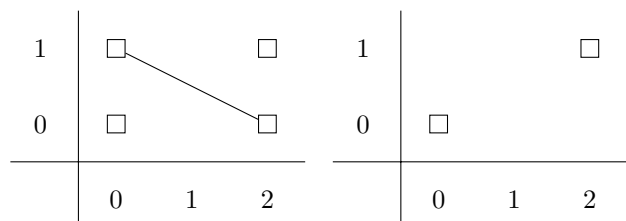


The `sseqpages` package provides two environments – the `sseqdata` environment, for specifying the raw data of a spectral sequence, and the `sseqpage` environment, for printing a page of a spectral sequence. There is also a command `\printpage` for when the body of `sseqpage` is empty. This package is built very closely on top of Tikz, so for the most part, standard Tikz options carry over to do the same thing when used with `sseqpages`. Because of this, the package is much easier to use if you already know how to use Tikz. Since that seems to be a minority of potential users, I will make sure to explain exactly what features are inherited from Tikz.

**Example 1.** Here is a very short example to illustrate the most basic usage:

```
\begin{sseqdata}[degree={#1}{-#1+1}]{example1}
\place(0,0)
\place(2,0)
\place(0,1)
\place(2,1)
\d2(0,1)
\end{sseqdata}
\printpage[name=example1,page=2]
\printpage[name=example1,page=3]
```

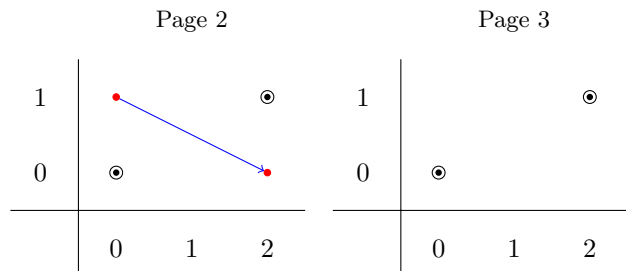


The option `degree={#1}{-#1+1}` specifies that the bidegree of the  $E_r$  differential is  $(r, -r + 1)$  – that is, a page  $r$  differential goes right  $r$  and down  $r - 1$ . `example1` is name of this spectral sequence. The command `\place(\langle x \rangle, \langle y \rangle)` places a class at  $(\langle x \rangle, \langle y \rangle)$ . We place four classes, then use the command `\d2(2,0)` which places a differential connecting the class at  $(2,0)$  to the one at  $(0,1)$ . Then we use `\printpage` to display pages two and three of the spectral sequence. We see a single differential on page two. On page three, its source and target disappear. This is the general behavior of the package – whenever a class supports or accepts a differential, it disappears on subsequent pages.

**Example 2.** The appearance of example 1 was pretty rough. This example polishes it up a bit. The `sseqpages` package offers a number of "style" options that control the appearance of certain features of the spectral sequence. For instance, `classes` controls the appearance of every class and `differentials` controls the appearance of every differential. There are also options `permanent cycles` and `transient cycles`, that respectively affect the appearance of only classes that never support or are hit by a differential, or that are eventually hit by a differential.

Classes are internally drawn as Tikz nodes, so the options that you provide to `classes`, `permanent cycles`, `transient cycles`, and individual `\place` commands are mostly the same as options that you would provide to the `\node` command. Differentials are drawn as Tikz edges, so the options you can provide to `differentials` or a `\d` command are mostly the same as the options you would provide to `\draw`.

```
\sseqset{degree={#1}{-#1+1},classes={inner sep=0.2ex},differentials=->}
\begin{sseqdata}[
  classes={fill,circle},
  differentials=blue,
  permanent cycles={double distance=1,inner sep=0.2ex+0.35pt},
  transient cycles=red
]{example2}
\node[background,font=\small] at (\xmin/2+\xmax/2,\ymax+1) {\textup{Page \sseqthepage{}}};
\foreach \x in {0,2} \foreach \y in {0,1}{
  \place(\x,\y)
}
\d2(0,1)
\end{sseqdata}
\printpage[name=example2,page=2]
\printpage[name=example2,page=3]
```



The command `\sseqset` sets options for all spectral sequences in the current scope. Note in particular that these changes are local to the current environment. The option `classes` adds the given options to the style `every class` which controls the appearance of all classes. The classes above were too large for my taste, so I asked for them to be somewhat smaller by default in all future diagrams using the Tikz option `inner sep=0.2ex`. Also, the option `differentials=->` will make all future differentials end in an arrow tip, the same as it does in the Tikz command `\draw[->] <source>--<target>;`.

