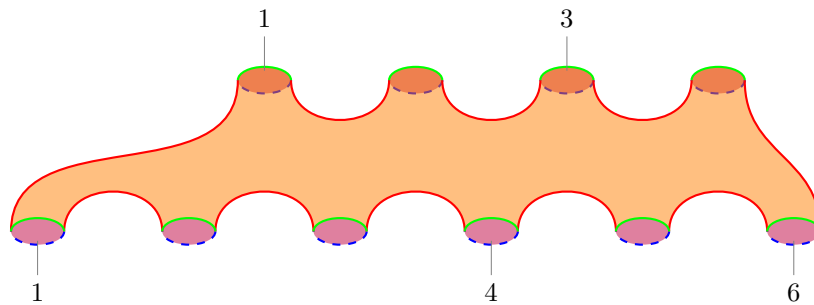# The **tqft** package: codebase

Andrew Stacey

loopspace@mathforge.org

? from ?

1 Introduction

This is a package for drawing TQFT diagrams using PGF/TikZ. Its inspiration was a question and answer on the website http://tex.stackexchance.com.

2 Implementation

2.1 Old Version: Node Shapes

```
1 \RequirePackage{pgfkeys}
2 \RequirePackage{pgf}
```

We can view the cobordisms from the *input* or *output* ends, the implementation of the choice is to draw an arc from 0 to 180 or from 0 to -180 so we just need to track minus signs. These macros are for that.

```
3 \def\pgf@tqft@minus{-}
4 \let\pgf@tqft@upper\@empty
5 \let\pgf@tqft@lower\pgf@tqft@minus
```

Some helpful extra functions.

`\tqftset` is our equivalent of `\tikzset`.

```
6 \def\tqftset#1{\pgfkeys{/pgf/tqft/.cd,#1}}
```

`\tqft@process` This macro applies our flow transformation to the given coordinates, with the result stored in `\pgf@x` and `\pgf@y`.

```
7 \def\tqft@process#1#2{%
8   \edef\tqft@px{#1}
9   \edef\tqft@py{#2}
10   \pgf@process{
```

```
11      \pgftransformreset
12      \let\tikz@transform=\pgfutil@empty
13      \expandafter\tikzset\expandafter{\tqft@transformation}
14      \tikz@transform
15      \pgfpointtransformed{\pgfqpoint{\tqft@px}{\tqft@py}}
16   }
17 }
```

Declare some dimension registers to hold the specifications of the cobordism.

```
18 \newdimen\tqft@xa
19 \newdimen\tqft@xb
20 \newdimen\tqft@c
21 \newdimen\tqft@ch
22 \newdimen\tqft@h
23 \newdimen\tqft@s
24 \newdimen\tqft@w
25 \newif\iftqft@within@node
```

Now we set up all the keys that we'll need in the course of this shape

```
26 \pgfkeys{
```

Add a key to switch between the two versions.

```
27   /tikz/tqft/use nodes/.is choice,
28   /tikz/tqft/use nodes/true/.code={%
29     \tikzset{
```

If using nodes, set the defaults

```
30       tqft/.style={%
31         /tikz/shape=tqft cobordism,
32         /pgf/tqft,
33         /tikz/every tqft/.try
34       },
```

Unknowns go to `/pgf/tqft`

```
35       tqft/.unknown/.code={%
36         \let\tqft@searchname=\pgfkeyscurrentname%
37         \pgfkeys{%
38           /pgf/tqft/\tqft@searchname={##1}
39         }
40       },
41     }%
42   },
```

If not using nodes, set the defaults for the library

```
43   /tikz/tqft/use nodes/false/.code={%
44     \tikzset{
45       tqft/.style={%
46         pic type=cobordism,
47         tqft/.cd,
48         every tqft/.try,
49       },
```

Pass unknown keys on to TikZ.

```
50       tqft/.unknown/.code={%
51         \let\tqft@searchname=\pgfkeyscurrentname%
52         \pgfkeys{%
```

```
53          /tikz/\tqft@searchname={##1}
54        }
55      },
56    }%
57  },
58  /tikz/tqft/use nodes=true,
```

This deals with unknown keys, passing them on to TikZ.

```
59  /pgf/tqft/.unknown/.code={%
60    \let\tqft@searchname=\pgfkeyscurrentname%
61    \pgfkeysalso{%
62      /tikz/\tqft@searchname={#1}
63    }
64  },
```

Let's play happy families!

```
65  /pgf/tqft/.is family,
66  /pgf/tqft,
```

This sets our shape to be the boundary circle

```
67  boundary circle/.style={
68    /tikz/shape=tqft boundary circle
69  },
```

These set our number of boundary components

```
70  incoming boundary components/.initial=5,
71  outgoing boundary components/.initial=4,
```

This is the "horizontal" offset of the first outgoing component from the first incoming one.

```
72  offset/.initial=0,
```

This is the "vertical" separation between boundary components.

```
73  cobordism height/.initial=2cm,
```

This is the "horizontal" separation between boundary components.

```
74  boundary separation/.initial=2cm,
```

These are the "horizontal" and "vertical" radii, respectively, of the boundary components (perhaps poorly named!).

```
75  circle width/.initial=10pt,
76  circle depth/.initial=5pt,
```

These control the separation between the node and its anchors.

```
77  outer xsep/.initial=0pt,
78  outer ysep/.initial=0pt,
79  outer sep/.style={
80    outer xsep=#1,
81    outer ysep=#1
82  },
```

This is our flow control. The `flow` key installs a transformation to be applied to our node shape. The possible transformations are stored in the following keys. They aren't just rotations so that the numbering is always "top to bottom" or "left to right".

```
83  flow/.code={%
84    \pgfkeys{/pgf/tqft/flow transformation/.expand twice/.expand once=\pgfkeysvalueof{/pgf/t
```

```
85    },
86    flow transformation south/.initial={},
87    flow transformation north/.initial={%
88      xscale=-1,rotate=180
89    },
90    flow transformation east/.initial={%
91      rotate=90,xscale=-1
92    },
93    flow transformation west/.initial={%
94      rotate=270
95    },
96    flow transformation/.initial={},
```

These control the direction from which we view the cobordism.

```
97    view from/.is choice,
98    view from/incoming/.code={%
99      \let\pgf@tqft@upper\pgf@tqft@minus
100     \let\pgf@tqft@lower\@empty
101   },
102   view from/outgoing/.code={%
103   \let\pgf@tqft@lower\pgf@tqft@minus
104     \let\pgf@tqft@upper\@empty
105   },
```

The next set of keys are for styling the different pieces of a cobordism.

```
106   boundary lower style contents/.initial={},
107   boundary lower style/.code={%
108     \pgfkeys{/pgf/tqft/boundary lower style contents/.style={%
109         /tikz/.cd,#1
110       }
111     }
112   },
113   boundary style contents/.initial={},
114   boundary style/.code={%
115     \pgfkeys{/pgf/tqft/boundary style contents/.style={%
116         /tikz/.cd,#1
117       }
118     }
119   },
120   boundary upper style contents/.initial={},
121   boundary upper style/.code={%
122     \pgfkeys{/pgf/tqft/boundary upper style contents/.style={%
123         /tikz/.cd,#1
124       }
125     }
126   },
127   cobordism style contents/.initial={},
128   cobordism style/.code={%
129     \pgfkeys{/pgf/tqft/cobordism style contents/.style={%
130         /tikz/.cd,#1%
131       }
132     }
133   },
```

The next set of keys define some default shapes.

```
134  pair of pants/.style={
135    /tikz/tqft,
136    incoming boundary components=1,
137    outgoing boundary components=2,
138    offset=-.5
139  },
140  /tikz/tqft pair of pants/.style={
141    /pgf/tqft/pair of pants,
142  },
143  reverse pair of pants/.style={
144    /tikz/tqft,
145    incoming boundary components=2,
146    outgoing boundary components=1,
147    offset=.5
148  },
149  /tikz/tqft reverse pair of pants/.style={
150    /pgf/tqft/reverse pair of pants,
151  },
152  cylinder to prior/.style={
153    /tikz/tqft,
154    incoming boundary components=1,
155    outgoing boundary components=1,
156    offset=-.5
157  },
158  /tikz/tqft cylinder to prior/.style={
159    /pgf/tqft/cylinder to prior,
160  },
161  cylinder to next/.style={
162    /tikz/tqft,
163    incoming boundary components=1,
164    outgoing boundary components=1,
165    offset=.5
166  },
167  /tikz/tqft cylinder to next/.style={
168    /pgf/tqft/cylinder to next,
169  },
170  cylinder/.style={
171    /tikz/tqft,
172    incoming boundary components=1,
173    outgoing boundary components=1
174  },
175  /tikz/tqft cylinder/.style={
176    /pgf/tqft/cylinder,
177  },
178  cup/.style={
179    /tikz/tqft,
180    incoming boundary components=1,
181    outgoing boundary components=0
182  },
183  /tikz/tqft cup/.style={
184    /pgf/tqft/cup,
185  },
186  cap/.style={
187    /tikz/tqft,
```

```
188     incoming boundary components=0,
189     outgoing boundary components=1
190   },
191   /tikz/tqft cap/.style={
192     /pgf/tqft/cap,
193   },
194 }
```

tqft shape This is a generic cobordism shape

```
195 \pgfdeclareshape{tqft cobordism}{
```

Save our specifications: incoming and outgoing boundary components

```
196   \savedmacro{\tqft@incoming}{\edef\tqft@incoming{\pgfkeysvalueof{/pgf/tqft/incoming boundar
197   \savedmacro{\tqft@outgoing}{\edef\tqft@outgoing{\pgfkeysvalueof{/pgf/tqft/outgoing boundar
```

and the offset (in units of boundary components) between the leading incoming
and outgoing components (regarded as a shift of the outgoing components relative
to the incoming)

```
198   \savedmacro{\tqft@offset}{\edef\tqft@offset{\pgfkeysvalueof{/pgf/tqft/offset}}}
```

