The **tqft** package: codebase
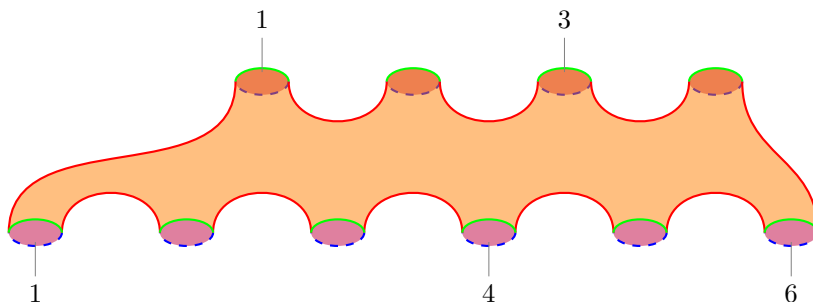
Andrew Stacey

loopspace@mathforge.org

June 1, 2017

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/tqf
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 boundary
197   \savedmacro{\tqft@outgoing}{\edef\tqft@outgoing{\pgfkeysvalueof{/pgf/tqft/outgoing boundary
```

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%
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@count
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) * \noe
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@count
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) * \noe
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@count
351      \expandafter\xdef\csname pgf@anchor@tqft cobordism@after incoming boundary\space\the\c@pgf
352    \noexpand\tqft@start@incoming
353    \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y +.75 * \noexpand\tqft@cont
354    \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \no
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@count
367      \expandafter\xdef\csname pgf@anchor@tqft cobordism@after outgoing boundary\space\the\c@pgf
368    \noexpand\tqft@start@outgoing
369    \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y -.75 * \noexpand\tqft@cont
370    \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \no
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}}{\pgfqpoint
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(\tqf
407     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoin
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@separa
```

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

11

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}}{\pgfqpoin
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@separat
437       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqf
438       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoi
439      \else
440       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqf
441       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@ch}}{\pgfqpo
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{0
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@separa
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}}{\pgfqpoint
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
```

13

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@separa
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
```

14

```
526        \let\tikz@options=\pgfutil@empty
527        \tqftset{cobordism style contents}
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}}{\pgfqpoint+
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(\tqf
547     \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
548     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoin
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@separa
```

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(\tqf
557     \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
558     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@ch}}{\pgfqpoi
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}}{\pgfqpoin
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@separat
580       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqf
581       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoi
582      \else
583       \pgfmathsetlength{\tqft@ch}{min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqf
584       \pgfpathcurveto{\pgfqpoint{\tqft@xb}{-\tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@ch}}{\pgfqpo
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?

16

```
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@separa
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}}
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    }%
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@de
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        }}%
```

19

```
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 t
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
```

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.

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

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

```
755 \tikzset{
```

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

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

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

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

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

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

Let's play happy families!

```
771   tqft/.cd,
```

These set our number of boundary components and genus.

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

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

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

This is the "vertical" separation between boundary components.

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

This is the "horizontal" separation between boundary components.

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

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

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

These control the direction from which we view the cobordism.

```
782   view from/.is choice,
783   view from/incoming/.code={%
```

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

We simulate node placement using the following key.

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

The next set of keys define some default shapes.

```
792    pair of pants/.style={
793      /tikz/tqft,
794      incoming boundary components=1,
795      outgoing boundary components=2,
796      offset=-.5
797    },
798    /tikz/tqft pair of pants/.style={
799      /tikz/tqft/pair of pants,
800    },
801    reverse pair of pants/.style={
802      /tikz/tqft,
803      incoming boundary components=2,
804      outgoing boundary components=1,
805      offset=.5
806    },
807    /tikz/tqft reverse pair of pants/.style={
808      /tikz/tqft/reverse pair of pants,
809    },
810    cylinder to prior/.style={
811      /tikz/tqft,
812      incoming boundary components=1,
813      outgoing boundary components=1,
814      offset=-.5
815    },
816    /tikz/tqft cylinder to prior/.style={
817      /tikz/tqft/cylinder to prior,
818    },
819    cylinder to next/.style={
820      /tikz/tqft,
821      incoming boundary components=1,
822      outgoing boundary components=1,
823      offset=.5
824    },
825    /tikz/tqft cylinder to next/.style={
826      /tikz/tqft/cylinder to next,
827    },
828    cylinder/.style={
829      /tikz/tqft,
830      incoming boundary components=1,
```

```
831      outgoing boundary components=1
832    },
833    /tikz/tqft cylinder/.style={
834      /tikz/tqft/cylinder,
835    },
836    cup/.style={
837      /tikz/tqft,
838      incoming boundary components=1,
839      outgoing boundary components=0
840    },
841    /tikz/tqft cup/.style={
842      /tikz/tqft/cup,
843    },
844    cap/.style={
845      /tikz/tqft,
846      incoming boundary components=0,
847      outgoing boundary components=1
848    },
849    /tikz/tqft cap/.style={
850      /tikz/tqft/cap,
851    },
852 }
```