For this particular diagram, I asked for the classes to be filled circles with `classes={fill,circle}`, and for the differentials to be blue by setting `differentials=blue`. We ask for permanent cycles to be drawn with an extra circle around them, and for transient cycles to be red. All of these are also Tikz options.

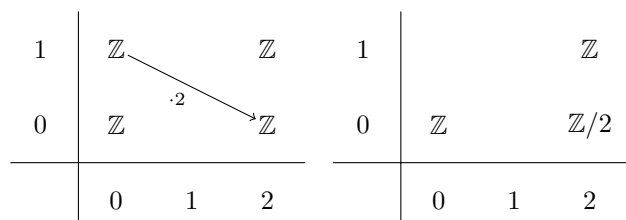
The command `\node[background,font=\small] at (\xmin/2+\xmax/2,\ymax+1) {\textup{Page \sseqthepage{}}};` adds a title to the background that says what page it is. This is actually a direct Tikz command, although

the commands `\xmin`, `\xmax`, `\ymax`, and `\sseqthepage` are defined by `sseqpages`. The option `background` is specific to `sseqpages` and it tells the package to put this Tikz command in the background (rather than on top of all classes, structlines, and differentials as is the default).

Instead of calling the `\place` command four times by hand, we use the Tikz command `\foreach` (actually `\foreach` is from PGF, but whatever).

**Example 3.** Named cells and the `\replace` command.

```
\sseqset{execute at begin node=$, execute at end node=$}
\begin{sseqdata}[
  classes={draw=none},
]{example3}
\foreach \x in {0,2} \foreach \y in {0,1}{
  \place["\mathbb{Z}"](\x,\y)
}
\d["\cdot 2"]2(0,1)
\replace["\mathbb{Z}/2"](2,0)
\end{sseqdata}
\printpage[name=example3,page=2]
\printpage[name=example3,page=3]
```



All of the labels in this document are in math mode, so I ask Tikz to automatically put every node in math mode using the options `execute at begin node=$`, `execute at end node=$`. This includes both the names of classes and any labels given for differentials.

If you want to put something in the body of the node printed by the `\place` command, this is specified in quotes. For instance the command `\place["\mathbb{Z}"](\x,\y)` puts a  $\mathbb{Z}$  in the node. Likewise, to label a differential, place the desired label in quotes, like `\d["\cdot 2"]2(0,1)`. (For differentials, this uses Tikz `edge quotes`.) I don't want the  $\mathbb{Z}$  to be surrounded by a drawn border so I pass the global option `classes={draw=none}` to tell Tikz not to draw the boundary of the node.

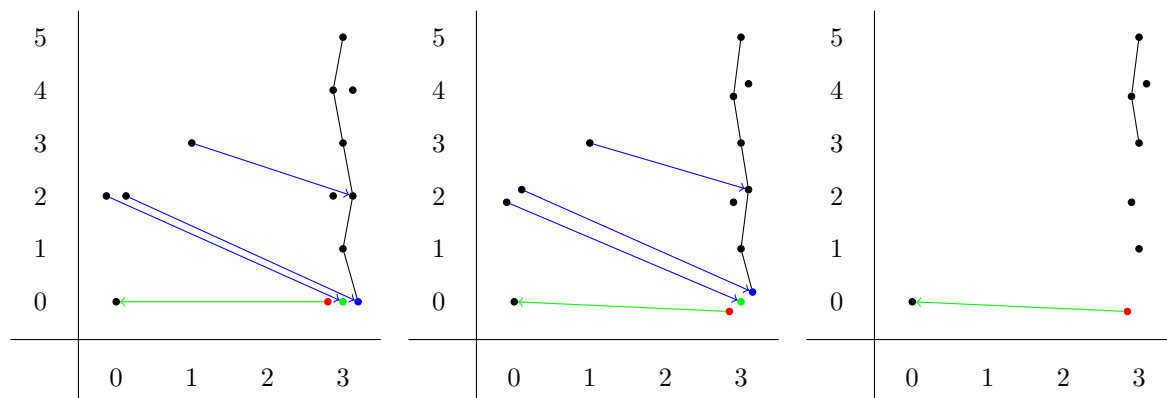
Note also the use of the command `\replace["\mathbb{Z}/2"](2,0)`. After a class supports or is hit by a differential (or both), it will disappear on subsequent pages. If you want to put a new node in the same location (or the same node), it is necessary to use the `\replace` command.