Now we save our dimensions: height, separation, the radii of the boundary circles,
and outer seps, and the heights of the control points.

```
199   \saveddimen{\tqft@height}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/cobordism height}}
200   \saveddimen{\tqft@separation}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/boundary separation}}
201   \saveddimen{\tqft@width}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/circle width}}
202   \saveddimen{\tqft@depth}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/circle depth}}
203   \saveddimen{\tqft@outerxsep}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/outer xsep}}
204   \saveddimen{\tqft@outerysep}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/outer ysep}}
205   \saveddimen{\tqft@control}{
206     \pgfkeysgetvalue{/pgf/tqft/cobordism height}{\tqft@tempa}
207     \pgfkeysgetvalue{/pgf/tqft/circle depth}{\tqft@tempb}
208     \pgfmathsetlength{\pgf@x}{.5 * \tqft@tempa - 4 * \tqft@tempb}
209   }
```

This is the internal transformation that is in place

```
210   \savedmacro{\tqft@transformation}{%
211     \pgfkeysgetvalue{/pgf/tqft/flow transformation}{\tqft@transformation}
212   }
```

For the externally available anchors, we need to save a few things as well.
    Position of first incoming boundary in internal coordinates

```
213   \savedanchor{\tqft@start@incoming}{%
214     \pgfmathsetlength{\pgf@x}{-(max(\pgfkeysvalueof{/pgf/tqft/incoming boundary components}
215     \pgfmathsetlength{\pgf@y}{.5 * \pgfkeysvalueof{/pgf/tqft/cobordism height}}
216   }
```

Position of first outgoing boundary in internal coordinates

```
217   \savedanchor{\tqft@start@outgoing}{%
218     \pgfmathsetlength{\pgf@x}{-(max(\pgfkeysvalueof{/pgf/tqft/incoming boundary components}
219     \pgfmathsetlength{\pgf@y}{-.5 * \pgfkeysvalueof{/pgf/tqft/cobordism height}}
220   }
```

For completeness, we record the size of the text box (not that we expect any text,
but you never know)

```
221 \savedanchor{\tqft@textsize}{%
222   \pgf@y=-.5\ht\pgfnodeparttextbox%
```

6

```
223    \pgf@x=-.5\wd\pgfnodeparttextbox%
224 }
```

These are our externally available anchors

```
225    \anchor{centre}{\pgfpointorigin}
226    \anchor{center}{\pgfpointorigin}
227 \anchor{text}{
228    \tqft@textsize
229 }
230 \anchor{north}{%
231    \pgf@ya=\tqft@height\relax
232    \pgf@yb=.5\pgf@ya
233    \advance\pgf@yb by \tqft@outerysep\relax
234    \tqft@process{0pt}{\the\pgf@yb}
235 }
236 \anchor{south}{%
237    \pgf@yb=\tqft@height\relax
238    \pgf@ya=.5\pgf@yb
239    \advance\pgf@ya by \tqft@outerysep\relax
240    \pgf@yb=-\pgf@ya
241    \tqft@process{0pt}{\the\pgf@yb}
242 }
243 \anchor{west}{%
244    \tqft@start@incoming
245    \pgf@xa=\pgf@x
246    \advance\pgf@xa by -\tqft@width
247    \pgf@ya=\pgf@y
248    \tqft@start@outgoing
249    \pgf@xb=\pgf@x
250    \advance\pgf@xb by -\tqft@width
251    \pgf@yb=\pgf@y
252    \pgf@xc=.5\pgf@xa
253    \advance\pgf@xc by .5\pgf@xb
254    \pgf@yc=.5\pgf@ya
255    \advance\pgf@yc by .5\pgf@yb
256    \advance\pgf@xc by -\tqft@outerxsep\relax
257    \tqft@process{\the\pgf@xc}{\the\pgf@yc}
258 }
259 \anchor{east}{%
260    \tqft@start@incoming
261    \pgf@xa=\pgf@x
262    \pgfmathsetlength{\pgf@xa}{\pgf@xa + (\tqft@incoming - 1) * \tqft@separation}
263    \advance\pgf@xa by \tqft@width\relax
264    \pgf@ya=\pgf@y
265    \tqft@start@outgoing
266    \pgf@xb=\pgf@x
267    \pgfmathsetlength{\pgf@xb}{\pgf@xb + (\tqft@outgoing - 1) * \tqft@separation}
268    \advance\pgf@xb by \tqft@width\relax
269    \pgf@yb=\pgf@y
270    \pgf@xc=.5\pgf@xa
271    \advance\pgf@xc by .5\pgf@xb
272    \pgf@yc=.5\pgf@ya
273    \advance\pgf@yc by .5\pgf@yb
274    \advance\pgf@xc by \tqft@outerxsep\relax
275    \tqft@process{\the\pgf@xc}{\the\pgf@yc}
```

```
276 }
277 \anchor{north west}{
278     \tqft@start@incoming
279     \pgf@xc=\pgf@x
280     \pgf@yc=\pgf@y
281     \advance\pgf@xc by -\tqft@width\relax
282     \advance\pgf@yc by \tqft@outerysep\relax
283     \advance\pgf@xc by -\tqft@outerxsep\relax
284     \tqft@process{\the\pgf@xc}{\the\pgf@yc}
285 }
286 \anchor{south west}{
287     \tqft@start@outgoing
288     \pgf@xc=\pgf@x
289     \pgf@yc=\pgf@y
290     \advance\pgf@xc by -\tqft@width\relax
291     \advance\pgf@yc by -\tqft@outerysep\relax
292     \advance\pgf@xc by -\tqft@outerxsep\relax
293     \tqft@process{\the\pgf@xc}{\the\pgf@yc}
294 }
295 \anchor{north east}{
296     \tqft@start@incoming
297     \pgf@xc=\pgf@x
298     \pgfmathsetlength{\pgf@xc}{\pgf@xc + (\tqft@incoming - 1)*\tqft@separation}
299     \pgf@yc=\pgf@y
300     \advance\pgf@xc by \tqft@width\relax
301     \advance\pgf@yc by \tqft@outerysep\relax
302     \advance\pgf@xc by \tqft@outerxsep\relax
303     \tqft@process{\the\pgf@xc}{\the\pgf@yc}
304 }
305 \anchor{south east}{
306     \tqft@start@outgoing
307     \pgf@xc=\pgf@x
308     \pgfmathsetlength{\pgf@xc}{\pgf@xc + (\tqft@outgoing - 1)*\tqft@separation}
309     \pgf@yc=\pgf@y
310     \advance\pgf@xc by \tqft@width\relax
311     \advance\pgf@yc by -\tqft@outerysep\relax
312     \advance\pgf@xc by \tqft@outerxsep\relax
313     \tqft@process{\the\pgf@xc}{\the\pgf@yc}
314 }
```

To define anchors at the boundary components requires a bit of trickery borrowed
from the "regular polygon" shape.

```
315 \expandafter\pgfutil@g@addto@macro\csname pgf@sh@s@tqft cobordism\endcsname{%
316     \c@pgf@counta\tqft@incoming\relax%
317     \pgfmathloop%
318     \ifnum\c@pgf@counta>0\relax%
319     \pgfutil@ifundefined{pgf@anchor@tqft cobordism@incoming boundary\space\the\c@pgf@counta}{%
320         \expandafter\xdef\csname pgf@anchor@tqft cobordism@incoming boundary\space\the\c@pgf@cou
321     \noexpand\tqft@start@incoming
322     \noexpand\pgfmathsetlength{\noexpand\pgf@y}{\noexpand\pgf@y + \noexpand\tqft@outerysep}
323     \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - 1) * \n
324     \noexpand\tqft@process{\noexpand\the\noexpand\pgf@x}{\noexpand\the\noexpand\pgf@y}
325     }
326     }{\c@pgf@counta0\relax}%
```

```
327    \advance\c@pgf@counta-1\relax%
328    \repeatpgfmathloop%
329 }
330 \expandafter\pgfutil@g@addto@macro\csname pgf@sh@s@tqft cobordism\endcsname{%
331    \c@pgf@counta\tqft@outgoing\relax%
332    \pgfmathloop%
333    \ifnum\c@pgf@counta>0\relax%
334    \pgfutil@ifundefined{pgf@anchor@tqft cobordism@outgoing boundary\space\the\c@pgf@counta}{%
335      \expandafter\xdef\csname pgf@anchor@tqft cobordism@outgoing boundary\space\the\c@pgf@cou
336    \noexpand\tqft@start@outgoing
337    \noexpand\pgfmathsetlength{\noexpand\pgf@y}{\noexpand\pgf@y - \noexpand\tqft@outerysep}
338    \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - 1) * \n
339    \noexpand\tqft@process{\noexpand\the\noexpand\pgf@x}{\noexpand\the\noexpand\pgf@y}
340    }
341    }{\c@pgf@counta0\relax}%
342    \advance\c@pgf@counta-1\relax%
343    \repeatpgfmathloop%
344 }
345 \expandafter\pgfutil@g@addto@macro\csname pgf@sh@s@tqft cobordism\endcsname{%
346    \c@pgf@counta\tqft@incoming\relax%
347    \advance\c@pgf@counta-1\relax
348    \pgfmathloop%
349    \ifnum\c@pgf@counta>0\relax%
350    \pgfutil@ifundefined{pgf@anchor@tqft cobordism@after incoming boundary\space\the\c@pgf@cou
351      \expandafter\xdef\csname pgf@anchor@tqft cobordism@after incoming boundary\space\the\c@p
352    \noexpand\tqft@start@incoming
353    \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y +.75 * \noexpand\tqft@co
354    \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \
355    \noexpand\tqft@process{\noexpand\the\noexpand\pgf@x}{\noexpand\the\noexpand\pgf@y}
356    }
357    }{\c@pgf@counta0\relax}%
358    \advance\c@pgf@counta-1\relax%
359    \repeatpgfmathloop%
360 }
361 \expandafter\pgfutil@g@addto@macro\csname pgf@sh@s@tqft cobordism\endcsname{%
362    \c@pgf@counta\tqft@outgoing\relax%
363    \advance\c@pgf@counta-1\relax
364    \pgfmathloop%
365    \ifnum\c@pgf@counta>0\relax%
366    \pgfutil@ifundefined{pgf@anchor@tqft cobordism@after outgoing boundary\space\the\c@pgf@cou
367      \expandafter\xdef\csname pgf@anchor@tqft cobordism@after outgoing boundary\space\the\c@p
368    \noexpand\tqft@start@outgoing
369    \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y -.75 * \noexpand\tqft@co
370    \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \
371    \noexpand\tqft@process{\noexpand\the\noexpand\pgf@x}{\noexpand\the\noexpand\pgf@y}
372    }
373    }{\c@pgf@counta0\relax}%
374    \advance\c@pgf@counta-1\relax%
375    \repeatpgfmathloop%
376 }
```

