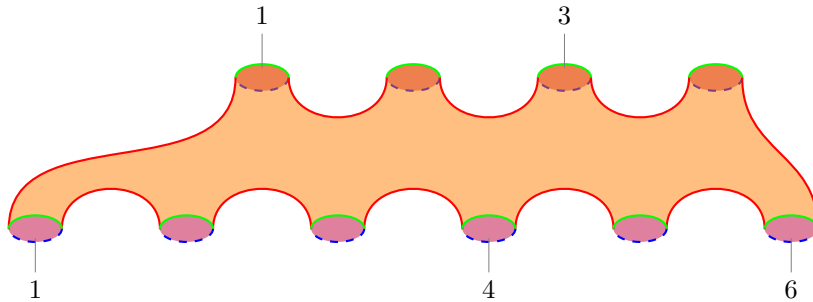


The `tqft` package: codebase

Andrew Stacey
stacey@math.ntnu.no

April 7, 2014



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.stackexchange.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    \pgfpointransformed{\pgfpoint{\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/tqft/flow transformation south/.initial={}},
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@counta
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) * \no
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@counta
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) * \no
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@counta}
351 \expandafter\xdef\csname pgf@anchor@tqft cobordism@after incoming boundary\space\the\c@pgf@counta}
352 \noexpand\tqft@start@incoming
353 \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y + .75 * \noexpand\tqft@center@y}
354 \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \noexpand\tqft@center@x}
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@counta}
367 \expandafter\xdef\csname pgf@anchor@tqft cobordism@after outgoing boundary\space\the\c@pgf@counta}
368 \noexpand\tqft@start@outgoing
369 \noexpand\pgfmathsetlength{\noexpand\pgf@y}{.25 * \noexpand\pgf@y - .75 * \noexpand\tqft@center@y}
370 \noexpand\pgfmathsetlength{\noexpand\pgf@x}{\noexpand\pgf@x + (\the\c@pgf@counta - .5) * \noexpand\tqft@center@x}
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(\tq
407     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoi
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@separ
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(\tq
416     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@ch}}{\pgfqpoi
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{\pgfpoint{\tqft@xb}{-\tqft@c}}{\pgfpoint{\tqft@xa}{-\tqft@c}}{\pgfpoint{\tqft@xa}{-\tqft@c}}
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@separation}
437 \pgfmathsetlength{\tqft@ch}{\min(0,\max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqft@h - \tqft@c)) / \tqft@h))}
438 \pgfpathcurveto{\pgfpoint{\tqft@xb}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{\tqft@ch}}{\pgfpoint{\tqft@xa}{\tqft@ch}}
439 \else
440 \pgfmathsetlength{\tqft@ch}{\min(0,\max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqft@h - \tqft@c)) / \tqft@h))}
441 \pgfpathcurveto{\pgfpoint{\tqft@xb}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{-\tqft@ch}}
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) * \tqft@w)}
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@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

```

Yes, so iterate over them

