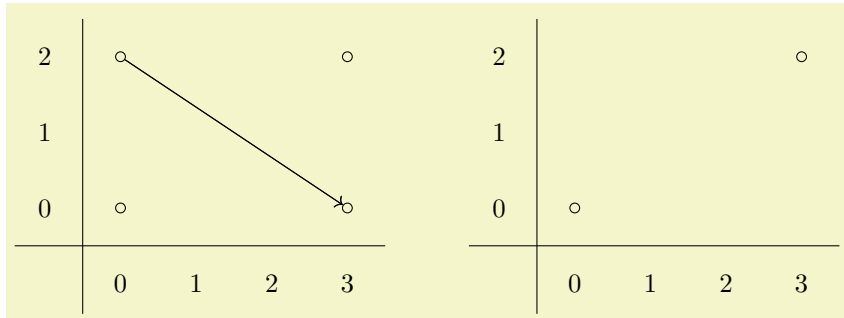


# 1 Introduction

The `sseqpages` package consists of two main environments – the `sseqdata` environment, which specifies the data for a named spectral sequence diagram, and the `sseqpage` environment, which prints a single page of a spectral sequence diagram. The command `\printpage` is also available as a synonym for a `sseqpage` environment with an empty body.

Here is a basic example:



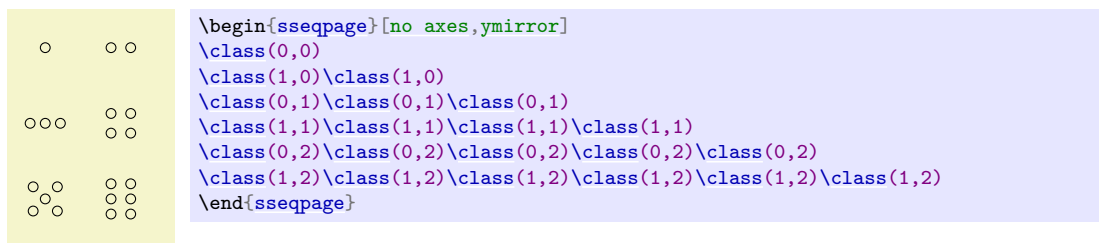
```
\begin{sseqdata}[name=ex1,cohomological Serre grading]
\class(0,0)
\class(0,2)
\class(3,0)
\class(3,2)
\d3(0,2)
\end{sseqdata}
\printpage[name=ex1,page=3]\hskip1cm
\printpage[name=ex1,page=4]
```

`\begin{sseqdata}[name=ex1,degree={#1}{1-#1}]` starts the declaration of the data of a spectral sequence named *ex1* whose page  $r$  differentials go  $r$  to the right and down  $r - 1$  (this is cohomological Serre grading). Then we specify four classes and one page 3 differential, and we ask `sseqpages` to print the third and fourth pages of the spectral sequence. Note that on the fourth page, the source and target of the differential have disappeared.

## 2 The main commands

$$\backslash\text{class}[\langle\textit{options}\rangle]\langle\textit{coordinate}\rangle$$

This places a class at  $\langle coordinate \rangle = (\langle xcoord \rangle, \langle ycoord \rangle)$  where  $\langle xcoord \rangle$  and  $\langle ycoord \rangle$  are integers. If multiple classes occur at the same position, **sseqpages** will automatically arrange them in a pre-specified pattern:

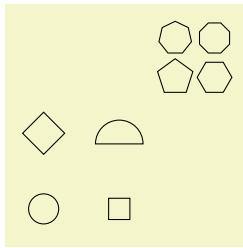


The effect of the `\class` command is to print a `TikZ` node. Any option that would work for a `TikZ` `\node` command will also work in the same way for the `\class`, `\replaceclass`, and `\classoptions` commands. For instance:

A *TikZ* shape

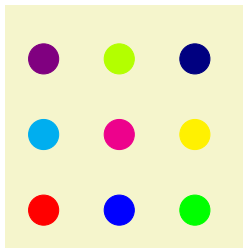
If you give the name of a TikZ shape, the class node will be of that shape. The standard TikZ shapes are `circle` and `rectangle`, but there are many more TikZ shapes in the shapes library.