Now we define the background path. This is the upper part of the cobordism.

```
377    \backgroundpath{
```

Apply the internal transformation

```
378        \let\tikz@transform=\pgfutil@empty
379        \expandafter\tikzset\expandafter{\tqft@transformation}
380        \tikz@transform
```

Convert the boundary separation and width to lengths

```
381        \pgfmathsetlength{\tqft@s}{\tqft@separation}
382        \pgfmathsetlength{\tqft@w}{2*\tqft@width}
```

Compute the starting position of the incoming boundary components so that we get the centre anchor on the centre of the cobordism

```
383 \tqft@start@incoming
384 \tqft@xa=\pgf@x
385        \advance\tqft@xa by -.5\tqft@w\relax
386 \tqft@h=\pgf@y
387        \tqft@xb=\tqft@xa
388        \advance\tqft@xb by \tqft@w\relax
389 \tqft@c=\tqft@control\relax
```

Do we have any incoming boundary components at all?

```
390        \ifnum\tqft@incoming>0
```

Yes, so move to the position of the first and draw it

```
391        \pgfpathmoveto{\pgfqpoint{\tqft@xa}{\tqft@h}}
392        \pgfpatharc{\pgf@tqft@upper180}{0}{\tqft@width and \tqft@depth}
```

Do we have any more incoming boundary components?

```
393        \ifnum\tqft@incoming>1
```

Yes, so iterate over the remaining incoming boundary components

```
394        \foreach \tqft@k in {2,...,\tqft@incoming} {
395        \advance\tqft@xa by \tqft@k\tqft@s
396        \advance\tqft@xb by \tqft@k\tqft@s
397        \advance\tqft@xb by -2\tqft@s
398        \advance\tqft@xa by -\tqft@s
399        \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@c}}{\pgfqpoint{\tqft@xa}{\tqft@c}}{\pgfqpoin
400        \pgfpatharc{\pgf@tqft@upper180}{0}{\tqft@width and \tqft@depth}
401        }
402        \fi
```

If we don't have any outgoing boundary components, may as well close up now.

```
403        \ifnum\tqft@outgoing=0
404        \advance\tqft@xb by \tqft@incoming\tqft@s
405        \advance\tqft@xb by -\tqft@s
406        \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
407        \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpo
408        \fi
409        \fi
```

Shift down to the outgoing components, if we have any

```
410        \ifnum\tqft@outgoing>0
411        \advance\tqft@xb by \tqft@incoming\tqft@s
412        \advance\tqft@xb by -\tqft@s
413        \pgfmathsetlength{\tqft@xa}{\tqft@xa + (\tqft@outgoing - 1 + \tqft@offset) * \tqft@sepa
```

If we had incoming boundaries, this is a curveto, otherwise it's a moveto

```
414        \ifnum\tqft@incoming>0
415        \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
```

```
416       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{-
     \tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
417       \else
418       \pgfpathmoveto{\pgfqpoint{\tqft@xa}{-\tqft@h}}
419       \fi
420       \tqft@xb=\tqft@xa
421       \advance\tqft@xb by -\tqft@w
```

Now draw the lower components

```
422       \pgfpatharc{0}{\pgf@tqft@upper180}{\tqft@width and \tqft@depth}
```

Now iterate over the remaining outgoing boundary components

```
423       \ifnum\tqft@outgoing>1
424       \foreach \tqft@k in {2,...,\tqft@outgoing} {
425       \advance\tqft@xa by -\tqft@k\tqft@s
426       \advance\tqft@xb by -\tqft@k\tqft@s
427       \advance\tqft@xb by 2\tqft@s
428       \advance\tqft@xa by \tqft@s
429       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@c}}{\pgfqpoint{\tqft@xa}{-
     \tqft@c}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
430       \pgfpatharc{0}{\pgf@tqft@upper180}{\tqft@width and \tqft@depth}
431       }
432       \fi
```

Shift back up to the incoming components, if we had any, otherwise arc back to
our starting point

```
433       \advance\tqft@xb by -\tqft@outgoing\tqft@s
434       \advance\tqft@xb by \tqft@s
435       \ifnum\tqft@incoming>0
436       \pgfmathsetlength{\tqft@xa}{\tqft@xa - (\tqft@outgoing -1 + \tqft@offset) * \tqft@separ
437       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
438       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqp
439       \else
440       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
441       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{-
     \tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
442       \fi
443       \fi
```

Close the path

```
444       \pgfpathclose
445   }
```

End of background path Now we define the behind background path. This is the
lower part of the boundary circles.

```
446   \behindbackgroundpath{
```

Apply the internal transformation

```
447       \let\tikz@transform=\pgfutil@empty
448       \expandafter\tikzset\expandafter{\tqft@transformation}
449       \tikz@transform
```

Convert the boundary separation and width to lengths

```
450       \pgfmathsetlength{\tqft@s}{\tqft@separation}
451       \pgfmathsetlength{\tqft@w}{2*\tqft@width}
```

Compute the starting position of the incoming boundary components so that we get the centre anchor on the centre of the cobordism

```
452    \pgfmathsetlength{\tqft@xa}{-(max(\tqft@incoming - 1,\tqft@outgoing - 1 + \tqft@offset)
453    \pgfmathsetlength{\tqft@h}{.5 * \tqft@height}
```

This section draws the boundary circles

```
454    {
```

Initialise the TikZ path settings and read in the style options for the boundary

```
455        \tikz@mode@fillfalse%
456        \tikz@mode@drawfalse%
457        \let\tikz@mode=\pgfutil@empty
458        \let\tikz@options=\pgfutil@empty
459        \tqftset{boundary style contents}
460        \tikz@mode
461        \tikz@options
```

Do we have any incoming boundary components at all?

```
462    \ifnum\tqft@incoming>0
```

Yes, so iterate over them

```
463    \foreach \tqft@k in {1,...,\tqft@incoming} {
464    \advance\tqft@xa by \tqft@k\tqft@s
465    \pgfpathellipse{\pgfqpoint{\tqft@xa}{\tqft@h}}{\pgfqpoint{\tqft@width}{0pt}}{\pgfqpoint
466    }
467    \fi
```

Now iterate over the outgoing boundary components, if we have any

```
468    \ifnum\tqft@outgoing>0
469    \pgfmathsetlength{\tqft@xa}{\tqft@xa + (\tqft@outgoing + \tqft@offset + 1) * \tqft@sepa
470    \foreach \tqft@k in {1,...,\tqft@outgoing} {
471    \advance\tqft@xa by -\tqft@k\tqft@s
472 %    \advance\tqft@xa by \tqft@s
473    \pgfpathellipse{\pgfqpoint{\tqft@xa}{-\tqft@h}}{\pgfqpoint{\tqft@width}{0pt}}{\pgfqpoin
474    }
475    \fi

476      \edef\tikz@temp{\noexpand\pgfusepath{%
477          \iftikz@mode@fill fill,\fi%
478          \iftikz@mode@draw draw\fi%
479      }}%
480      \tikz@temp
481    }
```

This section draws the lower parts of the boundary circles

```
482    {
```

Initialise the TikZ path settings and read in the style options for the boundary

```
483        \tikz@mode@fillfalse%
484        \tikz@mode@drawfalse%
485        \let\tikz@mode=\pgfutil@empty
486        \let\tikz@options=\pgfutil@empty
487        \tqftset{boundary lower style contents}
488        \tikz@mode
489        \tikz@options
490        \advance\tqft@xa by .5\tqft@w
```

Do we have any incoming boundary components at all?

```
491      \ifnum\tqft@incoming>0
```

Yes, so iterate over them

```
492      \foreach \tqft@k in {1,...,\tqft@incoming} {
493      \advance\tqft@xa by \tqft@k\tqft@s
494      \pgfpathmoveto{\pgfqpoint{\tqft@xa}{\tqft@h}}
495      \pgfpatharc{0}{\pgf@tqft@lower180}{\tqft@width and \tqft@depth}
496      }
497      \fi
```

Now iterate over the outgoing boundary components, if we have any

```
498      \ifnum\tqft@outgoing>0
499      \pgfmathsetlength{\tqft@xa}{\tqft@xa + (\tqft@outgoing + \tqft@offset + 1) * \tqft@sepa
500      \foreach \tqft@k in {1,...,\tqft@outgoing} {
501      \advance\tqft@xa by -\tqft@k\tqft@s
502 %     \advance\tqft@xa by \tqft@s
503      \pgfpathmoveto{\pgfqpoint{\tqft@xa}{-\tqft@h}}
504      \pgfpatharc{0}{\pgf@tqft@lower180}{\tqft@width and \tqft@depth}
505      }
506      \fi
507      \edef\tikz@temp{\noexpand\pgfusepath{%
508          \iftikz@mode@fill fill,\fi%
509          \iftikz@mode@draw draw\fi%
510      }}%
511      \tikz@temp
512      }
513   }
```

End of behind background path.