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

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

Now we define the code for the actual cobordism shape.

```
854 \tikzset{
855    cobordism/.pic={
```

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

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

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

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

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

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

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

```
859        \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).

```
860        \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`.

```
861        \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 too lists because they are rendered in the opposite order to how they are labelled.

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

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

```
868        \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.

```
869        \xdef\tqft@fullpath{%
870          \tqft@fullpath
871          (-\tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180, end angle=0, x rac
872        }%
```

And add the centre to the list for available shifts.

```
873        \xdef\tqft@alist{%
874          \tqft@alist,-incoming boundary 1/{(0,0)},-incoming boundary/{(0,0)}%
875        }%
```

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.

```
876      \ifnum\tqft@val{incoming boundary components}>1\relax
877      \foreach \k in {2,...,\tqft@val{incoming boundary components}} {
878        \edef\tqft@temp{\noexpand\pgfutil@in@{,\k,}{,\tqft@val{skip incoming boundary components
879        \tqft@temp
880        \ifpgfutil@in@
881        \xdef\tqft@cibdrylist{\tqft@cibdrylist,\k}
882        \else
883        \xdef\tqft@ibdrylist{\tqft@ibdrylist,\k}
884        \fi
885      }
886      \ifx\tqft@ibdrylist\pgfutil@gobble
887      \else
888      \foreach \k [
889        remember=\k as \kmo (initially 1),
890          evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation} -\tqft@val{circle x r
891      ] in \tqft@ibdrylist {
892        \pgfmathsetmacro\xppos{(\kmo - 1)*\tqft@val{boundary separation} + \tqft@val{circle x ra
893        \pgfmathsetmacro\cpos{(\xpos + \xppos)/2}
```

Add the curve and the arc.

```
894          \xdef\tqft@fullpath{%
895            \tqft@fullpath
```

```
896            .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-\tqft@val{cobordism height
897        }%
```

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

```
898        \xdef\tqft@blist{%
899          \tqft@blist,incoming boundary \k/incoming/{%
900        (\xppos pt,0) .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-\tqft@val{cobordi
901        }%
```

We add a coordinate at the midpoint of the curve.

```
902        \xdef\tqft@clist{%
903          \tqft@clist,-between incoming \kmo\space and \k/{(\cpos pt,-\tqft@val{cobordism heigh
904        }%
```

And add the centre to the list for available shifts.

```
905        \xdef\tqft@alist{%
906          \tqft@alist,-incoming boundary \k/{(\kmo * \tqft@val{boundary separation},0)}%
907        }%
908      }%
909      \fi
910      \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.

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

There are, so we add a curve from the end of the last incoming to the last outgoing
component to the full path,

```
912        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset})
913        \pgfmathsetmacro\tqft@ht{abs(\tqft@val{incoming boundary components} - \tqft@val{outgoi
914        \pgfmathsetmacro\tqft@ht{1/3 + 2/3*\tqft@ht/(\tqft@ht + 1)}%
915        \xdef\tqft@fullpath{%
916          \tqft@fullpath
917          .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
918        }%
```

and the edge path.

