Documentation

Andrew Stacey
stacey@math.ntnu.no

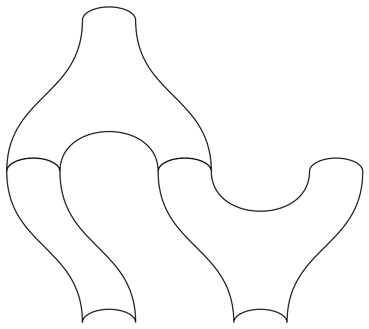
v1.0 from 2011/05/03



## 1 Introduction

This package defines some TikZ/PGF node shapes that can be used to construct the diagrams common in Topological Quantum Field Theory (TQFT). An example follows:

```
\begin{tikzpicture}
\node[draw,tqft/pair of pants] (a) {};
\node[draw,tqft/cylinder to next,anchor=incoming
   boundary 1] (c) at (a.outgoing boundary 1) {};
\node[draw,tqft/reverse pair of pants, anchor=incoming
   boundary 1] at (a.outgoing boundary 2) (b) {};
\end{tikzpicture}
```

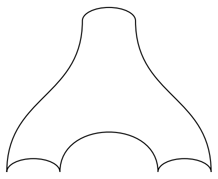


Before giving any details, a word is in order about the keys involved in this package. There are many options and keys that can be set via the `\pgfkeys` system (which is used for setting options in Ti*k*Z). Such keys live in a "directory" but often that can be omitted. For example, in the Ti*k*Z command `\draw[red] (0,0) -- (1,0);` the key `red` is actually in the "directory" `/tikz` but it is not necessary to specify that as it is assumed. Defining a "directory" helps separate keys and ensure that there is no conflict. The keys in this package are (mostly) defined in the directory `/pgf/tqft` but the very first call to a `tqft` key will (in general) set the "current directory" to `/pgf/tqft` and so all subsequent keys do not need prefixing. Moreover, any unknown keys are passed on to the `/tikz` directory so there is (or should be!) no harm in mixing `tqft` specific keys with ordinary Ti*k*Z keys. Some examples take advantage of this switch so when copying and modifying examples from this document, it is important to remember that the first `tqft` specific key needs an explicit `tqft/` prefix. More detailed information is in the section on styling.

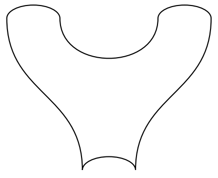
## 2 The Shapes

There are only two shapes, `tqft cobordism` and `tqft boundary circle`. The first, which is the main shape, is a cobordism between a number of incoming circles and a number of outgoing circles, where the numbers of boundary components can be specified as options to the shape. The second is just the boundary circle. It is used as a sub-node of the first to add extra anchors, but can be used by itself. There are certain common shapes that are predefined as aliases to the main shape with specified boundaries. The list of predefined shapes follows. The names are all in the `tqft` family, but an alias is made so that `tqft nodeshape` will work without any further qualification.

1. `pair of pants`

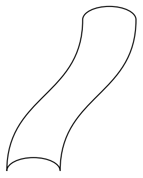


2. `reverse pair of pants`



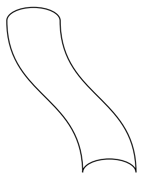
3. `cylinder to prior`

This is a cylinder that has been skewed to one side, thus following the same path as the `pair of pants` cobordism but with only one outgoing boundary component. The name `to prior` is because it goes towards the lower-numbered component on the `pair of pants`.



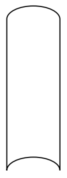
4. `cylinder to next`

   This is a cylinder that has been skewed to one side, thus following the same path as the `pair of pants` cobordism but with only one outgoing boundary component. The name `to next` is because it goes towards the higher-numbered component on the `pair of pants`.

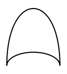
5. `cylinder`

   This is a straight cylinder.

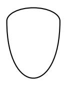
6. `cap`

   This is a cap.

7. `cup`

   This is a cup (an upside-down cap).