which you can load using the command `\usetikzlibrary{shapes}`



```
\begin{sseqpage}[no axes,classes={inner sep=0.4em},
                  class placement transform={scale=2}]
\class(0,0)
\class[rectangle](1,0)
\class[diamond](0,1)
\class[semicircle](1,1)
\class[regular polygon, regular polygon sides=5](2,2)
\class[regular polygon, regular polygon sides=6](2,2)
\class[regular polygon, regular polygon sides=7](2,2)
\class[regular polygon, regular polygon sides=8](2,2)
\end{sseqpage}
```

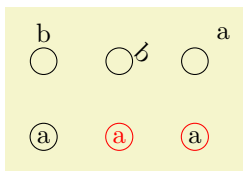
A TikZ color



```
\begin{sseqpage}[classes={fill,inner sep=0.4em}, no axes]
\class[red](0,0)
\class[blue](1,0)
\class[green](2,0)
\class[cyan](0,1)
\class[magenta](1,1)
\class[yellow](2,1)
\class[blue!50!red](0,2)
\class[green!30!yellow](1,2)
\class[blue!50!black](2,2)
\end{sseqpage}
```

"*text*"*<options>*

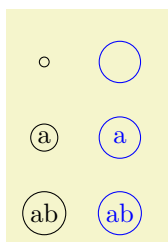
A label. This uses the TikZ quotes syntax, but the behavior specific to `sseqpages`. By default, the *<text>* is placed in the position **inside** the node – in effect, the *<text>* becomes the label text of the node (so saying `\class["label text"](0,0)` causes a similar effect to saying `\node at (0,0) {label text};`). There are other position options such as `left`, `above left`, etc which cause the label text to be placed in a separate node positioned appropriately. If the placement is `above`, `left`, etc, then any option that you may pass to a TikZ node will also work for the label, including general coordinate transformations. If the placement is “inside”, then the only relevant *<options>* are those that alter the appearance of text, such as opacity and color.



```
\begin{sseqpage}[no axes,classes={minimum width=width("a")+0.5em}]
\class["a"](0,0)
\class["a",red](1,0)
\class["a" black,red](2,0)
\class["b" above](0,1)
\class["b" {above right,transform shape,rotate=-45}](1,1)
\class["a" {above right={1em}}](2,1)
\end{sseqpage}
```

<code>minimum width=</code> <i>&lt;dimension&gt;</i>	(no default)
<code>minimum height=</code> <i>&lt;dimension&gt;</i>	(no default)
<code>minimum size=</code> <i>&lt;dimension&gt;</i>	(no default)
<code>inner sep=</code> <i>&lt;dimension&gt;</i>	(no default)
<code>outer sep=</code> <i>&lt;dimension&gt;</i>	(no default)

These options control the size of a node. This is typically useful to make the size of nodes consistent independent of the size of their label text. For instance:

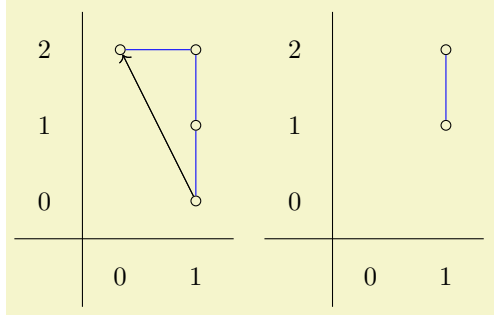


```
\begin{sseqdata}[no axes,name=minimum width example]
\class["ab"](0,0)
\class["a"](0,1)
\class(0,2)
\end{sseqdata}
\printpage[name=minimum width example]
\printpage[name=minimum width example,
            change classes={blue,minimum width=width("ab")+0.5em}]
```

For more information, see the pgf manual.