```
919        \xdef\tqft@blist{%
920          \tqft@blist,between last incoming and last outgoing/incoming and outgoing/{%
921          (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@val-
922          .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
923        }%
```

In addition, we add a coordinate at the midpoint.

```
924        \pgfmathsetmacro\xppos{(\xppos + (\tqft@val{incoming boundary components} -1) * \tqft@va
925        \xdef\tqft@clist{%
926          \tqft@clist,-between last incoming and last outgoing/{(\xppos pt,-\tqft@val{cobordism
927        }%
928      \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.

```
929        \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{incoming boundary components} - 1)/\tqft@v
```

```
930        \xdef\tqft@fullpath{%
931          \tqft@fullpath
932          .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{col
933        }%
```

Same for the edge path.

```
934        \xdef\tqft@blist{%
935          \tqft@blist,between first incoming and last incoming/incoming and outgoing/{%
936          (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@val+
937          .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{col
938        }%
```

Add a coordinate at the midpoint.

```
939        \pgfmathsetmacro\xppos{(\tqft@val{incoming boundary components} -1) * \tqft@val{boundary
940        \xdef\tqft@clist{%
941          \tqft@clist,-between first incoming and last incoming/{(\xppos pt,-\tqft@ht*\tqft@val+
942        }%
943      \fi
944      \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.

```
945      \ifnum\tqft@val{outgoing boundary components}>0\relax
946        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset})
```

Add a move to the full path,

```
947        \xdef\tqft@fullpath{%
948          \tqft@fullpath
949          (\xppos pt, -\tqft@val{cobordism height})
950        }%
951      \fi
952      \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're counting backwards.

```
953      \ifnum\tqft@val{outgoing boundary components}>0\relax
954        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset}) *
```

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

```
955        \xdef\tqft@fullpath{%
956          \tqft@fullpath
957          arc[end angle=\pgf@tqft@upper180, start angle=0, x radius=\tqft@val{circle x radius},
958        }%
```

And add the centre to the list for available shifts.

```
959        \xdef\tqft@alist{%
960          \tqft@alist,-outgoing boundary \tqft@val{outgoing boundary components}/{(\xppos pt + '
961        }%
```

Do we have more than one boundary component?

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

Yes, so add a curve and arc for each.

```
963      \foreach \k [evaluate=\k as \ok using int(\tqft@val{outgoing boundary components} - \k + 1
964        \edef\tqft@temp{\noexpand\pgfutil@in@{,\ok,}{,\tqft@val{skip outgoing boundary component
965        \tqft@temp
966        \ifpgfutil@in@
967        \xdef\tqft@cobdrylist{\tqft@cobdrylist,\k}
968        \else
969        \xdef\tqft@obdrylist{\tqft@obdrylist,\k}
970        \fi
971      }
972      \ifx\tqft@obdrylist\pgfutil@gobble
973      \else
974      \foreach \k [
975        remember=\k as \kmo (initially 1),
976          evaluate=\k as \xpos using (\tqft@val{outgoing boundary components} - \k + \tqft@val{o
977      ] in \tqft@obdrylist {
978        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} - \kmo + \tqft@val{offse
979          \pgfmathsetmacro\cpos{(\xpos + \xppos)/2}
980          \pgfmathsetmacro\nk{int(\tqft@val{outgoing boundary components} - \k + 1)}
981          \pgfmathsetmacro\nkpo{int(\tqft@val{outgoing boundary components} - \kmo + 1)}
```

Both are added to the full path.

```
982            \xdef\tqft@fullpath{%
983              \tqft@fullpath
984                .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height}/
985          }%
```

Just the arc for the edge paths.

```
986            \xdef\tqft@blist{%
987              \tqft@blist,between outgoing \nk\space and \nkpo/outgoing/{%
988  (\xppos pt,-\tqft@val{cobordism height})
989                .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height}/
990          }%
```

And a coordinate at the midpoint.

```
991            \xdef\tqft@clist{%
992              \tqft@clist,-between outgoing \nk\space and \nkpo/{(\cpos pt,-3*\tqft@val{cobordism he
993          }%
```

And add the centre to the list for available shifts.

```
994            \xdef\tqft@alist{%
995              \tqft@alist,-outgoing boundary \nk/{(\xpos pt - \tqft@val{circle x radius},-\tqft@val
996          }%
997          }%
998      \fi
999      \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.