The general shape is controlled by the following keys:

| | |
|---|---|
| `flow` | • A cobordism "flows" from its incoming to its outgoing boundaries. This key controls the direction of that flow. The shape is transformed so that the incoming-outgoing axis aligns with the argument. However, the transformation may be more than just a rotation as the shape is set up so that the numbering of the boundary components is always left-to-right or top-to-bottom (as appropriate). Currently, this key can take the values `north`, `south` (default), `east`, and `west`. |
| `view from` | • To get a simulated 3D effect, the cobordism is drawn as if viewed from a slight angle. The value of this key determines whether the cobordism is viewed from the direction of the incoming boundary components or the outgoing ones. This key can take the values `incoming` and `outgoing`. The default is `outgoing`. |
| `cobordism height` | • This is the height of the cobordism ("height" interpreted in its own internal coordinate system). With no offset (q.v.), this would be the distance between the centres of the first incoming and first outgoing boundary components. |
| `boundary separation` | • This is the distance between the centres of the boundary components of the same type. |
| `circle width` | • This is the half-width of the boundary circles. |
| `circle depth` | • This is the half-depth of the boundary circles ("depth" since, in the internal coordinate system, this corresponds to the $z$-axis out of the page). |
| `incoming boundary components` | • The number of incoming boundary components (can be zero). |
| `outgoing boundary components` | • The number of outgoing boundary components (can be zero). |
| `offset` | • This offsets the first outgoing boundary component horizontally relative to the first incoming boundary component. It is a dimensionless number (not necessarily an integer) and is interpreted so that a value of 1 aligns the first outgoing boundary component with the second incoming boundary component. |

## 3 Styling

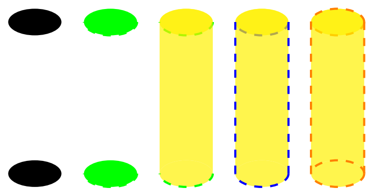
There are various options for styling the diagrams. To understand how they work, it is important to know the order in which a cobordism is drawn and how many pieces it decomposes into. This is the following list, with the corresponding key:

1. The boundary circles are drawn. `boundary style`
2. The lower edges of the boundary circles are redrawn. `boundary lower style`
3. The cobordism itself is drawn.
4. The non-boundary edge of the cobordism is redrawn. `cobordism style`

5. The upper edges of the boundary circles are redrawn. `boundary upper style`

The fact that there are so many is to allow different style to be applied to different pieces. The duplication is to allow certain composite pieces to be *filled*. All of these items can be styled separately. The style given to the node itself is passed on to the third item in that list, the cobordism itself. The styles of the others are controlled by a series of keys, each of should be a list of styles to be applied to that component. Not all options make sense, in particular only the first and third can be filled. (That is, the `fill` style is ignored on the other components.) Here is a progressively built up cobordism.

```
\begin{tikzpicture}
\begin{scope}[tqft/boundary style={fill=purple, fill
    opacity=1}]
\node[tqft/cylinder] at (1,0) {};
\begin{scope}[tqft/boundary lower
    style={draw,dashed,green,thick}]
\node[tqft/cylinder] at (2,0) {};
\begin{scope}
\node[tqft/cylinder, fill=yellow, fill opacity=.7] at
    (3,0) {};
\begin{scope}[tqft/cobordism style={draw,thick,blue}]
\node[tqft/cylinder, fill=yellow, fill opacity=.7] at
    (4,0) {};
\begin{scope}[tqft/boundary upper
    style={draw,thick,orange}]
\node[tqft/cylinder, fill=yellow, fill opacity=.7] at
    (5,0) {};
\end{scope}
\end{scope}
\end{scope}
\end{scope}
\end{scope}
\end{tikzpicture}
```



## 4 Anchors

As with all PGF node shapes, there are certain anchors defined by the `tqft` shape. These are the `center` (and `centre`) anchors and the `incoming boundary n`, `outgoing boundary n` anchors. The positioning of the `center` anchor is slightly unusual in that if there are no, say, incoming boundary components then the centre anchor is still at the same height above the outgoing boundary components as if there were incoming boundary components. The reason for this is two-fold: computing the *actual* centre of the shape in such circumstances would be tricky, and when aligning these shapes it is more useful to have the anchors consistent across shapes of varying boundary components.

There are also the directional anchors `north`, `south`, `east`, `west`, `north east`, `north west`, `south east`, `south west`. The `east` and `west` anchors are placed at the midpoints of the sides. The `north` and `south` anchors are placed in a vertical line with the `center` anchor and vertically aligned with the centres of the corresponding boundary circles. The other four directional anchors are placed at the corners of the cobordism (the placement of these anchors in the case that there are no boundary circles in the corresponding direction may change in future versions).