```

492 \foreach \tqft@k in {1,...,\tqft@incoming} {
493 \advance\tqft@xa by \tqft@k\tqft@s
494 \pgfpathmoveto{\pgfpoint{\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@separation}
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{\pgfpoint{\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}
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(\tqft@
547     \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
548     \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{\tqft@ch}}{\pgfqpoint{
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@separat
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(\tqft@
557         \pgfpathmoveto{\pgfqpoint{\tqft@xb}{\tqft@h}}
558         \pgfpathcurveto{\pgfqpoint{\tqft@xb}{\tqft@ch}}{\pgfqpoint{\tqft@xa}{-\tqft@ch}}{\pgfqpoint{
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{\pgfpoint{\tqft@xb}{-\tqft@c}}{\pgfpoint{\tqft@xa}{-\tqft@c}}{\pgfpoint{\tqft@xa}{-\tqft@c}}
572 \advance\tqft@xa by -\tqft@w
573 \pgfpathmoveto{\pgfpoint{\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@separation}
580 \pgfmathsetlength{\tqft@ch}{\min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqft@c) - \tqft@h) / \tqft@h))}
581 \pgfpathcurveto{\pgfpoint{\tqft@xb}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{\tqft@ch}}{\pgfpoint{\tqft@xa}{\tqft@ch}}
582 \else
583 \pgfmathsetlength{\tqft@ch}{\min(0,max(-\tqft@h,\tqft@h - (\tqft@h - \tqft@c) * ((abs(\tqft@c) - \tqft@h) / \tqft@h))}
584 \pgfpathcurveto{\pgfpoint{\tqft@xb}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{-\tqft@ch}}{\pgfpoint{\tqft@xa}{-\tqft@ch}}
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@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{Opt}{-\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 interpreted 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@depth}}
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 s
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@n
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   genus/.initial = 0,
```

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

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

This is the “vertical” separation between boundary components.

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

This is the “horizontal” separation between boundary components.

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

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

```
778   circle x radius/.initial=10pt,
779   circle y radius/.initial=5pt,
```

These control the direction from which we view the cobordism.

```
780   view from/.is choice,
781   view from/incoming/.code={%
782     \let\pgf@tqft@upper\pgf@tqft@minus
783     \let\pgf@tqft@lower\@empty
```

```

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

```

We simulate node placement using the following key.

```

789 anchor/.initial = none,

```

The next set of keys define some default shapes.

```

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

```

```

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

```

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

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

Now we define the code for the actual cobordism shape.

```
852 \tikzset{
853   cobordism/.pic={

```

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

```
854   \gdef\tqft@fullpath{}
```

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

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

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

```
856   \gdef\tqft@gclip{}
```

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

```
857   \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).

```
858   \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.

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

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

```
860   \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.

```

861     \xdef\tqft@fullpath{%
862         \tqft@fullpath
863         (-\tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180, end angle=0, x rad
864     }%

```

And add the centre to the list for available shifts.

```

865     \xdef\tqft@alist{%
866         \tqft@alist,-incoming boundary 1/{(0,0)},-incoming boundary/{(0,0)}%
867     }%

```

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.

```

868     \ifnum\tqft@val{incoming boundary components}>1\relax
869     \foreach[
870         evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation} -\tqft@val{circle x r
871         evaluate=\k as \xpos using (\k-2)*\tqft@val{boundary separation} +\tqft@val{circle x
872         evaluate=\k as \cpos using (\k-1.5)*\tqft@val{boundary separation},
873         evaluate=\k as \kmo using int(\k-1)
874     ] \k in {2,...,\tqft@val{incoming boundary components}} {

```

Add the curve and the arc.

```

875         \xdef\tqft@fullpath{%
876             \tqft@fullpath
877             .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-\tqft@val{cobordism height
878         }%

```

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

```

879         \xdef\tqft@blist{%
880             \tqft@blist,incoming boundary \k/incoming/{%
881             (\xpos pt,0) .. controls +(0,-\tqft@val{cobordism height}/3) and +(0,-\tqft@val{cobord
882         }%

```

We add a coordinate at the midpoint of the curve.

```

883         \xdef\tqft@clist{%
884             \tqft@clist,-between incoming \kmo\space and \k/{(\cpos pt,-\tqft@val{cobordism height
885         }%

```

And add the centre to the list for available shifts.

```

886         \xdef\tqft@alist{%
887             \tqft@alist,-incoming boundary \k/{(\kmo * \tqft@val{boundary separation},0)}%
888         }%
889     }%
890     \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.

