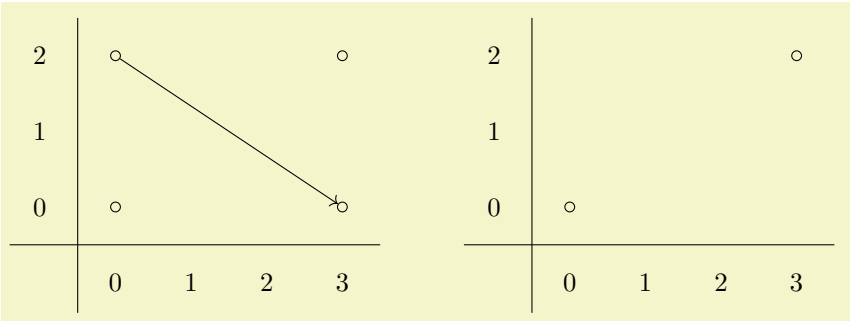
1 Introduction

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 enivornment with an empty body.

Here is a basic example:



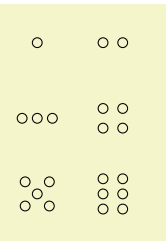

```
\begin{sseqdata}[name=ex1,cohomological Serre grading]
\class(0,0)
\class(0,2)
\class(3,0)
\class(3,2)
\d3(0,2)
\end{sseqdata}
\printpage[name=ex1,page=3]\hskip1cm
\printpage[name=ex1,page=4]
```

`\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

`\class[`⟨*options*⟩`]`⟨*coordinate*⟩

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



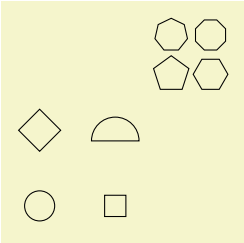
```
\begin{sseqpage}[no axes,ymirror]
\class(0,0)
\class(1,0)\class(1,0)
\class(0,1)\class(0,1)\class(0,1)
\class(1,1)\class(1,1)\class(1,1)\class(1,1)
\class(0,2)\class(0,2)\class(0,2)\class(0,2)\class(0,2)
\class(1,2)\class(1,2)\class(1,2)\class(1,2)\class(1,2)\class(1,2)
\end{sseqpage}
```

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

A Ti*k*Z shape

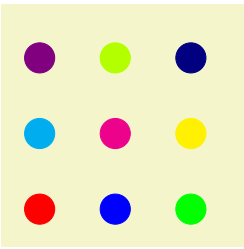
If you give the name of a Ti*k*Z shape, the class node will be of that shape. The standard Ti*k*Z shapes are `circle` and `rectangle`, but there are many more Ti*k*Z shapes in the shapes library,

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}
```

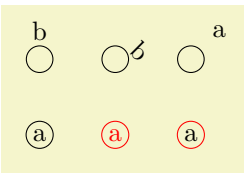
A Ti*k*Z color



```
\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*⟩

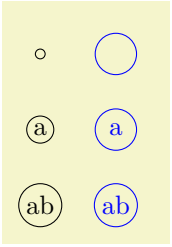
A label. This uses the Ti*k*Z quotes syntax, but the behavior specific to `sseqpages`. By default, the ⟨*text*⟩ is placed in the position `inside` the node – in effect, the ⟨*text*⟩ becomes the label text of the node (so saying `\class["label text"](0,0)` causes a similar effect to saying `\node at (0,0) {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 Ti*k*Z 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}
```

`minimum width`=⟨*dimension*⟩ (no default)
`minimum height`=⟨*dimension*⟩ (no default)
`minimum size`=⟨*dimension*⟩ (no default)
`inner sep`=⟨*dimension*⟩ (no default)
`outer sep`=⟨*dimension*⟩ (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:

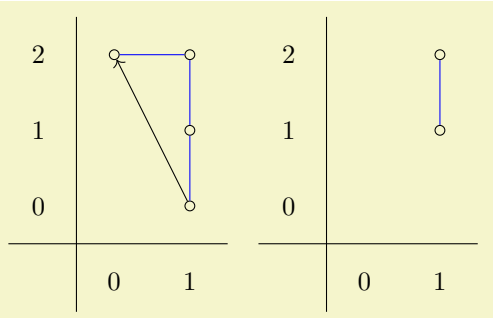