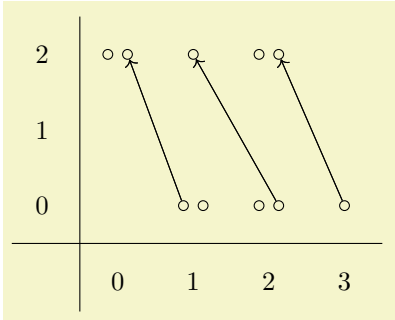
`\d[options](page)(source coordinate)`

This creates a differential starting at  $(\textit{source coordinate})$  of length determined by the specified page. In order to use the `\d` command, you must first specify the **degree** of the differentials as an option to the `sseqdata` or `sseqpage` environment. The degree indicates how far to the right and how far up a page  $r$  differential will go as a function of  $r$ . If there is a page  $r$  differential, on page  $r + 1$ , the source, target, and any `\structlines` connected to the source and target of the differential disappear.



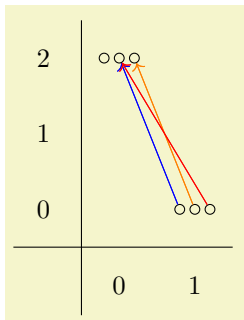
```
\begin{sseqdata}[name=d example,degree={-1}{#1},
    struct lines=blue]
\class(0,2)
\class(1,2)
\class(1,1)
\class(1,0)
\structline(1,2)(0,2)
\structline(1,2)(1,1)
\structline(1,1)(1,0)
\d2(1,0)
\end{sseqdata}
\printpage[name=d example,page=2]
\hskip0.3cm
\printpage[name=d example,page=3]
```

If there are multiple nodes in the source or target coordinate, then there is a funny syntax for indicating which one should be the source and target: `\d<page>(\langle x \rangle, \langle y \rangle, \langle source n \rangle, \langle target n \rangle)`



```
\begin{sseqpage}[Adams grading]
\class(1,0)\class(1,0)
\class(0,2)\class(0,2)
\d2(1,0,1,2)
\class(2,0)\class(2,0)
\class(1,2)
\d2(2,0,2)
\class(3,0)
\class(2,2)\class(2,2)
\d2(3,0,,2)
\end{sseqpage}
```

Negative indices will count from the most recent class in the coordinate (so  $-1$  is the most recent,  $-2$  is the second most recent, etc):

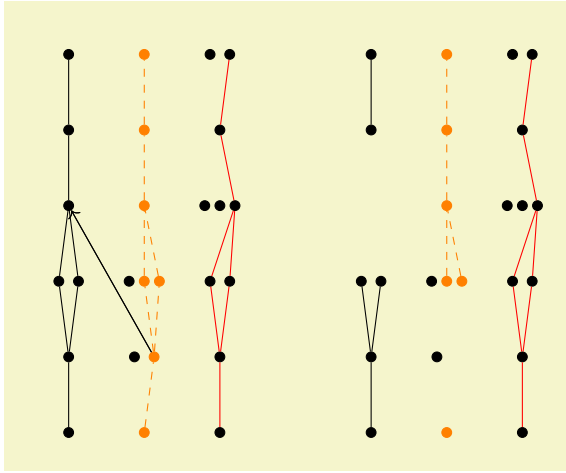


```
\begin{sseqpage}[Adams grading]
\class(1,0)
\class(0,2)\class(0,2)
\d[blue]2(1,0,-1,-1)
\class(1,0)
\class(0,2)
\d[orange]2(1,0,-1,-1)
\class(1,0)
\d[red]2(1,0,-1,-2)
\end{sseqpage}
```

`\structline[options](source coordinate)(target coordinate)`

This command creates a structure line from  $(\textit{source coordinate})$  to  $(\textit{target coordinate})$ . The source and target coordinates are of the form  $(\langle x \rangle, \langle y \rangle, \langle n \rangle)$ . If there are multiple classes at  $(x, y)$ , then  $\langle n \rangle$  specifies which of the classes at  $(x, y)$  the structline starts and ends at – if  $n$  is positive, then it counts from the first class in that position, if  $n$  is negative, it counts backwards from the most recent.