   Now we define the before background path. This is the upper part of the
boundary circles and the cobordism edge.

```
514   \beforebackgroundpath{
```

We *don't* apply the internal transformation as it is already in place from the
\backgroundpath. Convert the boundary separation and width to lengths

```
515      \pgfmathsetlength{\tqft@s}{\tqft@separation}
516      \pgfmathsetlength{\tqft@w}{2*\tqft@width}
```

Compute the starting position of the incoming boundary components so that we
get the centre anchor on the centre of the cobordism

```
517      \pgfmathsetlength{\tqft@xa}{-(max(\tqft@incoming - 1,\tqft@outgoing - 1 + \tqft@offset)
518      \tqft@xb=\tqft@xa
519      \advance\tqft@xb by \tqft@w
520 \tqft@c=\tqft@control\relax
521      \pgfmathsetlength{\tqft@h}{.5 * \tqft@height}
```

This section draws the non-boundary part of the cobordism.

```
522      {
```

Initialise the TikZ path settings and read in the style options for the boundary

```
523          \tikz@mode@fillfalse%
524          \tikz@mode@drawfalse%
525          \let\tikz@mode=\pgfutil@empty
526          \let\tikz@options=\pgfutil@empty
527          \tqftset{cobordism style contents}
```

13

```
528        \tikz@mode
529        \tikz@options
530 % Do we have any incoming boundary components at all?
531 %    \begin{macrocode}
532        \ifnum\tqft@incoming>0
```

Do we have more than one?

```
533        \ifnum\tqft@incoming>1
```

Yes, so iterate over the remaining incoming boundary components

```
534        \foreach \tqft@k in {2,...,\tqft@incoming} {
535        \advance\tqft@xa by \tqft@k\tqft@s
536        \advance\tqft@xb by \tqft@k\tqft@s
537        \advance\tqft@xb by -2\tqft@s
538        \advance\tqft@xa by -\tqft@s
539        \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
540        \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@c}}{\pgfqpoint{\tqft@xa}{\tqft@c}}{\pgfqpoin
541        }
542        \fi
```

If we don't have any outgoing boundary components, may as well close up now.

```
543        \ifnum\tqft@outgoing=0
544        \advance\tqft@xb by \tqft@incoming\tqft@s
545        \advance\tqft@xb by -\tqft@s
546        \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
547        \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
548        \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpo
549        \fi
550        \fi
```

Shift down to the outgoing components, if we have any

```
551        \ifnum\tqft@outgoing>0
552        \advance\tqft@xb by \tqft@incoming\tqft@s
553        \advance\tqft@xb by -\tqft@s
554        \pgfmathsetlength{\tqft@xa}{\tqft@xa + (\tqft@outgoing - 1 + \tqft@offset) * \tqft@sepa
```

If we had incoming boundaries, this is a curveto, otherwise it's a moveto

```
555        \ifnum\tqft@incoming>0
556        \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
557        \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
558        \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{-
   \tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
559        \else
560        \pgfpathmoveto{\pgfqpoint{\tqft@xa}{-\tqft@h}}
561        \fi
562        \tqft@xb=\tqft@xa
563        \advance\tqft@xb by -\tqft@w
```

Now draw the lower components

```
564        \pgfpathmoveto{\pgfqpoint{\tqft@xb}{-\tqft@h}}
```

Now iterate over the remaining outgoing boundary components

```
565        \ifnum\tqft@outgoing>1
566        \foreach \tqft@k in {2,...,\tqft@outgoing} {
567        \advance\tqft@xa by -\tqft@k\tqft@s
568        \advance\tqft@xb by -\tqft@k\tqft@s
569        \advance\tqft@xb by 2\tqft@s
```

```
570      \advance\tqft@xa by \tqft@s
571      \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@c}}{\pgfqpoint{\tqft@xa}{-
   \tqft@c}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
572      \advance\tqft@xa by -\tqft@w
573      \pgfpathmoveto{\pgfqpoint{\tqft@xa}{-\tqft@h}}
574      }
575      \fi
```

Shift back up to the incoming components, if we had any, otherwise arc back to
our starting point

```
576      \advance\tqft@xb by -\tqft@outgoing\tqft@s
577      \advance\tqft@xb by \tqft@s
578      \ifnum\tqft@incoming>0
579      \pgfmathsetlength{\tqft@xa}{\tqft@xa - (\tqft@outgoing -1 + \tqft@offset) * \tqft@separ
580      \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
581      \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqp
582      \else
583      \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\t
584      \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{-
   \tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@h}}
585      \fi
586      \fi
587      \edef\tikz@temp{\noexpand\pgfusepath{%
588          \iftikz@mode@fill fill,\fi%
589          \iftikz@mode@draw draw\fi%
590      }}%
591      \tikz@temp
592 }
```

This section draws the upper parts of the boundary circles

```
593      {
```

Initialise the TikZ path settings and read in the style options for the boundary

```
594 \let\tqft@bdry@path=\pgfutil@empty
595 \let\tqft@bdry@node@path=\pgfutil@empty
596 \pgfsyssoftpath@setcurrentpath{\tqft@bdry@path}
597      \tikz@mode@fillfalse%
598      \tikz@mode@drawfalse%
599      \let\tikz@mode=\pgfutil@empty
600      \let\tikz@options=\pgfutil@empty
601      \tqftset{boundary upper style contents}
602      \tikz@mode
603      \tikz@options
604      \advance\tqft@xa by -\tqft@s
605      \advance\tqft@xa by \tqft@w
```

Do we have any incoming boundary components at all?

```
606      \ifnum\tqft@incoming>0
```

Yes, so iterate over them

```
607      \foreach \tqft@k in {1,...,\tqft@incoming} {
608      \advance\tqft@xa by \tqft@k\tqft@s
609      \pgfpathmoveto{\pgfqpoint{\tqft@xa}{\tqft@h}}
610      \pgfpatharc{0}{\pgf@tqft@upper180}{\tqft@width and \tqft@depth}
611   \ifx\tikz@fig@name\pgfutil@empty
612   \else
```

```
613  {
614      \advance\tqft@xa by -\tqft@width
615      \pgftransformshift{\pgfqpoint{\tqft@xa}{\tqft@h}}
616      \tqft@within@nodetrue
617      \pgfsyssoftpath@getcurrentpath{\tqft@bdry@path}
618      \pgfsyssoftpath@setcurrentpath{\tqft@bdry@node@path}
619      \pgfnode{tqft boundary circle}{centre}{}{\tikz@fig@name\space incoming \tqft@k}{}
620      \pgfsyssoftpath@getcurrentpath{\tqft@bdry@node@path}
621      \pgfsyssoftpath@setcurrentpath{\tqft@bdry@path}
622  }
623  \fi
624      }
625      \fi
```

Now iterate over the outgoing boundary components, if we have any

```
626      \ifnum\tqft@outgoing>0
627      \pgfmathsetlength{\tqft@xa}{\tqft@xa + (\tqft@outgoing + \tqft@offset + 1) * \tqft@sepa
628      \foreach \tqft@k in {1,...,\tqft@outgoing} {
629      \advance\tqft@xa by -\tqft@k\tqft@s
630 %    \advance\tqft@xa by \tqft@s
631      \pgfpathmoveto{\pgfqpoint{\tqft@xa}{-\tqft@h}}
632      \pgfpatharc{0}{\pgf@tqft@upper180}{\tqft@width and \tqft@depth}
633  \ifx\tikz@fig@name\pgfutil@empty
634  \else
635  {
636      \pgfmathtruncatemacro{\tqft@l}{\tqft@outgoing + 1 - \tqft@k}
637      \advance\tqft@xa by -\tqft@width
638      \pgftransformshift{\pgfqpoint{\tqft@xa}{-\tqft@h}}
639      \tqft@within@nodetrue
640      \pgfsyssoftpath@getcurrentpath{\tqft@bdry@path}
641      \pgfsyssoftpath@setcurrentpath{\tqft@bdry@node@path}
642      \pgfnode{tqft boundary circle}{centre}{}{\tikz@fig@name\space outgoing \tqft@l}{}
643      \pgfsyssoftpath@getcurrentpath{\tqft@bdry@node@path}
644      \pgfsyssoftpath@setcurrentpath{\tqft@bdry@path}
645  }
646  \fi
647      }
648      \fi
649      \edef\tikz@temp{\noexpand\pgfusepath{%
650          \iftikz@mode@fill fill,\fi%
651          \iftikz@mode@draw draw\fi%
652      }}%
653      \tikz@temp
654      }
655  }
656 }
```

boundary circle shape This is a the shape of the boundary circles

```
657 \pgfdeclareshape{tqft boundary circle}{
```

Now we save our dimensions: height, separation, and the radii of the boundary circles

```
658  \saveddimen{\tqft@height}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/cobordism height}}
659  \saveddimen{\tqft@separation}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/boundary separation}}
```

16

```
660    \saveddimen{\tqft@width}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/circle width}}
661    \saveddimen{\tqft@depth}{\pgf@x=\pgfkeysvalueof{/pgf/tqft/circle depth}}
```

For the externally available anchors, we need to save the declared transformation; we save the actual transformation, not the macro that points to it. If we're called within the main cobordism shape, the transformation is already applied so we ignore it.

```
662    \savedmacro{\tqft@transformation}{%
663    \iftqft@within@node
664    \let\tqft@transformation=\pgfutil@empty
665    \else
666    \pgfkeysgetvalue{/pgf/tqft/flow transformation}{\tqft@transformation}
667    \fi
668 }

669    \savedanchor{\tqft@centre}{%
670    \pgfpointorigin}
```

For completeness, we record the size of the text box (not that we expect any text, but you never know)

```
671 \savedanchor{\tqft@textsize}{%
672    \pgf@y=-.5\ht\pgfnodeparttextbox%
673    \pgf@x=-.5\wd\pgfnodeparttextbox%
674 }
```

These are our externally available anchors

```
675    \anchor{centre}{\tqft@centre}
676    \anchor{center}{\tqft@centre}
677 \anchor{text}{
678    \tqft@textsize
679 }
680    \anchor{next}{%
681    \tqft@process{\tqft@separation}{0pt}}%

682    \anchor{prior}{%
683    \tqft@process{-\tqft@separation}{0pt}}%

684    \anchor{above}{%
685    \tqft@process{0pt}{\tqft@height}}%

686    \anchor{below}{%
687    \tqft@process{0pt}{-\tqft@height}}%
```