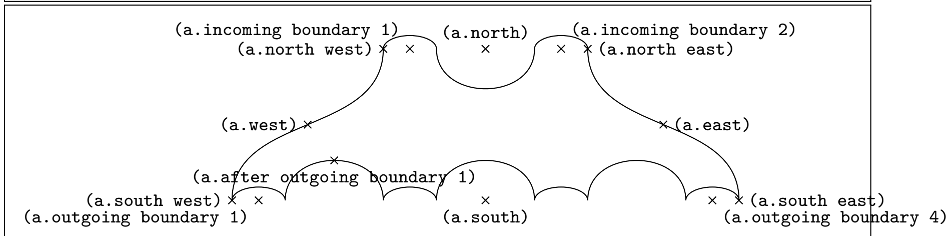
The `incoming boundary n` and `outgoing boundary n` are placed at the centres of the corresponding boundary components, with the numbering starting at the left or the top as appropriate to the flow of the cobordism. A hack borrowed from the `regular polygon` shape ensures that there are always enough anchors for the boundary components.

There are also anchors placed at the midpoint of the cobordism edge between the boundary circles. The names of these are `after incoming boundary n` and `after outgoing boundary n`.

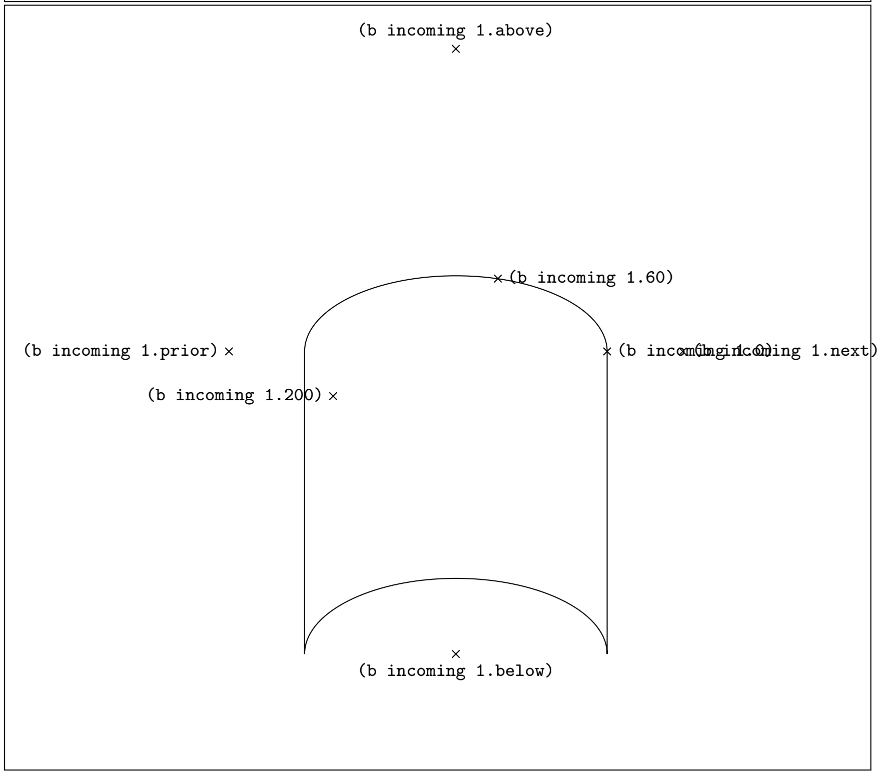
The above anchors can all be "floated" off the cobordism using the keys `outer sep`, `outer xsep`, and `outer ysep`. The last two are the ones actually used, the first is a shortcut for setting both simultaneously.