**Example 4.** Multiple nodes in the same location and the `\structline` command.

```
\begin{sseqdata}[classes={circle,fill},differentials=blue,yscale=0.7]{example4}
\place(0,2)\place(0,2)
\place(3,0)\place(3,0)
\place(3,2)

\place(3,0)\place(3,4)
\foreach \y in {1,...,5} {
  \place(3,\y)
  \structline(3,\y-1,-1)(3,\y,-1)
}

\place(1,3)
\d2(1,3,, -1)
\d3(0,2,1,2)
\d3(0,2,-1,-1)
\place(0,0)
\draw[->,green](3,0,1)--(0,0);
\end{sseqdata}
\printpage[name=example4,page=3]
\printpage[name=example4,page=3,class placement transform={rotate=40}]
\printpage[name=example4,page=4,class placement transform={rotate=40}]
```



The option `yscale=0.7` is a Tikz coordinate transformation option that rescales the y-axis to be 0.7 times its original length.

If you set `page=0`, then `sseqpages` will draw all differentials of any length as well as all classes. In this example, we see the differentials of length three and one differential of length two in the same diagram.

Multiple nodes in the same location are automatically arranged in a standard pattern. In the future it will be possible for the user to exert a significant amount of control over this pattern, but right now, there's only one choice. The `sseqpages` option `class placement transform` allows the user to specify a Tikz coordinate transform to adjust the relative position of multiple nodes in the same position. This coordinate transform can only involve rotation and scaling, no translation. In this case, `class placement transform={rotate=40}` rotates the classes in the same position by 40 degrees.

To place multiple nodes in the same location, you just call `\place` with the same coordinate twice. To

place a differential where the source or target have multiple nodes, you say `\d<math>r</math>(<math>\langle x \text{ source} \rangle, \langle y \text{ source} \rangle, \langle n \text{ source} \rangle, \langle n \text{ target} \rangle</math>)` where  $\langle n \text{ source} \rangle$  and  $\langle n \text{ target} \rangle$  specify which of the nodes in the source and target location to use. If the number  $\langle n \rangle$  is positive, it counts from the first node placed in that position, whereas if it is negative, it counts from the most recent node placed in that position.

The command `\structline<math>\langle source \text{ coord} \rangle \langle target \text{ coord} \rangle</math>` places a line from the source class to the target class, but will only draw this line on pages where both the source and target are still alive. If multiple nodes are in the same position, then the syntax for the coordinate is  $(\langle x \rangle, \langle y \rangle, \langle n \rangle)$ , where as with `\d`, if  $\langle n \rangle$  is negative it counts backwards from the most recently placed node.

Using negative values is particularly useful when drawing repeated structures. This way, independent of how many classes are already present in a given position, the new structure will be properly grouped. Consider, for instance, the following code:

```
\def\tower#1{
  \begin{scope}[#1]
    \place(0,0)
    \place(0,2)
    \foreach \y in {1,...,5} {
      \place(0,\y)
      \structline(0,\y-1,-1)(0,\y,-1)
    }
    \structline[yellow](0,1)(0,2,-2)
    \structline(0,2,-2)(0,3)
  \end{scope}
}

\begin{sseqdata}[classes={circle,fill}]{example5}
\place(1,1)\place(1,2)
\place(2,3)\place(2,3)\place(2,5)
\tower{xshift=0,classes=blue}
\tower{xshift=1,dashed,green}
\tower{xshift=2,struct lines=red}
\end{sseqdata}
\printpage[name=example5,page=0]
```