```

891     \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,

```

892     \pgfmathsetmacro\xppos{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset})
893     \pgfmathsetmacro\tqft@ht{abs(\tqft@val{incoming boundary components} - \tqft@val{outgoing
894     \pgfmathsetmacro\tqft@ht{1/3 + 2/3*\tqft@ht/(\tqft@ht + 1)}%
895     \xdef\tqft@fullpath{%
896         \tqft@fullpath
897         .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
898     }%

```

and the edge path.

```

899     \xdef\tqft@blist{%
900         \tqft@blist,between last incoming and last outgoing/incoming and outgoing/{%
901         (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@val{
902         .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cob
903     }%

```

In addition, we add a coordinate at the midpoint.

```

904     \pgfmathsetmacro\xppos{(\xppos + (\tqft@val{incoming boundary components} -1) * \tqft@val{
905     \xdef\tqft@clist{%
906         \tqft@clist,-between last incoming and last outgoing/{(\xppos pt,-\tqft@val{cobordism
907     }%
908     \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.

```

909     \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{incoming boundary components} - 1)/\tqft@
910     \xdef\tqft@fullpath{%
911         \tqft@fullpath
912         .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cob
913     }%

```

Same for the edge path.

```

914     \xdef\tqft@blist{%
915         \tqft@blist,between first incoming and last incoming/incoming and outgoing/{%
916         (\tqft@val{incoming boundary components} * \tqft@val{boundary separation} + \tqft@val{
917         .. controls +(0,-\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cob
918     }%

```

Add a coordinate at the midpoint.

```

919     \pgfmathsetmacro\xppos{(\tqft@val{incoming boundary components} -1) * \tqft@val{boundary
920     \xdef\tqft@clist{%
921         \tqft@clist,-between first incoming and last incoming/{(\xppos pt,-\tqft@ht*\tqft@val{
922     }%
923     \fi
924     \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.

```

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

```

Add a move to the full path,

```

927     \xdef\tqft@fullpath{%
928         \tqft@fullpath
929         (\xppos pt, -\tqft@val{cobordism height})
930     }%
931     \fi
932     \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.

```

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

```

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

```

935     \xdef\tqft@fullpath{%
936         \tqft@fullpath
937         arc[end angle=\pgf@tqft@upper180, start angle=0, x radius=\tqft@val{circle x radius},
938     }%

```

And add the centre to the list for available shifts.

```

939     \xdef\tqft@alist{%
940         \tqft@alist,-outgoing boundary \tqft@val{outgoing boundary components}/{(\xppos pt +
941     }%

```

Do we have more than one boundary component?

```

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

```

Yes, so add a curve and arc for each.

```

943     \foreach[
944         evaluate=\k as \xpos using (\tqft@val{outgoing boundary components} - \k + \tqft@val{
945         evaluate=\k as \xppos using (\tqft@val{outgoing boundary components} - \k + 1 + \tqft@
946         evaluate=\k as \cpos using (\tqft@val{outgoing boundary components} - \k + .5 + \tqft@
947         evaluate=\k as \nk using int(\tqft@val{outgoing boundary components} - \k + 1),
948         evaluate=\k as \nkpo using int(\tqft@val{outgoing boundary components} - \k + 2),
949     ] \k in {2,...,\tqft@val{outgoing boundary components}} {

```

Both are added to the full path.

```

950     \xdef\tqft@fullpath{%
951         \tqft@fullpath
952         .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height}),
953     }%

```

Just the arc for the edge paths.

```

954     \xdef\tqft@blist{%
955         \tqft@blist,between outgoing \nk\space and \nkpo/outgoing/{%
956     (\xppos pt,-\tqft@val{cobordism height})
957         .. controls +(0,\tqft@val{cobordism height}/3) and +(0,\tqft@val{cobordism height}),
958     }%

```

And a coordinate at the midpoint.

```

959     \xdef\tqft@clist{%

```

```

960     \tqft@clist,-between outgoing \nk\space and \nkpo/{(\cpos pt,-3*\tqft@val{cobordism height}
961     }%

```

And add the centre to the list for available shifts.