```
1000        \ifnum\tqft@val{incoming boundary components}>0\relax
```

27

There are, so we draw the path back up.

```
1001        \pgfmathsetmacro\tqft@ht{1/3 + 2/3*abs(\tqft@val{offset})/(abs(\tqft@val{offset}) + 1)}%
1002        \xdef\tqft@fullpath{%
1003          \tqft@fullpath
1004          .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cob
1005        }%
```

And the edge path does the same.

```
1006        \xdef\tqft@blist{%
1007          \tqft@blist,between first incoming and first outgoing/incoming and outgoing/{%
1008        (\tqft@val{offset} * \tqft@val{boundary separation} - \tqft@val{circle x radius},-\tqft@va
1009          .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cob
1010        }%
```

Add a coordinate at the midpoint.

```
1011        \xdef\tqft@clist{%
1012          \tqft@clist,-between first incoming and first outgoing/{(\tqft@val{offset}*\tqft@val{b
1013        }%
1014        \else
```

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

```
1015        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset}) *
1016        \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{outgoing boundary components} - 1)/\tqft@v
```

Full path.

```
1017        \xdef\tqft@fullpath{%
1018          \tqft@fullpath
1019          .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cobor
1020        }%
```

Edge path.

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

Add a coordinate at the midpoint.

```
1026        \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components}/2 + \tqft@val{offset} -1
1027        \pgfmathsetmacro\tqft@ht{1 -\tqft@ht*3/4}%
1028        \xdef\tqft@clist{%
1029          \tqft@clist,-between first and last outgoing/{(\xppos pt,-\tqft@ht*\tqft@val{cobordism
1030        }%
1031        \fi
1032        \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.

```
1033        \pgfmathsetmacro\xpos{%
1034          (
```

```
1035          \tqft@val{outgoing boundary components} > 0 ?
1036          (
1037          \tqft@val{incoming boundary components} > 0 ?
1038          min(0,\tqft@val{offset}) : \tqft@val{offset}
1039          ) : 0
1040          )
1041          *\tqft@val{boundary separation} - 2*\tqft@val{circle x radius}}%
1042          \xdef\tqft@gclip{(\xpos pt,2*\tqft@val{circle y radius}) rectangle }%
```

Now the bottom right.

```
1043          \pgfmathsetmacro\xpos{%
1044          ((
1045          \tqft@val{outgoing boundary components} > 0 ?
1046          (
1047          \tqft@val{incoming boundary components} > 0 ?
1048          max(\tqft@val{incoming boundary components},\tqft@val{outgoing boundary components} +
1049          ) : \tqft@val{incoming boundary components}
1050          )-1)
1051          *\tqft@val{boundary separation} + 2*\tqft@val{circle x radius}}%
```

Together, these make a rectangle.

```
1052          \xdef\tqft@gclip{\tqft@gclip (\xpos pt,-\tqft@val{cobordism height} - 2*\tqft@val{circle
```

Are there any holes?

```
1053          \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.

```
1054          \pgfmathsetmacro\xpos{%
1055          (
1056          \tqft@val{outgoing boundary components} > 0 ?
1057          (
1058          \tqft@val{incoming boundary components} > 0 ?
1059          \tqft@val{offset}/2 : \tqft@val{offset}
1060          ) : 0
1061          )
1062          *\tqft@val{boundary separation} - \tqft@val{circle x radius}}%
```

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

```
1063          \pgfmathsetmacro\ypos{%
1064          (
1065          \tqft@val{outgoing boundary components} > 0 ?
1066          (
1067          \tqft@val{incoming boundary components} > 0 ?
1068          -\tqft@val{cobordism height}/2 : -1 + \tqft@val{cobordism height}/3
1069          ) : - \tqft@val{cobordism height}/3
1070          )}%
```

Start our clip path at this point

```
1071          \xdef\tqft@gclip{%
1072          \tqft@gclip
1073          (\xpos pt,\ypos pt)
1074          }%
```

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.