The anchor border is the ellipse, but we need to take into account the possible transformation. (This isn't right if the origin is shifted.) At the moment, '0 degrees' is interpreted in the transformed coordinate system. Should provide a system whereby that can be intepreted in the main coordinate system.

```
688    \anchorborder{
```

This next `\pgf@process` makes the angles absolute. Comment it out to make the angles relative.

```
689    \tqft@process{\the\pgf@x}{\the\pgf@y}
690    \edef\tqft@marshal{%
691       \noexpand\pgfpointborderellipse
692       {\noexpand\pgfqpoint{\the\pgf@x}{\the\pgf@y}}
693       {\noexpand\pgfqpoint{\tqft@width}{\tqft@depth}}
694    }%
```

17

```
695    \tqft@marshal
696    \tqft@process{\the\pgf@x}{\the\pgf@y}
697 }
```

Now we define the background path. This is the upper part of the cobordism.

```
698    \backgroundpath{
```

Apply the internal transformation if we're not within a node

```
699        \let\tikz@transform=\pgfutil@empty
700        \expandafter\tikzset\expandafter{\tqft@transformation}
701        \tikz@transform
```

Draw the boundary circle

```
702    \pgfpathellipse{\pgfqpoint{0pt}{0pt}}{\pgfqpoint{\tqft@width}{0pt}}{\pgfqpoint{0pt}{\tqft@
703    }
```

We draw the upper and lower arcs again with the appropriate styles

```
704    \beforebackgroundpath{
705    \iftqft@within@node
706    \else
707      \tikz@mode@fillfalse%
708      \tikz@mode@drawfalse%
709      \let\tikz@mode=\pgfutil@empty
710      \let\tikz@options=\pgfutil@empty
711      {
712      \pgfsys@beginscope
713        \tqftset{boundary lower style contents}
714        \tikz@mode
715        \tikz@options
716        \pgfpathmoveto{\pgfqpoint{\tqft@width}{0pt}}
717        \pgfpatharc{0}{\pgf@tqft@lower180}{\tqft@width and \tqft@depth}
718        \edef\tikz@temp{\noexpand\pgfusepath{%
719            \iftikz@mode@fill fill,\fi%
720            \iftikz@mode@draw draw\fi%
721        }}%
722        \tikz@temp
723        \pgfsys@endscope
724      }
725      {
726        \pgfsys@beginscope
727        \tqftset{boundary upper style contents}
728        \tikz@mode
729        \tikz@options
730        \pgfpathmoveto{\pgfqpoint{\tqft@width}{0pt}}
731        \pgfpatharc{0}{\pgf@tqft@upper180}{\tqft@width and \tqft@depth}
732        \edef\tikz@temp{\noexpand\pgfusepath{%
733            \iftikz@mode@fill fill,\fi%
734            \iftikz@mode@draw draw\fi%
735        }}%
736        \tikz@temp
737        \pgfsys@endscope
738      }
739    \fi
740    }
741    }
```

2.2 New Version: Picture Shapes

Issue a warning if the pic syntax is not available.

```
742 \ifcsname pgfk@/handlers/.pic/.@cmd\endcsname
743 \else
744 \pgfwarning{This library only works with TikZ 3.0 or later; for earlier versions of TikZ use
745 \fi
```

For the boundaries, we need elliptical node shapes.

```
746 \usetikzlibrary{shapes.geometric}
```

We can view the cobordisms from the *input* or *output* ends, the implementation of the choice is to draw an arc from 0 to 180 or from 0 to -180 so we just need to track minus signs. These macros are for that.

```
747 \def\pgf@tqft@minus{-}
748 \let\pgf@tqft@upper\@empty
749 \let\pgf@tqft@lower\pgf@tqft@minus
```

Should we twist the cobordism?

```
750 \newif\iftqft@twisted
```

Split an anchoring coordinate. The y–value is simply multiplied by the cobordism height (but pointing downwards, so that 1 is level with the outgoing boundary). The x–value is multiplied by the boundary separation, but is shifted so that at the incoming boundary level, or above, then it is in line with the incoming boundaries and similarly at the outgoing boundary level, or below, it is in line with the outgoing boundaries.

```
751 \def\tqft@split(#1,#2){%
752   \pgfmathsetmacro\tqft@y{#2 * (-\tqft@val{cobordism height})}%
753   \pgfmathsetmacro\tqft@x{(#1 - 1 + max(min(#2,1),0)*\tqft@val{offset}) * \tqft@val{boundary
754   \def\tqft@shift{(\tqft@x pt, \tqft@y pt)}%
755 }%
```

Now we set up all the keys that we'll need in the course of this shape

```
756 \tikzset{
```

Fix for the fact that the `alias` key doesn't use the prefix and suffix.

```
757   pic alias/.code={%
758     \tikz@fig@mustbenamed
759     \expandafter\def\expandafter\tikz@alias\expandafter{\tikz@alias\pgfnodealias{\tikz@pp@
760   },
```

This key is our basic installer key, setting the pic and putting us in the right key family.

```
761   tqft/.style={%
762     pic type=cobordism,
763     every tqft/.try,
764     tqft/.cd,
765   },
```

This deals with unknown keys, passing them on to TikZ.

```
766   tqft/.unknown/.code={%
767     \let\tqft@searchname=\pgfkeyscurrentname%
768     \pgfkeysalso{%
769       /tikz/\tqft@searchname={#1}
770     }
771   },
```

19

Let's play happy families!

```
772   tqft/.cd,
```

These set our number of boundary components and genus.

```
773   incoming boundary components/.initial=5,
774   outgoing boundary components/.initial=4,
775   skip incoming boundary components/.initial={},
776   skip outgoing boundary components/.initial={},
777   genus/.initial = 0,
```

This is the "horizontal" offset of the first outgoing component from the first incoming one.

```
778   offset/.initial=0,
```

This is the "vertical" separation between boundary components.

```
779   cobordism height/.initial=2cm,
```

This is the "horizontal" separation between boundary components.

```
780   boundary separation/.initial=2cm,
```

These are the "horizontal" and "vertical" radii, respectively, of the boundary components.

```
781   circle x radius/.initial=10pt,
782   circle y radius/.initial=5pt,
```

These control the direction from which we view the cobordism.

```
783   view from/.is choice,
784   view from/incoming/.code={%
785     \let\pgf@tqft@upper\pgf@tqft@minus
786     \let\pgf@tqft@lower\@empty
787   },
788   view from/outgoing/.code={%
789   \let\pgf@tqft@lower\pgf@tqft@minus
790     \let\pgf@tqft@upper\@empty
791   },
```

Should we twist the cobordism?

```
792   twisted/.is if=tqft@twisted,
```

We simulate node placement using the following key.

```
793   anchor/.initial = none,
```

The next set of keys define some default shapes.

```
794   pair of pants/.style={
795     /tikz/tqft,
796     incoming boundary components=1,
797     outgoing boundary components=2,
798     offset=-.5
799   },
800   /tikz/tqft pair of pants/.style={
801     /tikz/tqft/pair of pants,
802   },
803   reverse pair of pants/.style={
804     /tikz/tqft,
805     incoming boundary components=2,
806     outgoing boundary components=1,
807     offset=.5
```

```
808   },
809   /tikz/tqft reverse pair of pants/.style={
810     /tikz/tqft/reverse pair of pants,
811   },
812   cylinder to prior/.style={
813     /tikz/tqft,
814     incoming boundary components=1,
815     outgoing boundary components=1,
816     offset=-.5
817   },
818   /tikz/tqft cylinder to prior/.style={
819     /tikz/tqft/cylinder to prior,
820   },
821   cylinder to next/.style={
822     /tikz/tqft,
823     incoming boundary components=1,
824     outgoing boundary components=1,
825     offset=.5
826   },
827   /tikz/tqft cylinder to next/.style={
828     /tikz/tqft/cylinder to next,
829   },
830   cylinder/.style={
831     /tikz/tqft,
832     incoming boundary components=1,
833     outgoing boundary components=1
834   },
835   /tikz/tqft cylinder/.style={
836     /tikz/tqft/cylinder,
837   },
838   cup/.style={
839     /tikz/tqft,
840     incoming boundary components=1,
841     outgoing boundary components=0
842   },
843   /tikz/tqft cup/.style={
844     /tikz/tqft/cup,
845   },
846   cap/.style={
847     /tikz/tqft,
848     incoming boundary components=0,
849     outgoing boundary components=1
850   },
851   /tikz/tqft cap/.style={
852     /tikz/tqft/cap,
853   },
854 }
```

This is a little helper macro for getting the values of tqft keys.

```
855 \def\tqft@val#1{\pgfkeysvalueof{/tikz/tqft/#1}}
```

Now we define the code for the actual cobordism shape.

```
856 \tikzset{
857   cobordism/.pic={
```

Defining the cobordism paths. This holds the full boundary path of the cobordism shape.

```
858        \gdef\tqft@fullpath{}%
```

This is a list of the edge pieces without the boundary circles.

```
859        \global\let\tqft@blist\pgfutil@gobble%
```

This punches the holes (if there are any) in the cobordism shape.

```
860        \gdef\tqft@gclip{}%
```

This is a list of the paths for drawing the holes.

```
861        \global\let\tqft@glist\pgfutil@gobble%
```

This collects any coordinates that are to be defined (it appears to be difficult to define them as we go along).

```
862        \global\let\tqft@clist\pgfutil@gobble%
```

This collects any coordinates that can be used to shift the shape that aren't to be defined using \tqft@clist.

```
863        \global\let\tqft@alist\pgfutil@gobble
```