```

962     \xdef\tqft@alist{%
963     \tqft@alist,-outgoing boundary \nk/{(\xpos pt - \tqft@val{circle x radius},-\tqft@val{circle x radius})}
964     }%
965     }%
966     \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.

```

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

```

There are, so we draw the path back up.

```

968     \pgfmathsetmacro\tqft@ht{1/3 + 2/3*abs(\tqft@val{offset})/(abs(\tqft@val{offset}) + 1)}
969     \xdef\tqft@fullpath{%
970     \tqft@fullpath
971     .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cobordism height})
972     }%

```

And the edge path does the same.

```

973     \xdef\tqft@blist{%
974     \tqft@blist,between first incoming and first outgoing/incoming and outgoing/{%
975     (\tqft@val{offset} * \tqft@val{boundary separation} - \tqft@val{circle x radius},-\tqft@val{circle x radius})}
976     .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,-\tqft@ht*\tqft@val{cobordism height})
977     }%

```

Add a coordinate at the midpoint.

```

978     \xdef\tqft@clist{%
979     \tqft@clist,-between first incoming and first outgoing/{(\tqft@val{offset}*\tqft@val{circle x radius})}
980     }%
981     \else

```

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

```

982     \pgfmathsetmacro\xppto{(\tqft@val{outgoing boundary components} -1+\tqft@val{offset}) *
983     \pgfmathsetmacro\tqft@ht{1/3 + 2/3*(\tqft@val{outgoing boundary components} - 1)/\tqft@val{circle x radius}}

```

Full path.

```

984     \xdef\tqft@fullpath{%
985     \tqft@fullpath
986     .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cobordism height})
987     }%

```

Edge path.

```

988     \xdef\tqft@blist{%
989     \tqft@blist,between first and last outgoing/incoming and outgoing/{%
990     (\tqft@val{offset} * \tqft@val{boundary separation} - \tqft@val{circle x radius},-\tqft@val{circle x radius})}
991     .. controls +(0,\tqft@ht*\tqft@val{cobordism height}) and +(0,\tqft@ht*\tqft@val{cobordism height})
992     }%

```

Add a coordinate at the midpoint.

```

993     \pgfmathsetmacro\xpos{(\tqft@val{outgoing boundary components}/2 + \tqft@val{offset} -
994     \pgfmathsetmacro\tqft@ht{1 -\tqft@ht*3/4}%
995     \xdef\tqft@clist{%
996         \tqft@clist,-between first and last outgoing/{(\xpos pt,-\tqft@ht*\tqft@val{cobordism
997     }%
998     \fi
999     \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.

```

1000    \pgfmathsetmacro\xpos{%
1001    (
1002        \tqft@val{outgoing boundary components} > 0 ?
1003    (
1004        \tqft@val{incoming boundary components} > 0 ?
1005        min(0,\tqft@val{offset}) : \tqft@val{offset}
1006    ) : 0
1007    )
1008    *\tqft@val{boundary separation} - 2*\tqft@val{circle x radius}}%
1009    \xdef\tqft@gclip{(\xpos pt,2*\tqft@val{circle y radius}) rectangle }%

```

Now the bottom right.

```

1010    \pgfmathsetmacro\xpos{%
1011    ((
1012        \tqft@val{outgoing boundary components} > 0 ?
1013    (
1014        \tqft@val{incoming boundary components} > 0 ?
1015        max(\tqft@val{incoming boundary components},\tqft@val{outgoing boundary components} +
1016        ) : \tqft@val{incoming boundary components}
1017    )-1)
1018    *\tqft@val{boundary separation} + 2*\tqft@val{circle x radius}}%

```

Together, these make a rectangle.

```

1019    \xdef\tqft@gclip{\tqft@gclip (\xpos pt,-\tqft@val{cobordism height} - 2*\tqft@val{circle

```

Are there any holes?