```
1075        \pgfmathsetmacro\gsize{%
1076          ((
1077          \tqft@val{outgoing boundary components} > 0 ?
1078          (
1079          \tqft@val{incoming boundary components} > 0 ?
1080          (\tqft@val{incoming boundary components} + \tqft@val{outgoing boundary components})/2
1081          ) : \tqft@val{incoming boundary components}
1082          )-1)
1083          *\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.

```
1084        \pgfmathsetmacro\gscale{min(1,\gsize/(3*\tqft@val{genus}+1))}%
1085        \pgfmathsetmacro\gyscale{\tqft@val{circle y radius}*\gscale*.707}%
1086        \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.

```
1087        \pgfmathsetmacro\gsep{(((\gsize - 2*\tqft@val{genus}*\gscale)/(\tqft@val{genus} + 1)*\tqf
```

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

```
1088        \xdef\tqft@gclip{%
1089          \tqft@gclip
1090          ++(\gsep/2 pt,0)
1091        }%
```

Some useful quantities.

```
1092        \pgfmathsetmacro\omrstwo{1 - 1/sqrt(2)}%
1093        \pgfmathsetmacro\sqrtwo{sqrt(2)}%
```

Now we iterate over the holes.

```
1094        \foreach[
1095          evaluate=\k as \kmo using int(2 * \k-1)
1096        ] \k in {1,...,\tqft@val{genus}} {
```

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

```
1097          \xdef\tqft@gclip{%
1098            \tqft@gclip
```

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

```
1099            ++(\gsep/2 pt + \omrstwo*\gxscale pt,0)
```

Now curve up over the hole,

```
1100            .. controls +(\gxscale*\sqrtwo/3 pt,4/3*\gyscale pt) and +(-\gxscale*\sqrtwo/3 pt,4/
1101            .. ++(\sqrtwo*\gxscale pt,0)
```

and return on the underside.

```
1102            .. controls +(-\gxscale*\sqrtwo/3 pt,-4/3*\gyscale pt) and +(\gxscale*\sqrtwo/3 pt,-
1103            .. ++(-\sqrtwo*\gxscale pt,0)
```

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

```
1104            ++(2*\gxscale pt -\omrstwo*\gxscale pt + \gsep/2 pt,0)
1105          }
```

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.

```
1106          \xdef\tqft@glist{%
1107            \tqft@glist,%
1108              hole \k/lower/{%
```

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

```
1109                (\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt + \gxscale pt -\omrstwo*\gxscale p
1110              .. controls +(-\gxscale pt*\sqrtwo/3,\pgf@tqft@upper4/3*\gyscale pt) and +(\gxscale
1111              .. ++(-\sqrtwo*\gxscale pt,0)},%
1112 % Move to the left-hand corner of the path, with the upper or lower chosen by the \verb+view f
1113              hole \k/upper/{(\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt - \gxscale pt,\ypos p
```

Add the larger of the two curves.

```
1114              .. controls +(\gxscale pt*2/3,\pgf@tqft@lower8/3*\gyscale pt) and +(-\gxscale pt*2/
1115              .. ++(2*\gxscale pt,0)}%
1116          }%
```

Add a coordinate at the centre of the hole.

```
1117          \xdef\tqft@clist{%
1118            \tqft@clist,-hole \k/{(\xpos pt + \k * \gsep pt +  \kmo * \gxscale pt,\ypos pt)}%
1119          }%
1120        }%
1121      \fi
```

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

```
1122 \gdef\tqft@shift{(0,0)}%
1123 \edef\tqft@anchor{\tqft@val{anchor}}%
1124 \expandafter\pgfutil@in@\expandafter{\expandafter,\expandafter}\expandafter{\tqft@anchor}%
1125 \ifpgfutil@in@
1126 \expandafter\tqft@split\tqft@anchor\relax
1127 \else
1128 \edef\tqft@anchor{-\tqft@val{anchor}}%
1129 \xdef\tqft@alist{\tqft@clist,\tqft@alist}%
1130 \foreach \anchor/\coord in \tqft@alist
1131 {
1132   \ifx\anchor\tqft@anchor\relax
1133   \global\let\tqft@shift\coord
1134   \fi
1135 }%
1136 \fi
1137 \tikz@scan@one@point\pgfutil@firstofone\tqft@shift\relax
1138 \begin{scope}[shift={(-\pgf@x,-\pgf@y)}]
```

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