These will be lists of the boundary components, divided into sets as to whether or not they are rendered. For the outgoing ones, we need two lists because they are rendered in the opposite order to how they are labelled.

```
864        \global\let\tqft@ibdrylist=\pgfutil@gobble
865        \global\let\tqft@cibdrylist=\pgfutil@gobble
866        \global\let\tqft@obdrylist=\pgfutil@gobble
867        \global\let\tqft@cobdrylist=\pgfutil@gobble
868        \global\let\tqft@robdrylist=\pgfutil@gobble
869        \global\let\tqft@rcobdrylist=\pgfutil@gobble
```

Is the cobordism twisted? If so, we need to reverse the order of the outgoing boundary components.

```
870      \iftqft@twisted
871      \pgfmathsetmacro\tqft@outgoing@end{0}%
872      \pgfmathsetmacro\tqft@outgoing@dir{-1}%
873      \else
874      \pgfmathsetmacro\tqft@outgoing@end{1}%
875      \pgfmathsetmacro\tqft@outgoing@dir{1}%
876      \fi
```

The first stage is to iterate over the incoming boundary components (if there are any), building up the various paths.

```
877        \ifnum\tqft@val{incoming boundary components}>0\relax
```

We have some so draw the half circle for the first component. Note that we use \pgf@tqft@upper to flip the sign of the start angle depending on the `view from` setting.

```
878        \xdef\tqft@fullpath{%
879          \tqft@fullpath
880          (-\tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180, end angle=0, x r
881        }%
```

And add the centre to the list for available shifts.

```
882        \xdef\tqft@alist{%
883          \tqft@alist,-incoming boundary 1/{(0,0)},-incoming boundary/{(0,0)}%
884        }%
```

If there are more than one then for each subsequent one we add the curve between
them and the corresponding arc of the boundary circle.

```
885      \ifnum\tqft@val{incoming boundary components}>1\relax
886      \foreach \k in {2,...,\tqft@val{incoming boundary components}} {
887        \edef\tqft@temp{\noexpand\pgfutil@in@{,\k,}{,\tqft@val{skip incoming boundary componen
888        \tqft@temp
889        \ifpgfutil@in@
890        \xdef\tqft@cibdrylist{\tqft@cibdrylist,\k}
891        \else
892        \xdef\tqft@ibdrylist{\tqft@ibdrylist,\k}
893        \fi
894      }
895      \ifx\tqft@ibdrylist\pgfutil@gobble
896      \else
897      \foreach \k [
898        remember=\k as \kmo (initially 1),
899          evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation} -
    \tqft@val{circle x radius},
900      ] in \tqft@ibdrylist {
901        \pgfmathsetmacro\xppos{(\kmo - 1)*\tqft@val{boundary separation} + \tqft@val{circle x
902        \pgfmathsetmacro\cpos{(\xpos + \xppos)/2}
```

Add the curve and the arc.

```
903          \xdef\tqft@fullpath{%
904            \tqft@fullpath
905            .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-\tqft@val{cobordism heig
906          }%
```

But for the edge path, just add the curve to the list.

```
907          \xdef\tqft@blist{%
908            \tqft@blist,incoming boundary \k/incoming/{%
909          (\xppos pt,0) .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-
    \tqft@val{cobordism height}/3) .. (\xpos pt,0)}%
910          }%
```

We add a coordinate at the midpoint of the curve.

```
911          \xdef\tqft@clist{%
912            \tqft@clist,-between incoming \kmo\space and \k/{(\cpos pt,-\tqft@val{cobordism heig
913          }%
```

And add the centre to the list for available shifts.

```
914          \xdef\tqft@alist{%
915            \tqft@alist,-incoming boundary \k/{(\kmo * \tqft@val{boundary separation},0)}%
916          }%
917      }%
918      \fi
919        \fi
```

We're at the edge of the last incoming boundary component. What we do now
depends on whether or not there are outgoing boundary components.

```
920        \ifnum\tqft@val{outgoing boundary components}>0\relax
```

There are.

We start by adding a curve from the end of the last incoming to the last
outgoing component to the full path,

```
921        \pgfmathsetmacro\xppos{%
```

```
922          (\tqft@outgoing@end * (\tqft@val{outgoing boundary components} -1)
923          + \tqft@val{offset}) * \tqft@val{boundary separation}
924          + \tqft@outgoing@dir * \tqft@val{circle x radius}}%
925          \pgfmathsetmacro\tqft@ht{abs(\tqft@val{incoming boundary components} - \tqft@val{outgo
926          \pgfmathsetmacro\tqft@ht{1/3 + 2/3*\tqft@ht/(\tqft@ht + 1)}%
927          \xdef\tqft@fullpath{%
928            \tqft@fullpath
929            .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{co
   \tqft@val{cobordism height})
930          }%
```
and the edge path.
```
931          \xdef\tqft@blist{%
932            \tqft@blist,between last incoming and last outgoing/incoming and outgoing/{%
933            (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@va
934            .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{co
   \tqft@val{cobordism height})}%
935          }%
```
In addition, we add a coordinate at the midpoint.
```
936          \pgfmathsetmacro\xppos{(\xppos + (\tqft@val{incoming boundary components} -
   1) * \tqft@val{boundary separation} +\tqft@val{circle x radius})/2}%
937          \xdef\tqft@clist{%
938            \tqft@clist,-between last incoming and last outgoing/{(\xppos pt,-
   \tqft@val{cobordism height}/2)}%
939          }%
940          \else
```
There aren't any outgoing boundary components so we loop back to the start. We
adjust the height of the control points to take into account the overall width.
```
941          \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{incoming boundary components} - 1)/\tqft
942          \xdef\tqft@fullpath{%
943            \tqft@fullpath
944            .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-
   \tqft@ht*\tqft@val{cobordism height}) .. (-\tqft@val{circle x radius},0)
945          }%
```
Same for the edge path.
```
946          \xdef\tqft@blist{%
947            \tqft@blist,between first and last incoming/incoming and outgoing/{%
948            (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@va
949            .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-
   \tqft@ht*\tqft@val{cobordism height}) .. (-\tqft@val{circle x radius},0)}
950          }%
```
Add a coordinate at the midpoint.
```
951          \pgfmathsetmacro\xppos{(\tqft@val{incoming boundary components} -1) * \tqft@val{bounda
952          \xdef\tqft@clist{%
953            \tqft@clist,-between first and last incoming/{(\xppos pt,-\tqft@ht*\tqft@val{cobordi
   between first incoming and last incoming/{(\xppos pt,-\tqft@ht*\tqft@val{cobordism height}*3
954          }%
955          \fi
956          \else
```
There weren't any incoming boundary components, so we test to see if there were
any outgoing ones and move to the start of them.

```
957        \ifnum\tqft@val{outgoing boundary components}>0\relax
958          \pgfmathsetmacro\xppos{( \tqft@outgoing@end * (\tqft@val{outgoing boundary component
    1)+\tqft@val{offset}) * \tqft@val{boundary separation} + \tqft@outgoing@dir * \tqft@val{circ
```

Add a move to the full path,

```
959        \xdef\tqft@fullpath{%
960          \tqft@fullpath
961          (\xppos pt, -\tqft@val{cobordism height})
962        }%
963      \fi
964      \fi
```

We're done with the incoming boundary components, now we're set up for the
outgoing ones. However we got there, if we have outgoing boundary components
then we're now located at the start of them, although we might be counting
backwards .

```
965        \ifnum\tqft@val{outgoing boundary components}>0\relax
966      \pgfmathsetmacro\xppos{%
967        ( \tqft@outgoing@end * (\tqft@val{outgoing boundary components} -1)
968        + \tqft@val{offset}) * \tqft@val{boundary separation}
969        - \tqft@outgoing@dir * \tqft@val{circle x radius}}%
```

Draw the arc for the first (well, last actually) boundary component.

```
970        \xdef\tqft@fullpath{%
971          \tqft@fullpath
972          arc[end angle=\pgf@tqft@upper180, start angle=0, x radius=\tqft@outgoing@dir * \tqft
973        }%
```

And add the centre to the list for available shifts.

```
974        \xdef\tqft@alist{%
975          \tqft@alist,-outgoing boundary \tqft@val{outgoing boundary components}/{(\xppos pt +
    \tqft@val{cobordism height})},-outgoing boundary/{(\tqft@val{offset}*\tqft@val{boundary sepa
    \tqft@val{cobordism height})}}%
976        }%
```

Do we have more than one boundary component?

```
977        \ifnum\tqft@val{outgoing boundary components}>1\relax
```

Yes, so add a curve and arc for each.

```
978      \foreach \k [evaluate=\k as \ok using int(\tqft@outgoing@end * (\tqft@val{outgoing bound
979        \edef\tqft@temp{\noexpand\pgfutil@in@{,\ok,}{,\tqft@val{skip outgoing boundary compone
980        \tqft@temp
981        \ifpgfutil@in@
982        \xdef\tqft@cobdrylist{\tqft@cobdrylist,\k}
983        \else
984        \xdef\tqft@obdrylist{\tqft@obdrylist,\k}
985        \fi
986      }
987      \ifx\tqft@obdrylist\pgfutil@gobble
988      \else
989      \foreach \k [
990        remember=\k as \kmo (initially 1),
991          evaluate=\k as \xpos using ( \tqft@outgoing@end * (\tqft@val{outgoing boundary compo
992      ] in \tqft@obdrylist {
993        \pgfmathsetmacro\xppos{(\tqft@outgoing@end * (\tqft@val{outgoing boundary components}
994          \pgfmathsetmacro\cpos{(\xpos + \xppos)/2}
```

```
995        \pgfmathsetmacro\nk{int(\tqft@val{outgoing boundary components} - \k + 1)}
996        \pgfmathsetmacro\nkpo{int(\tqft@val{outgoing boundary components} - \kmo + 1)}
```

Both are added to the full path.

```
997        \xdef\tqft@fullpath{%
998          \tqft@fullpath
999          .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height
   \tqft@val{cobordism height}) arc[end angle=\pgf@tqft@upper180, start angle=0, x radius=\tqft
1000       }%
```