If the source or target of a structure line is hit by a differential, then on subsequent pages, the structure line disappears.



```
\sseqnewgroup\tower{
  \class(0,0)
  \class(0,2)
  \foreach \y in{1,...,5}{
    \class(0,\y)
    \structline(0,\y-1,-1)(0,\y,-1)
  }
  \structline(0,1,-1)(0,2,-2)
  \structline(0,2,-2)(0,3,-1)
}
\begin{sseqdata}[name=structline example,
  classes={circle,fill},
  Adams grading, no axes]
\class(1,1)\class(1,2)
\class(2,3)\class(2,3)\class(2,5)
\tower[classes=blue](0,0)
\tower[struct lines=dashed,orange](1,0)
\tower[struct lines=red](2,0)
\d2(1,1,2)
\end{sseqdata}
\printpage[name=structline example,page=2]
\hskip1cm
\printpage[name=structline example,page=3]
```

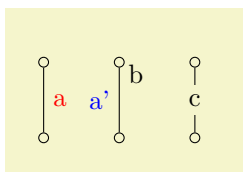
## 2.1 Options for `\d` and `\structline`

In general, any option that you could apply to a TikZ “to” command can be applied to both `\d` and `\structline`. Some such options are as follows:

`"<text>"'<options>`

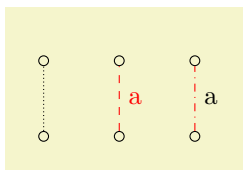
A label `"<text>"'<options>`. By default, such a label is placed to the right of the edge. The optional prime places it to the left of the edge instead. The options include anything you might pass as an option to a TikZ node, including arbitrary coordinate transforms, colors, opacity options, shapes, fill, draw, etc.

The special option “description,” stolen from `tikzcd`, places the label on top of the edge. In order to make this option work correctly, if the background color is not the default white, you must inform `sseqpages` about this using the key `background color=<color>`. In this case, the background color is called `graphicbackground`.



```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1,2} \foreach\y in {0,1}{
  \class(\x,\y)
}
\structline["a" red](0,0)(0,1)
\structline["a'"'blue,"b"{yshift=1em}](1,0)(1,1)
\structline["c" description](2,0)(2,1)
\end{sseqpage}
```

Colors and dash patterns:

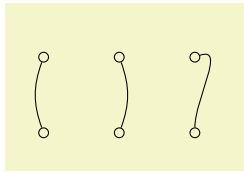


```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1,2} \foreach\y in {0,1}{
  \class(\x,\y)
}
\structline[densely dotted](0,0)(0,1)
\structline[dashed,red, "a"](1,0)(1,1)
\structline[dash dot,red, "a" black](2,0)(2,1)
\end{sseqpage}
```

`bend left=<angle>`  
`bend right=<angle>`

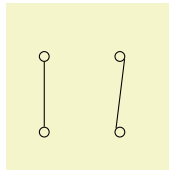
(no default)  
 (no default)

`in=<anchor>` (no default)  
`out=<anchor>` (no default)



```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1,2} \foreach\y in {0,1}{
  \class(\x,\y)
}
\structline[bend left=20](0,0)(0,1)
\structline[bend right=20](1,0)(1,1)
\structline[in=20,out=north](2,0)(2,1)
\end{sseqpage}
```

`source anchor=<anchor>` (no default)  
`target anchor=<anchor>` (no default)



```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1} \foreach\y in {0,1}{
  \class(\x,\y)
}
\structline(0,0)(0,1)
\structline[source anchor=north west,target anchor=-30](1,0)(1,1)
\end{sseqpage}
```

### 3 The Environments

`\begin{sseqdata} [<options>]`  
`<environment contents>`  
`\end{sseqdata}`

The `sseqdata` environment is for storing a spectral sequence to be printed later. This environment is intended for circumstances where you want to print multiple pages of the same spectral sequence. When using the `sseqdata` environment, you must use the `name` option to tell `sseqpages` where to store the spectral sequence so that you can access it later.