```
\begin{sseqdata}[no axes,name=minimum width example]
\class["ab"](0,0)
\class["a"](0,1)
\class(0,2)
\end{sseqdata}
\printpage[name=minimum width example]
\printpage[name=minimum width example,
           change classes={blue,minimum width=width("ab")+0.5em}]
```

For more information, see the pgf manual.

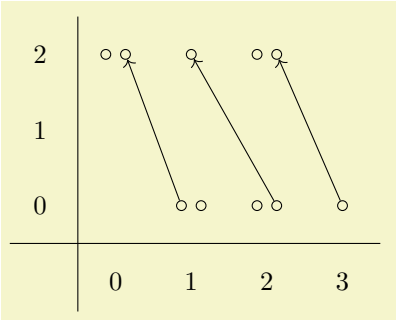
`\d`[⟨*options*⟩]⟨*page*⟩⟨*source coordinate*⟩

This creates a differential starting at ⟨*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 `\structline`s connected to the source and target of the differential disappear.



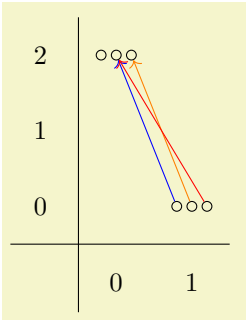

```
\begin{sseqdata}[name=d example,degree={-1}{#1},
                 struct lines=blue]
\class(0,2)
\class(1,2)
\class(1,1)
\class(1,0)
\structline(1,2)(0,2)
\structline(1,2)(1,1)
\structline(1,1)(1,0)
\d2(1,0)
\end{sseqdata}
\printpage[name=d example,page=2]
\hskip0.3cm
\printpage[name=d example,page=3]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

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*⟩`(`⟨x⟩`,`⟨y⟩`,`⟨*source* n⟩`,`⟨*target* n⟩`)`




```
\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):

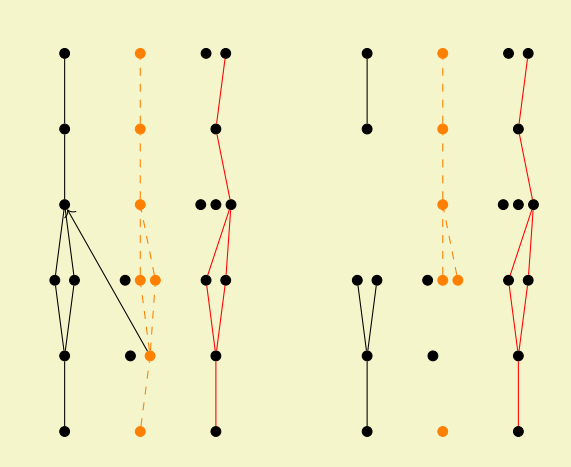



```
\begin{sseqpage}[Adams grading]
\class(1,0)
\class(0,2)\class(0,2)
\d[blue]2(1,0,-1,-1)
\class(1,0)
\class(0,2)
\d[orange]2(1,0,-1,-1)
\class(1,0)
\d[red]2(1,0,-1,-2)
\end{sseqpage}
```

`\structline`[⟨*options*⟩]⟨*source coordinate*⟩⟨*target coordinate*⟩

This command creates a structure line from ⟨*source coordinate*⟩ to ⟨*target coordinate*⟩. The source and target coordinates are of the form `(`⟨x⟩`,`⟨y⟩`,`⟨n⟩`)`. If there are multiple classes at (x, y), then ⟨n⟩ 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.

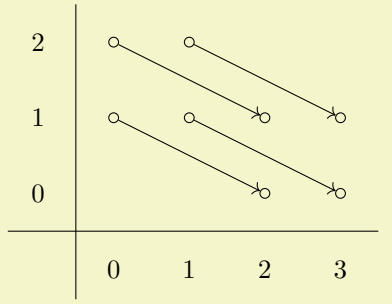
If the source or target of a structure line is hit by a differential, then on subsequent pages, the structure line disappears.