Just the arc for the edge paths.

```
1001       \xdef\tqft@blist{%
1002         \tqft@blist,between outgoing \nk\space and \nkpo/outgoing/{%
1003  (\xppos pt,-\tqft@val{cobordism height})
1004           .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height
   \tqft@val{cobordism height}) ++(-2*\tqft@val{circle x radius},0)}%
1005       }%
```

And a coordinate at the midpoint.

```
1006       \xdef\tqft@clist{%
1007         \tqft@clist,-between outgoing \nk\space and \nkpo/{(\cpos pt,-3*\tqft@val{cobordism
1008       }%
```

And add the centre to the list for available shifts.

```
1009       \xdef\tqft@alist{%
1010         \tqft@alist,-outgoing boundary \nk/{(\xpos pt - \tqft@val{circle x radius},-
   \tqft@val{cobordism height})}%
1011       }%
1012       }%
1013     \fi
1014     \fi
```

Now we're at the end of the outgoing boundary components (well, the start actually). What we do now depends on whether or not there are any incoming boundary components.

```
1015     \ifnum\tqft@val{incoming boundary components}>0\relax
```

There are, so we draw the path back up.

```
1016     \pgfmathsetmacro\tqft@ht{1/3 + 2/3*abs(\tqft@val{offset})/(abs(\tqft@val{offset}) + 1)
1017     \xdef\tqft@fullpath{%
1018       \tqft@fullpath
1019       .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{co
   \tqft@val{circle x radius},0)
1020       }%
```

And the edge path does the same.

```
1021     \xdef\tqft@blist{%
1022       \tqft@blist,between first incoming and first outgoing/incoming and outgoing/{%
1023  ({ ( (1 - \tqft@outgoing@end) * (\tqft@val{outgoing boundary components} - 1) + \tqft@va
   \tqft@val{cobordism height})
1024         .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{co
   \tqft@val{circle x radius},0)}%
1025       }%
```

Add a coordinate at the midpoint.

```
1026     \xdef\tqft@clist{%
```

```
1027          \tqft@clist,-between first incoming and first outgoing/{({ ( (1 - \tqft@outgoing@end
    \tqft@val{cobordism height}/2)}%
1028          }%
1029          \else
```

No incoming boundary components so loop back to the other end of the outgoing
boundary components.

```
1030          \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset})
1031          \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{outgoing boundary components} - 1)/\tqft
```

Full path.

```
1032          \xdef\tqft@fullpath{%
1033            \tqft@fullpath
1034            .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
    \tqft@val{cobordism height})
1035          }%
```

Edge path.

```
1036          \xdef\tqft@blist{%
1037            \tqft@blist,between first and last outgoing/incoming and outgoing/{%
1038          (\tqft@val{offset} * \tqft@val{boundary separation} - \tqft@val{circle x radius},-
    \tqft@val{cobordism height})
1039            .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
    \tqft@val{cobordism height})}%
1040          }%
```

Add a coordinate at the midpoint.

```
1041          \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components}/2 + \tqft@val{offset}
    1/2) * \tqft@val{boundary separation}}%
1042          \pgfmathsetmacro\tqft@ht{1 -\tqft@ht*3/4}%
1043          \xdef\tqft@clist{%
1044            \tqft@clist,-between first and last outgoing/{(\xppos pt,-\tqft@ht*\tqft@val{cobordi
    between first outgoing and last outgoing/{(\xppos pt,-\tqft@ht*\tqft@val{cobordism height})}
1045          }%
1046          \fi
1047          \fi
```

Now we define the clip path for the genus holes. We start with a big rectangle
that *ought* to be big enough to contain the whole shape. We start with the top
left corner.

```
1048          \pgfmathsetmacro\xpos{%
1049            (
1050            \tqft@val{outgoing boundary components} > 0 ?
1051            (
1052            \tqft@val{incoming boundary components} > 0 ?
1053            min(0,\tqft@val{offset}) : \tqft@val{offset}
1054            ) : 0
1055            )
1056            *\tqft@val{boundary separation} - 2*\tqft@val{circle x radius}}%
1057          \xdef\tqft@gclip{(\xpos pt,2*\tqft@val{circle y radius}) rectangle }%
```

Now the bottom right.

```
1058          \pgfmathsetmacro\xpos{%
1059            ((
1060            \tqft@val{outgoing boundary components} > 0 ?
1061            (
```

```
1062          \tqft@val{incoming boundary components} > 0 ?
1063          max(\tqft@val{incoming boundary components},\tqft@val{outgoing boundary components}
1064          ) : \tqft@val{incoming boundary components}
1065          )-1)
1066          *\tqft@val{boundary separation} + 2*\tqft@val{circle x radius}}%
```

Together, these make a rectangle.

```
1067          \xdef\tqft@gclip{\tqft@gclip (\xpos pt,-\tqft@val{cobordism height} - 2*\tqft@val{circ
```

Are there any holes?

```
1068          \ifnum\tqft@val{genus}>0\relax
```

Yes, so first we need to figure out where to place them. We work out the left-hand
edge of the cobordism.

```
1069          \pgfmathsetmacro\xpos{%
1070          (
1071          \tqft@val{outgoing boundary components} > 0 ?
1072          (
1073          \tqft@val{incoming boundary components} > 0 ?
1074          \tqft@val{offset}/2 : \tqft@val{offset}
1075          ) : 0
1076          )
1077          *\tqft@val{boundary separation} - \tqft@val{circle x radius}}%
```

Work out the height that the holes should be punched at.

```
1078          \pgfmathsetmacro\ypos{%
1079          (
1080          \tqft@val{outgoing boundary components} > 0 ?
1081          (
1082          \tqft@val{incoming boundary components} > 0 ?
1083          -\tqft@val{cobordism height}/2 : -1 + \tqft@val{cobordism height}/3
1084          ) : - \tqft@val{cobordism height}/3
1085          )}%
```

Start our clip path at this point

```
1086          \xdef\tqft@gclip{%
1087          \tqft@gclip
1088          (\xpos pt,\ypos pt)
1089          }%
```

Now work out the width of the cobordism, in units of circle half-widths. This may
not be very accurate if there aren't any boundary components of a given type.

```
1090          \pgfmathsetmacro\gsize{%
1091          ((
1092          \tqft@val{outgoing boundary components} > 0 ?
1093          (
1094          \tqft@val{incoming boundary components} > 0 ?
1095          (\tqft@val{incoming boundary components} + \tqft@val{outgoing boundary components})/
1096          ) : \tqft@val{incoming boundary components}
1097          )-1)
1098          *\tqft@val{boundary separation}/\tqft@val{circle x radius} + 2}%
```

Each hole should take up three half-widths, but we want a little extra on the edges
so the total number of half-widths we want is $3g + 1$. Do we need to scale down
the holes (we never scale up)? If so, \gscale holds the overall scale factor and
\gxscale and gyscale are the resulting horizontal and vertical measurements.
The baseline is the size of the boundary circles.

```
1099          \pgfmathsetmacro\gscale{min(1,\gsize/(3*\tqft@val{genus}+1))}%
1100          \pgfmathsetmacro\gyscale{\tqft@val{circle y radius}*\gscale*.707}%
1101          \pgfmathsetmacro\gxscale{\tqft@val{circle x radius}*\gscale}%
```

Each hole should take up 2 half widths, modulo scaling, so the total width used by the holes is $2gs$ leaving $w - 2gs$ left for the gaps which is divided in to $g + 1$ lots.

```
1102          \pgfmathsetmacro\gsep{(((\gsize - 2*\tqft@val{genus}*\gscale)/(\tqft@val{genus} + 1)*\t
```

We shift in by half of one unit of excess separation.

```
1103          \xdef\tqft@gclip{%
1104            \tqft@gclip
1105            ++(\gsep/2 pt,0)
1106          }%
```

Some useful quantities.

```
1107          \pgfmathsetmacro\omrstwo{1 - 1/sqrt(2)}%
1108          \pgfmathsetmacro\sqrtwo{sqrt(2)}%
```

Now we iterate over the holes.

```
1109          \foreach[
1110            evaluate=\k as \kmo using int(2 * \k-1)
1111          ] \k in {1,...,\tqft@val{genus}} {
```

For the clipping path, we just want the bare hole.

```
1112            \xdef\tqft@gclip{%
1113              \tqft@gclip
```

Move in by half an excess separation unit and move to the left-hand extent of the hole.

```
1114              ++(\gsep/2 pt + \omrstwo*\gxscale pt,0)
```

Now curve up over the hole,

```
1115              .. controls +(\gxscale*\sqrtwo/3 pt,4/3*\gyscale pt) and +(-\gxscale*\sqrtwo/3 pt,
1116              .. ++(\sqrtwo*\gxscale pt,0)
```

and return on the underside.

```
1117              .. controls +(-\gxscale*\sqrtwo/3 pt,-4/3*\gyscale pt) and +(\gxscale*\sqrtwo/3 pt
   4/3*\gyscale pt)
1118              .. ++(-\sqrtwo*\gxscale pt,0)
```

Lastly, move to the right-hand edge of the space taken up by this hole.

```
1119              ++(2*\gxscale pt -\omrstwo*\gxscale pt + \gsep/2 pt,0)
1120            }
```

For the genus *path* we want to add the little "tails" which means that the two curves are different, and we need to take into acount the `view from` direction.

```
1121            \xdef\tqft@glist{%
1122              \tqft@glist,%
1123                hole \k/lower/{%
```

Move to the starting point of the smaller curve and add that.

```
1124                (\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt + \gxscale pt -
   \omrstwo*\gxscale pt,\ypos pt)
1125                .. controls +(-\gxscale pt*\sqrtwo/3,\pgf@tqft@upper4/3*\gyscale pt) and +(\gxscal
1126                .. ++(-\sqrtwo*\gxscale pt,0)},%
1127 % Move to the left-hand corner of the path, with the upper or lower chosen by the \verb+view
1128                hole \k/upper/{(\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt - \gxscale pt,\ypos
```