There are also "sub-nodes". Provding the main node is named, each boundary circle is covered by a `tqft boundary circle` node. This means that the anchors of the `tqft boundary circle` can be used. These cannot be used for placing the main shape, but can be used afterwards. These are not affected by the `outer (x/y)sep` keys. The names of these sub-nodes are of the form `name incoming n` and `name outgoing n` where `name` is the name of the main node. The `tqft boundary circle` shape is based on an ellipse and defines a boundary so the syntax `(name.angle)` works as expected. It also defines anchors `next`, `prior`, `above`, and `below`. These correspond to where the boundary circle in the prescribed direction should be placed.

```
\begin{tikzpicture}
\node[tqft, incoming boundary components=2,outgoing
    boundary components=4,offset=-1,draw] (a) {};
\foreach \anchor/\placement in
{
north/above,
south/below,
east/right,
west/left,
north west/left,
south west/left,
north east/right,
south east/right,
incoming boundary 1/above left,
incoming boundary 2/above right,
outgoing boundary 1/below left,
outgoing boundary 4/below right,
after outgoing boundary 1/below}
\draw[shift=(a.\anchor)] plot[mark=x] coordinates{(0,0)}
    node[\placement] {\scriptsize\texttt{(a.\anchor)}};
\end{tikzpicture}
```



```
\begin{tikzpicture}
\node[tqft,cylinder, circle width=2cm, circle depth=1cm,
    cobordism height=4cm,boundary separation=3cm,draw]
    (b) {};
\foreach \anchor/\placement in
{
prior/left ,
next/right ,
above/above,
below/below,
0/right ,
60/right ,
200/left}
\draw[shift=(b incoming 1.\anchor)] plot[mark=x]
    coordinates{(0,0)} node[\placement]
    {\scriptsize\texttt{(b incoming 1.\anchor)}};
\end{tikzpicture}
```



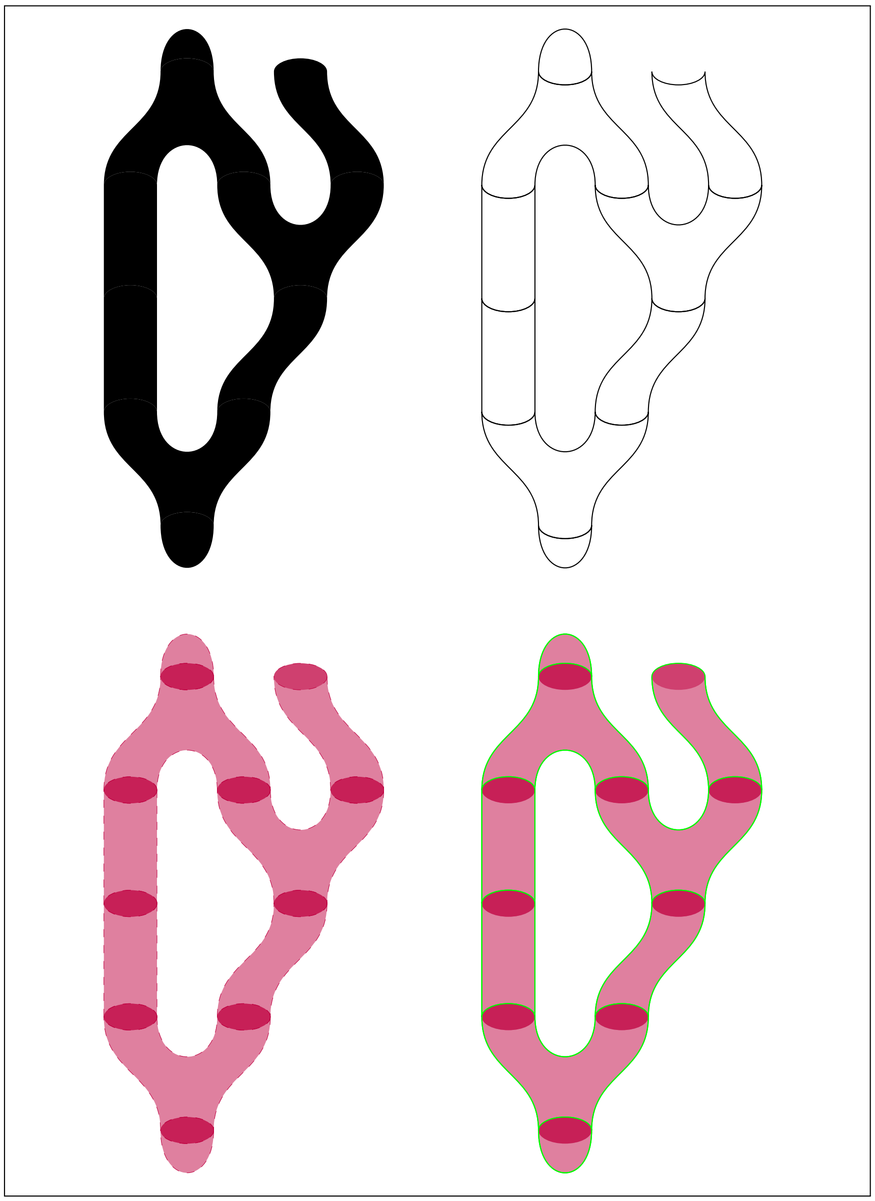
## 5 Improvements

Here are some ideas for extending this, and some minor “bugs”.