```
1139 \ifnum\tqft@val{incoming boundary components}>0\relax
1140 \ifx\tqft@ibdrylist\pgfutil@gobble
1141 \xdef\tqft@ibdrylist{1}
1142 \else
1143 \xdef\tqft@ibdrylist{1,\tqft@ibdrylist}
1144 \fi
1145     \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft@:
1146       \node[
1147         node contents={},
1148         ellipse,
1149         inner sep=0pt,
1150         outer sep=0pt,
1151         minimum width=2*\tqft@val{circle x radius},
1152         minimum height=2*\tqft@val{circle y radius},
1153         at={(\xpos pt,0)},
1154         name=-incoming boundary \k,
1155         /tikz/tqft/every boundary component/.try,
1156         /tikz/tqft/every incoming boundary component/.try,
1157         /tikz/tqft/incoming boundary component \k/.try
1158       ];
1159     }%
1160 \ifx\tqft@cibdrylist\pgfutil@gobble
1161 \else
1162     \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft@c
1163       \node[
1164         node contents={},
1165         ellipse,
1166         inner sep=0pt,
1167         outer sep=0pt,
1168         minimum width=2*\tqft@val{circle x radius},
1169         minimum height=2*\tqft@val{circle y radius},
1170         at={(\xpos pt,0)},
1171         name=-incoming boundary \k,
1172         /tikz/tqft/every skipped boundary component/.try,
1173         /tikz/tqft/every skipped incoming boundary component/.try,
1174         /tikz/tqft/skipped incoming boundary component \k/.try,
1175       ];
1176 }%
1177 \fi
```

Add an alias for the first.

```
1178       \path node also[pic alias=-incoming boundary] (-incoming boundary 1);
1179     \fi
```

Same for the outgoing boundary components.

```
1180       \ifnum\tqft@val{outgoing boundary components}>0\relax
1181 \ifx\tqft@obdrylist\pgfutil@gobble
1182 \xdef\tqft@obdrylist{1}
```

```
1183 \else
1184 \xdef\tqft@obdrylist{1,\tqft@obdrylist}
1185 \fi
1186 \foreach \k [evaluate=\k as \ok using int(\tqft@val{outgoing boundary components} - \k + 1)] :
1187   \xdef\tqft@robdrylist{\tqft@robdrylist,\ok}
1188 }
1189       \foreach[
1190         evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary separation}
1191       ] \k in \tqft@robdrylist {
1192           \node[
1193           node contents={},
1194           ellipse,
1195           inner sep=0pt,
1196           outer sep=0pt,
1197           minimum width=2*\tqft@val{circle x radius},
1198           minimum height=2*\tqft@val{circle y radius},
1199           at={(\xpos pt,-\tqft@val{cobordism height})},
1200           name=-outgoing boundary \k,
1201           /tikz/tqft/every boundary component/.try,
1202           /tikz/tqft/every outgoing boundary component/.try,
1203           /tikz/tqft/outgoing boundary component \k/.try
1204         ];
1205 }%
1206 \ifx\tqft@cobdrylist\pgfutil@gobble
1207 \else
1208 \foreach \k [evaluate=\k as \ok using int(\tqft@val{outgoing boundary components} - \k + 1)] :
1209   \xdef\tqft@rcobdrylist{\tqft@rcobdrylist,\ok}
1210 }
1211       \foreach[
1212         evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary separation}
1213       ] \k in \tqft@rcobdrylist {
1214           \node[
1215           node contents={},
1216           ellipse,
1217           inner sep=0pt,
1218           outer sep=0pt,
1219           minimum width=2*\tqft@val{circle x radius},
1220           minimum height=2*\tqft@val{circle y radius},
1221           at={(\xpos pt,-\tqft@val{cobordism height})},
1222           name=-outgoing boundary \k,
1223           /tikz/tqft/every skipped boundary component/.try,
1224           /tikz/tqft/every skipped outgoing boundary component/.try,
1225           /tikz/tqft/skipped outgoing boundary component \k/.try
1226         ];
1227       }%
1228 \fi
```

Add an alias for the first.