```

1020    \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.

```

1021    \pgfmathsetmacro\xpos{%
1022    (
1023        \tqft@val{outgoing boundary components} > 0 ?
1024    (
1025        \tqft@val{incoming boundary components} > 0 ?
1026        \tqft@val{offset}/2 : \tqft@val{offset}
1027    ) : 0
1028    )
1029    *\tqft@val{boundary separation} - \tqft@val{circle x radius}}%

```

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

```

1030 \pgfmathsetmacro\ypos{%
1031 (
1032 \tqft@val{outgoing boundary components} > 0 ?
1033 (
1034 \tqft@val{incoming boundary components} > 0 ?
1035 -\tqft@val{cobordism height}/2 : -1 + \tqft@val{cobordism height}/3
1036 ) : - \tqft@val{cobordism height}/3
1037 )}%

```

Start our clip path at this point

```

1038 \xdef\tqft@gclip{%
1039 \tqft@gclip
1040 (\xpos pt,\ypos pt)
1041 }%

```

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.

```

1042 \pgfmathsetmacro\gsize{%
1043 ((
1044 \tqft@val{outgoing boundary components} > 0 ?
1045 (
1046 \tqft@val{incoming boundary components} > 0 ?
1047 (\tqft@val{incoming boundary components} + \tqft@val{outgoing boundary components})/2
1048 ) : \tqft@val{incoming boundary components}
1049 )-1)
1050 *\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, `\gyscale` 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.

```

1051 \pgfmathsetmacro\gyscale{\min(1,\gsize/(3*\tqft@val{genus}+1))}%
1052 \pgfmathsetmacro\gyscale{\tqft@val{circle y radius}*\gyscale*.707}%
1053 \pgfmathsetmacro\gxscale{\tqft@val{circle x radius}*\gyscale}%

```

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 into $g + 1$ lots.

```

1054 \pgfmathsetmacro\gsep{((\gsize - 2*\tqft@val{genus}*\gyscale)/(\tqft@val{genus} + 1)*\tqft@val{circle x radius})}

```

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

```

1055 \xdef\tqft@gclip{%
1056 \tqft@gclip
1057 ++(\gsep/2 pt,0)
1058 }%

```

Some useful quantities.

```

1059 \pgfmathsetmacro\omrstwo{1 - 1/sqrt(2)}%
1060 \pgfmathsetmacro\sqrtwo{sqrt(2)}%

```

Now we iterate over the holes.

```

1061     \foreach[
1062       evaluate=\k as \kmo using int(2 * \k-1)
1063     ] \k in {1,...,\tqft@val{genus}} {

```

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

```

1064       \xdef\tqft@gclip{%
1065         \tqft@gclip

```

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

```

1066         ++(\gsep/2 pt + \omrstwo*\gxscale pt,0)

```

Now curve up over the hole,

```

1067         .. controls +(\gxscale*\sqrtwo/3 pt,4/3*\gyscale pt) and +(-\gxscale*\sqrtwo/3 pt,4/3*\gyscale pt)
1068         .. ++(\sqrtwo*\gxscale pt,0)

```

and return on the underside.

```

1069         .. controls +(-\gxscale*\sqrtwo/3 pt,-4/3*\gyscale pt) and +(\gxscale*\sqrtwo/3 pt,-4/3*\gyscale pt)
1070         .. ++(-\sqrtwo*\gxscale pt,0)

```

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

```

1071         ++(2*\gxscale pt -\omrstwo*\gxscale pt + \gsep/2 pt,0)
1072     }

```

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 account the view from direction.

```

1073     \xdef\tqft@glist{%
1074       \tqft@glist,%
1075       hole \k/lower/{%

```

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

```

1076         (\xpos pt + \k * \gsep pt + \kmo * \gxscale pt + \gxscale pt -\omrstwo*\gxscale pt)
1077         .. controls +(-\gxscale pt*\sqrtwo/3,\pgf@tqft@upper4/3*\gyscale pt) and +(\gxscale pt*\sqrtwo/3,\pgf@tqft@upper4/3*\gyscale pt)
1078         .. ++(-\sqrtwo*\gxscale pt,0)},%
1079 % Move to the left-hand corner of the path, with the upper or lower chosen by the \verb+view+ direction
1080         hole \k/upper/{(\xpos pt + \k * \gsep pt + \kmo * \gxscale pt - \gxscale pt,\ypos pt)

```

Add the larger of the two curves.