`update existing` (no value)

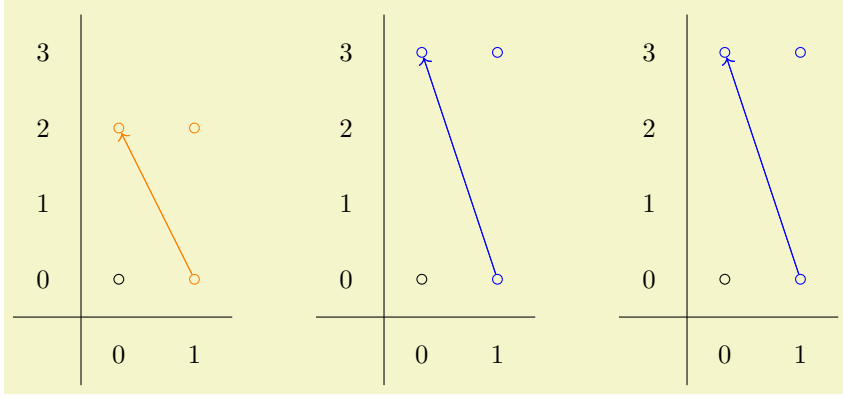
This key specifies that the current `sseqdata` environment is adding data to an already existing spectral sequence. If you don't pass this key, then giving a `sseqdata` environment the same `name` as a different `sseqdata` environment will cause an error.

`\begin{sseqpage} [<options>]`  
`<environment contents>`  
`\end{sseqpage}`

This environment is used for printing a page of an existing spectral sequence with some modification, or for printing a stand-alone page. If you use the `name` option, the name given must match with the name given for some `sseqdata` environment

`keep changes=<boolean>` (default true)(initially false)

This option is only for the `sseqpage` environment. If true, it specifies that all of the options and commands in the `sseqpage` environment should be preserved for future drawings of the spectral sequence. For example:



```

\begin{sseqdata}[name=keep changes example,Adams grading,y range={0}{3}]
\class(0,0)
\class(1,0)
\end{sseqdata}

\begin{sseqpage}[name=keep changes example,paths=orange]
\class(0,2)
\class(1,2)
\classoptions[orange](1,0)
\d2(1,0)
\end{sseqpage}
%
\hskip1cm
%
\begin{sseqpage}[name=keep changes example,paths=blue,keep changes]
\class(0,3)
\class(1,3)
\classoptions[blue](1,0)
\d3(1,0)
\end{sseqpage}
%
\hskip1cm
%
\printpage[name=keep changes example,page=3]

```

Note that the orange classes and differential do not persist because the `keep changes` option is not set in the first `sseqpage` environment, but the blue classes and differential do, since the `keep changes` option is set in the second `sseqpage` environment.

<code>no differentials</code>	(no value)
<code>draw differentials</code>	(no value)
<code>no structlines</code>	(no value)
<code>draw structlines</code>	(no value)

### 3.1 `\printpage`

### 3.2 Global options

`name`= $\langle sseq\ name \rangle$  (no default)

`page`= $\langle page\ number \rangle$  (no default, initially 0)

`degree`= $\{\langle x\ degree \rangle\}\{\langle y\ degree \rangle\}$  (no default)

`cohomological Serre grading` (no value)

`homological Serre grading` (no value)

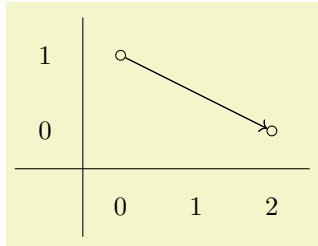
`Adams grading` (no value)

Specifies the degree of differentials. The  $\langle x\ degree \rangle$  and  $\langle y\ degree \rangle$  should both be mathematical expressions in one variable  $\#1$  that evaluate to integers on any input. They specify the x and y displacement

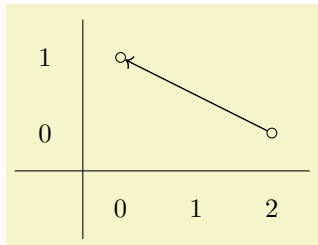
of a page #1 differential. In practice, they will usually be linear expressions with #1 coefficient 1, -1, or 0.

The `degree` option must be given before placing any differentials. It can be specified at the beginning of the `sseqdata` environment, at the beginning of the `sseqpage` environment if it is being used as a standalone page, or as a default by saying `\sseqset{degree={\langle x degree \rangle}{\langle y degree \rangle}}` or `\sseqset{Adams grading}` outside of the `sseqdata` and `sseqpages` environments.