```
1229       \path node also[pic alias=-outgoing boundary] (-outgoing boundary 1);
1230       \fi
```

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

```
1231        \ifnum\tqft@val{incoming boundary components}>0\relax
1232        \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft@:
1233          \path[
1234            /tikz/tqft/every lower boundary component/.try,
1235            /tikz/tqft/every incoming lower boundary component/.try,
1236            /tikz/tqft/incoming lower boundary component \k/.try
1237          ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@lower180,end ang
1238        }%
1239        \fi
```

Same for the outgoing boundary components.

```
1240        \ifnum\tqft@val{outgoing boundary components}>0\relax
1241        \foreach[
1242          evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1243        ] \k in \tqft@robdrylist {
1244          \path[
1245            /tikz/tqft/every lower boundary component/.try,
1246            /tikz/tqft/every outgoing lower boundary component/.try,
1247            /tikz/tqft/outgoing lower boundary component \k/.try
1248          ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start angle
1249        }%
1250        \fi
```

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

```
1251        \begin{scope}
1252        \path[overlay,clip] \tqft@gclip;
1253        \path[
1254          /tikz/tqft/cobordism/.try,
1255          pic actions,
1256          /tikz/tqft/cobordism outer path/.try,
1257        ] \tqft@fullpath;
1258        \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.

```
1259 \ifx\tqft@glist\pgfutil@gobble
1260 \else
1261 \foreach \tqft@gstyle/\tqft@gside/\tqft@gpath in \tqft@glist {
1262        \path[
1263          /tikz/tqft/cobordism/.try,
1264          pic actions,
1265          fill=none,
1266          shade=none,
1267          /tikz/tqft/cobordism edge/.try,
1268          /tikz/tqft/genus style/.try,
```

```
1269          /tikz/tqft/genus \tqft@gside/.try,
1270          /tikz/tqft/\tqft@gstyle/.try,
1271          /tikz/tqft/\tqft@gstyle\space\tqft@gside/.try,
1272      ] \tqft@gpath;
1273 }
1274 \fi
```

Now we redraw the non-boundary paths.

```
1275 \ifx\tqft@blist\pgfutil@gobble
1276 \else
1277 \foreach \tqft@bstyle/\tqft@btype/\tqft@bpath in \tqft@blist {
1278      \path[
1279        /tikz/tqft/cobordism edge/.try,
1280        /tikz/tqft/cobordism outer edge/.try,
1281        /tikz/tqft/between \tqft@btype/.try,
1282        /tikz/tqft/\tqft@bstyle/.try,
1283      ] \tqft@bpath;
1284 }
1285 \fi
```

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

```
1286      \ifx\tqft@clist\pgfutil@gobble
1287      \else
1288      \foreach \name/\coord in \tqft@clist {
1289        \path \coord node[coordinate,node contents={},name=\name];
1290      }
1291      \fi
```

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

```
1292      \ifnum\tqft@val{incoming boundary components}>0\relax
1293      \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in \tqft@:
1294        \path[
1295          /tikz/tqft/every upper boundary component/.try,
1296          /tikz/tqft/every incoming upper boundary component/.try,
1297          /tikz/tqft/incoming upper boundary component \k/.try
1298        ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180,end ang
1299      }
1300      \fi
```

Next, outgoing.

```
1301      \ifnum\tqft@val{outgoing boundary components}>0\relax
1302      \foreach[
1303        evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1304      ] \k in \tqft@robdrylist {
1305        \path[
1306          /tikz/tqft/every upper boundary component/.try,
1307          /tikz/tqft/every outgoing upper boundary component/.try,
1308          /tikz/tqft/outgoing upper boundary component \k/.try
1309        ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start angle
```

```
1310          }
1311          \fi
1312 \end{scope}
```

We're done! Phew.

```
1313    }
1314 }
```