```

1081         .. controls +(\gxscale pt*2/3,\pgf@tqft@lower8/3*\gyscale pt) and +(-\gxscale pt*2/3,\pgf@tqft@lower8/3*\gyscale pt)
1082         .. ++(2*\gxscale pt,0)}%
1083     }%

```

Add a coordinate at the centre of the hole.

```

1084     \xdef\tqft@cclist{%
1085       \tqft@cclist,-hole \k/{(\xpos pt + \k * \gsep pt + \kmo * \gxscale pt,\ypos pt)}%
1086     }%
1087   }%
1088 \fi

```

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

```

1089 \gdef\tqft@shift{(0,0)}%
1090 \edef\tqft@anchor{\tqft@val{anchor}}%

```

```

1091 \expandafter\pgfutil@in@\expandafter{\expandafter,\expandafter}\expandafter{\tqft@anchor}%
1092 \ifpgfutil@in@
1093 \expandafter\tqft@split\tqft@anchor\relax
1094 \else
1095 \edef\tqft@anchor{-\tqft@val{anchor}}%
1096 \xdef\tqft@alist{\tqft@clist,\tqft@alist}%
1097 \foreach \anchor/\coord in \tqft@alist
1098 {
1099   \ifx\anchor\tqft@anchor\relax
1100   \global\let\tqft@shift\coord
1101   \fi
1102 }%
1103 \fi
1104 \tikz@scan@one@point\pgfutil@firstofone\tqft@shift\relax
1105 \begin{scope}[shift={(-\pgf@x,-\pgf@y)}]

```

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

```

1106   \ifnum\tqft@val{incoming boundary components}>0\relax
1107   \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in {1,...,
1108     \node[
1109       node contents={},
1110       ellipse,
1111       inner sep=0pt,
1112       outer sep=0pt,
1113       minimum width=2*\tqft@val{circle x radius},
1114       minimum height=2*\tqft@val{circle y radius},
1115       at={(\xpos pt,0)},
1116       name=-incoming boundary \k,
1117       /tikz/tqft/every boundary component/.try,
1118       /tikz/tqft/every incoming boundary component/.try,
1119       /tikz/tqft/incoming boundary component \k/.try
1120     ];
1121   }%

```

Add an alias for the first.

```

1122   \path node also[pic alias=-incoming boundary] (-incoming boundary 1);
1123   \fi

```

Same for the outgoing boundary components.

```

1124   \ifnum\tqft@val{outgoing boundary components}>0\relax
1125   \foreach[
1126     evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1127   ] \k in {1,...,\tqft@val{outgoing boundary components}} {
1128     \node[
1129       node contents={},
1130       ellipse,
1131       inner sep=0pt,
1132       outer sep=0pt,
1133       minimum width=2*\tqft@val{circle x radius},
1134       minimum height=2*\tqft@val{circle y radius},

```

```

1135         at={(\xpos pt,-\tqft@val{cobordism height})},
1136         name=-outgoing boundary \k,
1137         /tikz/tqft/every boundary component/.try,
1138         /tikz/tqft/every outgoing boundary component/.try,
1139         /tikz/tqft/outgoing boundary component \k/.try
1140     ];
1141 }%

```

Add an alias for the first.

```

1142     \path node also[pic alias=-outgoing boundary] (-outgoing boundary 1);
1143     \fi

```

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

```

1144     \ifnum\tqft@val{incoming boundary components}>0\relax
1145     \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in {1,...,
1146         \path[
1147             /tikz/tqft/every lower boundary component/.try,
1148             /tikz/tqft/every incoming lower boundary component/.try,
1149             /tikz/tqft/incoming lower boundary component \k/.try
1150         ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@lower180,end angle=
1151         ]%
1152     \fi

```

Same for the outgoing boundary components.