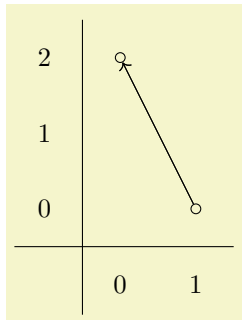
You can make a named grading convention by saying `\sseqset{my grading/.sseq grading={\langle x degree \rangle}{\langle y degree \rangle}}`. Then later passing `my grading` to a spectral sequence is equivalent to saying `degree={\langle x degree \rangle}{\langle y degree \rangle}`. The following grading conventions exist by default:



```
\begin{sseqpage}[cohomological Serre grading]% equivalent to degree={#1}{1-#1}
\class(0,1)
\class(2,0)
\d2(0,1)
\end{sseqpage}
```



```
\begin{sseqpage}[homological Serre grading]% equivalent to degree=-#1{#1-1}
\class(0,1)
\class(2,0)
\d2(2,0)
\end{sseqpage}
```



```
\begin{sseqpage}[Adams grading]% equivalent to degree=-1{#1-1}
\class(0,2)
\class(1,0)
\d2(1,0)
\end{sseqpage}
```

`x range`= $\langle x \min \rangle \langle x \max \rangle$  (no default)

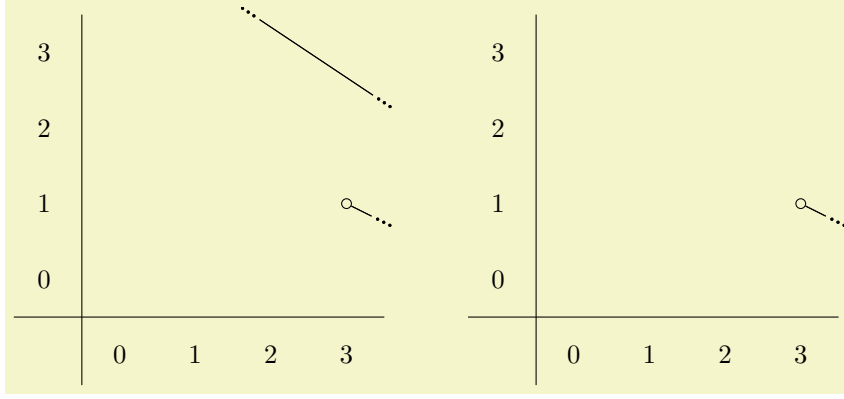
`y range`= $\langle y \min \rangle \langle y \max \rangle$  (no default)

These options force the x and y range to be a specific interval. By default, if no range is specified then the range is chosen to fit all the classes. If an x range is specified but no y range, then the y range is chosen to fit all the classes that lie inside the specified x range, and vice versa.

`no orphan edges` (no value)

`draw orphan edges`= $\langle \text{boolean} \rangle$  (default true)(initially true)

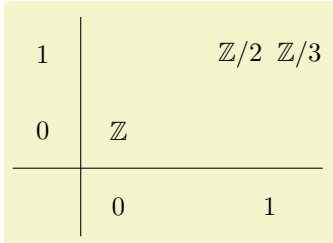
An edge is an “orphan” if both its source and target lie off the page. By default these are drawn, but with the option `no orphan edges` they are not. If the option `no orphan edges` has been set, `draw orphan edges` undoes it.



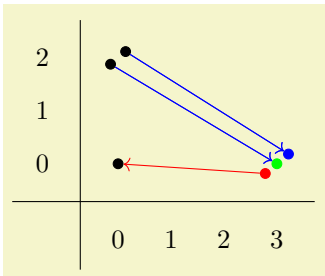
```
\begin{sseqdata}[name=orphan edges example,cohomological Serre grading, x range={0}{3}, y range={0}{3}]
\class(1,4)
\class(4,2)
\d3(1,4)
\class(3,1)
\class(5,0)
\d2(3,1)
\end{sseqdata}
\printpage[name=orphan edges example]
\hskip1cm
\printpage[name=orphan edges example,no orphan edges]
```