Add the larger of the two curves.

```
1129            .. controls +(\gxscale pt*2/3,\pgf@tqft@lower8/3*\gyscale pt) and +(-
    \gxscale pt*2/3,\pgf@tqft@lower8/3*\gyscale pt)
1130            .. ++(2*\gxscale pt,0)}%
1131        }%
```

Add a coordinate at the centre of the hole.

```
1132        \xdef\tqft@clist{%
1133          \tqft@clist,-hole \k/{(\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt,\ypos pt)}}%
1134        }%
1135      }%
1136      \fi
```

Now we start to lay out the cobordism Were we given a shift? If so, shift.

```
1137 \gdef\tqft@shift{(0,0)}%
1138 \edef\tqft@anchor{\tqft@val{anchor}}%
1139 \expandafter\pgfutil@in@\expandafter{\expandafter,\expandafter}\expandafter{\tqft@anchor}%
1140 \ifpgfutil@in@
1141 \expandafter\tqft@split\tqft@anchor\relax
1142 \else
1143 \edef\tqft@anchor{-\tqft@val{anchor}}%
1144 \xdef\tqft@alist{\tqft@clist,\tqft@alist}%
1145 \foreach \anchor/\coord in \tqft@alist
1146 {
1147   \ifx\anchor\tqft@anchor\relax
1148   \global\let\tqft@shift\coord
1149   \fi
1150 }%
1151 \fi
1152 \tikz@scan@one@point\pgfutil@firstofone\tqft@shift\relax
1153 \begin{scope}[shift={(-\pgf@x,-\pgf@y)}]
```

At each incoming boundary component we place an elliptical node of the right size.

```
1154 \ifnum\tqft@val{incoming boundary components}>0\relax
1155 \ifx\tqft@ibdrylist\pgfutil@gobble
1156 \xdef\tqft@ibdrylist{1}
1157 \else
1158 \xdef\tqft@ibdrylist{1,\tqft@ibdrylist}
1159 \fi
1160        \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft
1161   \node[
1162     transform shape,
1163          node contents={},
1164          ellipse,
1165          inner sep=0pt,
1166          outer sep=0pt,
1167          minimum width=2*\tqft@val{circle x radius},
1168          minimum height=2*\tqft@val{circle y radius},
1169          at={(\xpos pt,0)},
1170          name=-incoming boundary \k,
1171          /tikz/tqft/every boundary component/.try,
1172          /tikz/tqft/every incoming boundary component/.try,
1173          /tikz/tqft/incoming boundary component \k/.try
1174        ];
```

```
1175        }%
1176 \ifx\tqft@cibdrylist\pgfutil@gobble
1177 \else
1178        \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft
1179   \node[
1180     transform shape,
1181           node contents={},
1182           ellipse,
1183           inner sep=0pt,
1184           outer sep=0pt,
1185           minimum width=2*\tqft@val{circle x radius},
1186           minimum height=2*\tqft@val{circle y radius},
1187           at={(\xpos pt,0)},
1188           name=-incoming boundary \k,
1189           /tikz/tqft/every skipped boundary component/.try,
1190           /tikz/tqft/every skipped incoming boundary component/.try,
1191           /tikz/tqft/skipped incoming boundary component \k/.try,
1192         ];
1193 }%
1194 \fi
```

Add an alias for the first.

```
1195        \path node also[pic alias=-incoming boundary] (-incoming boundary 1);
1196       \fi
```

Same for the outgoing boundary components.

```
1197        \ifnum\tqft@val{outgoing boundary components}>0\relax
1198 \ifx\tqft@obdrylist\pgfutil@gobble
1199 \xdef\tqft@obdrylist{1}
1200 \else
1201 \xdef\tqft@obdrylist{1,\tqft@obdrylist}
1202 \fi
1203 \foreach \k [evaluate=\k as \ok using int(\tqft@outgoing@end * (\tqft@val{outgoing boundary
1204   \xdef\tqft@robdrylist{\tqft@robdrylist,\ok}
1205 }
1206        \foreach[
1207          evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary separation}
1208        ] \k in \tqft@robdrylist {
1209   \node[
1210     transform shape,
1211           node contents={},
1212           ellipse,
1213           inner sep=0pt,
1214           outer sep=0pt,
1215           minimum width=2*\tqft@val{circle x radius},
1216           minimum height=2*\tqft@val{circle y radius},
1217           at={(\xpos pt,-\tqft@val{cobordism height})},
1218           name=-outgoing boundary \k,
1219           /tikz/tqft/every boundary component/.try,
1220           /tikz/tqft/every outgoing boundary component/.try,
1221           /tikz/tqft/outgoing boundary component \k/.try
1222         ];
1223 }%
1224 \ifx\tqft@cobdrylist\pgfutil@gobble
1225 \else
```

```
1226 \foreach \k[evaluate=\k as \ok using int(\tqft@outgoing@end * (\tqft@val{outgoing boundary c
1227  in \tqft@cobdrylist {
1228   \xdef\tqft@rcobdrylist{\tqft@rcobdrylist,\ok}
1229 }
1230        \foreach[
1231          evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary separation}
1232        ] \k in \tqft@rcobdrylist {
1233   \node[
1234     transform shape,
1235          node contents={},
1236          ellipse,
1237          inner sep=0pt,
1238          outer sep=0pt,
1239          minimum width=2*\tqft@val{circle x radius},
1240          minimum height=2*\tqft@val{circle y radius},
1241          at={(\xpos pt,-\tqft@val{cobordism height})},
1242          name=-outgoing boundary \k,
1243          /tikz/tqft/every skipped boundary component/.try,
1244          /tikz/tqft/every skipped outgoing boundary component/.try,
1245          /tikz/tqft/skipped outgoing boundary component \k/.try
1246        ];
1247      }%
1248 \fi
```

Add an alias for the first.

```
1249        \path node also[pic alias=-outgoing boundary] (-outgoing boundary 1);
1250        \fi
```

Now we draw the lower paths of the incoming boundary components.

```
1251        \ifnum\tqft@val{incoming boundary components}>0\relax
1252        \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft
1253          \path[
1254          /tikz/tqft/every lower boundary component/.try,
1255          /tikz/tqft/every incoming lower boundary component/.try,
1256          /tikz/tqft/incoming lower boundary component \k/.try
1257        ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@lower180,end a
1258      }%
1259        \fi
```

Same for the outgoing boundary components.

```
1260        \ifnum\tqft@val{outgoing boundary components}>0\relax
1261        \foreach[
1262          evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1263        ] \k in \tqft@robdrylist {
1264          \path[
1265          /tikz/tqft/every lower boundary component/.try,
1266          /tikz/tqft/every outgoing lower boundary component/.try,
1267          /tikz/tqft/outgoing lower boundary component \k/.try
1268        ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start ang
1269      }%
1270        \fi
```

Full outer path, clipped against the genus holes in case it is filled.

```
1271        \begin{scope}
1272        \path[overlay,clip] \tqft@gclip;
1273        \path[
```

```
1274        /tikz/tqft/cobordism/.try,
1275        pic actions,
1276        /tikz/tqft/cobordism outer path/.try,
1277     ] \tqft@fullpath;
1278     \end{scope}
```

Now we draw the genus path, outside the clip. We view this as part of the full
cobordism path so try to apply the same style as for the full path, but if that is
filled then we turn the fill off. It can be turned back on again using the styles
`cobordism edge` or `genus style`. We also apply the `cobordism edge` style as it
could be thought of as part of the non-boundary edge. Finally, it has its own style
to enable overrides if the other two get confused.

```
1279 \ifx\tqft@glist\pgfutil@gobble
1280 \else
1281 \foreach \tqft@gstyle/\tqft@gside/\tqft@gpath in \tqft@glist {
1282     \path[
1283        /tikz/tqft/cobordism/.try,
1284        pic actions,
1285        fill=none,
1286        shade=none,
1287        /tikz/tqft/cobordism edge/.try,
1288        /tikz/tqft/genus style/.try,
1289        /tikz/tqft/genus \tqft@gside/.try,
1290        /tikz/tqft/\tqft@gstyle/.try,
1291        /tikz/tqft/\tqft@gstyle\space\tqft@gside/.try,
1292     ] \tqft@gpath;
1293 }
1294 \fi
```

Now we redraw the non-boundary paths.

```
1295 \ifx\tqft@blist\pgfutil@gobble
1296 \else
1297 \foreach \tqft@bstyle/\tqft@btype/\tqft@bpath in \tqft@blist {
1298     \path[
1299        /tikz/tqft/cobordism edge/.try,
1300        /tikz/tqft/cobordism outer edge/.try,
1301        /tikz/tqft/between \tqft@btype/.try,
1302        /tikz/tqft/\tqft@bstyle/.try,
1303     ] \tqft@bpath;
1304 }
1305 \fi
```

There were various coordinates that we wanted to define but couldn't. Here, we
put those in place.

```
1306        \ifx\tqft@clist\pgfutil@gobble
1307        \else
1308        \foreach \name/\coord in \tqft@clist {
1309          \path \coord node[coordinate,node contents={},name=\name];
1310        }
1311        \fi
```

The last task is to draw the upper paths of the boundary components. First,
incoming.

```
1312        \ifnum\tqft@val{incoming boundary components}>0\relax
1313        \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft
```

```
1314        \path[
1315          /tikz/tqft/every upper boundary component/.try,
1316          /tikz/tqft/every incoming upper boundary component/.try,
1317          /tikz/tqft/incoming upper boundary component \k/.try
1318        ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180,end a
1319      }
1320    \fi
```

Next, outgoing.

```
1321    \ifnum\tqft@val{outgoing boundary components}>0\relax
1322    \foreach[
1323      evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1324    ] \k in \tqft@robdrylist {
1325      \path[
1326        /tikz/tqft/every upper boundary component/.try,
1327        /tikz/tqft/every outgoing upper boundary component/.try,
1328        /tikz/tqft/outgoing upper boundary component \k/.try
1329      ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start ang
1330    }
1331    \fi
1332 \end{scope}
```

We're done! Phew.

```
1333  }
1334 }
```