```

1153     \ifnum\tqft@val{outgoing boundary components}>0\relax
1154     \foreach[
1155         evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1156     ] \k in {1,...,\tqft@val{outgoing boundary components}} {
1157         \path[
1158             /tikz/tqft/every lower boundary component/.try,
1159             /tikz/tqft/every outgoing lower boundary component/.try,
1160             /tikz/tqft/outgoing lower boundary component \k/.try
1161         ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start angle=
1162         ]%
1163     \fi

```

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

```

1164     \begin{scope}
1165     \path[overlay,clip] \tqft@gclip;
1166     \path[
1167         /tikz/tqft/cobordism/.try,
1168         pic actions,
1169         /tikz/tqft/cobordism outer path/.try,
1170     ] \tqft@fullpath;
1171     \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.


```

1172 \ifx\tqft@glist\pgfutil@gobble
1173 \else
1174 \foreach \tqft@gstyle/\tqft@gside/\tqft@gpath in \tqft@glist {
1175     \path[
1176         /tikz/tqft/cobordism/.try,
1177         pic actions,
1178         fill=none,
1179         shade=none,
1180         /tikz/tqft/cobordism edge/.try,
1181         /tikz/tqft/genus style/.try,
1182         /tikz/tqft/genus \tqft@gside/.try,
1183         /tikz/tqft/\tqft@gstyle/.try,
1184         /tikz/tqft/\tqft@gstyle\space\tqft@gside/.try,
1185     ] \tqft@gpath;
1186 }
1187 \fi

```

Now we redraw the non-boundary paths.

```

1188 \ifx\tqft@blist\pgfutil@gobble
1189 \else
1190 \foreach \tqft@bstyle/\tqft@btype/\tqft@bpath in \tqft@blist {
1191     \path[
1192         /tikz/tqft/cobordism edge/.try,
1193         /tikz/tqft/cobordism outer edge/.try,
1194         /tikz/tqft/between \tqft@btype/.try,
1195         /tikz/tqft/\tqft@bstyle/.try,
1196     ] \tqft@bpath;
1197 }
1198 \fi

```

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

```

1199     \ifx\tqft@clist\pgfutil@gobble
1200     \else
1201     \foreach \name/\coord in \tqft@clist {
1202         \path \coord node[coordinate,node contents={},name=\name];
1203     }
1204     \fi

```

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

```

1205     \ifnum\tqft@val{incoming boundary components}>0\relax
1206     \foreach[evaluate=\k as \xpos using (\k-1)*\tqft@val{boundary separation}] \k in {1,...,
1207         \path[
1208             /tikz/tqft/every upper boundary component/.try,
1209             /tikz/tqft/every incoming upper boundary component/.try,
1210             /tikz/tqft/incoming upper boundary component \k/.try
1211         ] (\xpos pt - \tqft@val{circle x radius},0) arc[start angle=\pgf@tqft@upper180,end ang
1212         ]
1213     \fi

```

Next, outgoing.

```
1214     \ifnum\tqft@val{outgoing boundary components}>0\relax
1215     \foreach[
1216       evaluate=\k as \xpos using (\k-1+\tqft@val{offset})*\tqft@val{boundary
1217     ] \k in {1,...,\tqft@val{outgoing boundary components}} {
1218       \path[
1219         /tikz/tqft/every upper boundary component/.try,
1220         /tikz/tqft/every outgoing upper boundary component/.try,
1221         /tikz/tqft/outgoing upper boundary component \k/.try
1222       ] (\xpos pt - \tqft@val{circle x radius},-\tqft@val{cobordism height}) arc[start angle
1223     ]
1224     \fi
1225 \end{scope}
We're done! Phew.
1226 }
1227 }
```