1. Make `incoming boundary` an alias of `incoming boundary 1` so that if there is only one incoming boundary component then we don't need to specify the number (ditto outgoing).
2. No thought has been given as to where the text gets placed if it is specified.
3. Add the ability to hide certain boundary components. This is useful if the shapes are not specified in their natural order so certain boundary components should be hidden behind earlier drawn shapes.
4. Some style options on the main node get passed to the other pieces (`fill opacity` being one). This shouldn't happen, or should happen by design not by accident.
5. The bounding box isn't as good as it could be.
6. Add a way to specify more directions for the flow.
7. Add the ability to apply different styles to the incoming and outgoing components.

## 6 More Examples

```
\begin{tikzpicture}[tqft/cobordism
    height=1.5cm, tqft/boundary  separation=1.5cm]
\foreach \coord/\style in {
{(0,0)}/{tqft/view from=outgoing, fill},
{(5,0)}/{tqft/view from=incoming,draw},
{(0,-8)}/{fill=orange, fill opacity=.5,tqft/boundary
    lower  style={draw,blue,ultra
    thin,dashed},tqft/boundary upper
    style={draw,green},tqft/cobordism
    style={draw,purple},tqft/boundary
    style={fill=yellow}},
{(5,-8)}/{fill=orange, fill opacity=.5,tqft/cobordism
    style={draw,purple},tqft/boundary
    style={fill=yellow,draw=green}}
} {
\begin{scope}
\edef\styleit{\noexpand\tikzset{every
    node/.style={\style}}}
\styleit
\node[tqft/cap] (h) at \coord {};
\node[tqft/pair of pants,anchor=incoming boundary 1] (a)
    at  (h.outgoing boundary 1) {};
\node[tqft/cylinder to next,anchor=incoming boundary 1]
    (d) at  (a.incoming boundary 2) {};
\node[tqft/reverse pair of pants,anchor=incoming
    boundary 1] (b) at (a.outgoing boundary 2) {};
\node[tqft/cylinder to prior,anchor=incoming boundary 1]
    (c) at  (b.outgoing boundary 1) {};
\node[tqft/cylinder,anchor=incoming boundary 1] (e) at
    (a.outgoing  boundary 1) {};
\node[tqft/cylinder,anchor=incoming boundary 1] (f) at
    (e.outgoing  boundary 1) {};
\node[tqft/reverse pair of pants,anchor=incoming
    boundary 1] (g) at  (f.outgoing boundary 1) {};
\node[tqft/cup,anchor=incoming boundary 1] (i) at
    (g.outgoing boundary 1) {};
\end{scope}
}
\end{tikzpicture}
```



```
\begin{tikzpicture}
\node[draw,tqft/pair of pants] (a) {};
\node[draw,tqft/cap,anchor=outgoing boundary 1] at
    (a.incoming boundary 1) {};
\node[fill,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 1) {};
\node[draw,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 2) {};
\begin{scope}[tqft/flow=east]
\node[draw,tqft/pair of pants] (a) at (4,0) {};
\node[draw,tqft/cap,anchor=outgoing boundary 1] at
    (a.incoming boundary 1) {};
\node[fill,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 1) {};
\node[draw,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 2) {};
\end{scope}
\begin{scope}[tqft/flow=north]
\node[draw,tqft/pair of pants] (a) at (0,-4) {};
\node[draw,tqft/cap,anchor=outgoing boundary 1] at
    (a.incoming boundary 1) {};
\node[fill,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 1) {};
\node[draw,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 2) {};
\end{scope}
\begin{scope}[tqft/flow=west]
\node[draw,tqft/pair of pants] (a) at (4,-4) {};
\node[draw,tqft/cap,anchor=outgoing boundary 1] at
    (a.incoming boundary 1) {};
\node[fill,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 1) {};
\node[draw,tqft/cup,anchor=incoming boundary 1] at
    (a.outgoing boundary 2) {};
\end{scope}
\end{tikzpicture}
```