**class placement transform**= $\langle \text{transform keys} \rangle$  (no default)

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 (x,y) position. This coordinate transform can only involve rotation and scaling, no translation. Specifying a scaling factor helps if the nodes are too large and overlap. In some cases a rotation makes it easier to see which class is the target of a differential.



```
\begin{sseqpage}[classes={draw=none},class placement transform={xscale=3},
xscale=2, x axis extend end=0.7cm]
\class["$\mathbb{Z}$"] (0,0)
\class["$\mathbb{Z}/2$"] (1,1)
\class["$\mathbb{Z}/3$"] (1,1)
\end{sseqpage}
```



```
\begin{sseqpage}[classes=fill,class placement transform={rotate=40},
cohomological Serre grading,differentials=blue,scale=0.7]
\class(0,0)
\class(0,2)\class(0,2)
\class[red](3,0)\class[green](3,0)\class[blue](3,0)
\d3(0,2,1,2)
\d3(0,2,-1,-1)
\draw[->,red](3,0,1)--(0,0);
\end{sseqpage}
```

### 3.3 Global Coordinate Transformations

Of the normal TikZ coordinate transformations, only the following are allowed to be applied to a sseq diagram:

**xscale**= $\langle \text{factor} \rangle$  (no default)



<code>yscale=⟨factor⟩</code>	(no default)
<code>xmirror</code>	(no value)
<code>ymirror</code>	(no value)

<code>rotate=⟨angle⟩</code>	(no default)
-----------------------------	--------------

### 3.4 Layout

<code>custom clip=</code>	(no default)
---------------------------	--------------

<code>clip=⟨boolean⟩</code>	(default true)(initially true)
-----------------------------	--------------------------------

<code>x axis gap=⟨dimension⟩</code>	(no default, initially 0.5cm)
<code>y axis gap=⟨dimension⟩</code>	(no default, initially 0.5cm)
<code>axes gap=⟨dimension⟩</code>	(no default, initially 0.5cm)

<code>x label gap=⟨dimension⟩</code>	(no default, initially 0.5cm)
<code>y label gap=⟨dimension⟩</code>	(no default, initially 0.5cm)

<code>x axis start extend=⟨dimension⟩</code>	(no default, initially 0.5cm)
<code>y axis start extend=⟨dimension⟩</code>	(no default, initially 0.5cm)
<code>x axis end extend=⟨dimension⟩</code>	(no default, initially 0.9cm)
<code>y axis end extend=⟨dimension⟩</code>	(no default, initially 0.9cm)

<code>x clip axis padding=⟨dimension⟩</code>	(no default, initially 0.1cm)
<code>y clip axis padding=⟨dimension⟩</code>	(no default, initially 0.1cm)

<code>right clip padding=⟨dimension⟩</code>	(no default, initially 0.1cm)
<code>left clip padding=⟨dimension⟩</code>	(no default, initially 0.4cm)
<code>top clip padding=⟨dimension⟩</code>	(no default, initially 0.1cm)
<code>bottom clip padding=⟨dimension⟩</code>	(no default, initially 0.4cm)

### 3.5 Axes Style

<code>x axis style=a</code>	(no default, initially border)
<code>y axis style=a</code>	(no default, initially border)
<code>axes style=</code>	(no default, initially border)

<code>x axis origin=</code>	(no default, initially 0)
<code>y axis origin=</code>	(no default, initially 0)

<code>no x axis</code>	(no value)
<code>no y axis</code>	(no value)
<code>no axes</code>	(no value)
<code>draw x axis</code>	(no value)
<code>draw y axis</code>	(no value)
<code>draw axes</code>	(no value)

<code>no x axis labels</code>	(no value)
<code>no y axis labels</code>	(no value)

<code>no axes labels</code>	(no value)
<code>draw x axis labels</code>	(no value)
<code>draw y axis labels</code>	(no value)
<code>draw axes labels</code>	(no value)
<code>x label step=</code>	(no default, initially 1)
<code>y label step=</code>	(no default, initially 1)
<code>label step=</code>	(no default, initially 1)
<code>rotate labels=&lt;boolean&gt;</code>	(default true)(initially false)