```
\seqnewgroup\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{seqdata}[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{seqdata}
\printpage[name=structline example,page=2]
\hskip1cm
\printpage[name=structline example,page=3]
```

`xshift`=⟨*integer*⟩ (no default)
`yshift`=⟨*integer*⟩ (no default)

Shifts by integer values are the only local coordinate changes that are allowed to be applied to `\class`, `\d`, and `\structline`. These shift commands help with reusing code. For instance:




```
\begin{sseqpage}[cohomological Serre grading]
\foreach \x in {0,1} \foreach \y in {0,1}{
    \begin{scope}[xshift=\x,yshift=\y]
    \class(2,0)
    \class(0,1)
    \d2(0,1)
    \end{scope}
}
\end{sseqpage}
```

A word of warning: the behavior of `xshift` in `sseqpages` is incompatible with the normal behavior of xshift in Ti*k*Z. For some reason, saying `xshift=1` in Ti*k*Z does not shift the coordinate `(0,0)` to the coordinate `(1,0)` – instead it shifts a tiny fraction of the distance.

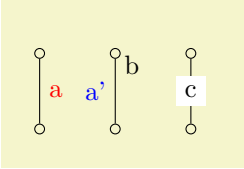
2.1 Options for \d and \structline

In general, any option that you could apply to a Ti*k*Z "to" command can be applied to both `\d` and `\structline`. Some such options are as follows:

`"`⟨*text*⟩`"'`⟨*options*⟩

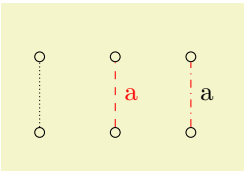
A label `"`⟨*text*⟩`"'`⟨*options*⟩. By default, such a label is placed to the right of the edge. The optional prime places it to the left of the edge instead. The options include anything you might pass as an option to a Ti*k*Z node, including arbitrary coordinate transforms, colors, opacity options, 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*.



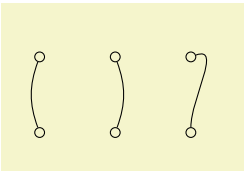

```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1,2} \foreach\y in {0,1}{
    \class(\x,\y)
}
\structline["a" red](0,0)(0,1)
\structline["a'"'blue,"b"{yshift=1em}](1,0)(1,1)
\structline["c" description](2,0)(2,1)
\end{sseqpage}
```

Colors and dash patterns:



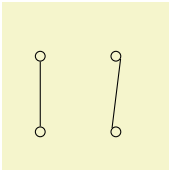

```
\begin{sseqpage}[background color=graphicbackground, no axes]
\foreach\x in {0,1,2} \foreach\y in {0,1}{
    \class(\x,\y)
}
\structline[densely dotted](0,0)(0,1)
\structline[dashed,red, "a"](1,0)(1,1)
\structline[dash dot,red, "a" black](2,0)(2,1)
\end{sseqpage}
```

bend left=⟨*angle*⟩ (no default)
bend right=⟨*angle*⟩ (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.

\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

`\printpage`[⟨*options*⟩]

This command prints a single page of an existing spectral sequence as-is. This is equivalent to a `sseqpage` environment with an empty body.

3.1 Global options

`name`=⟨*sseq name*⟩ (no default)

`page`=⟨*page number*⟩ (no default, initially `0`)

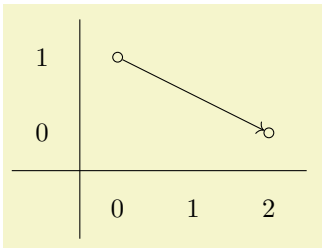
This key is for `sseqpage` and `\printpage`. It specifies which page of the spectral sequence is to be printed. On page r, all `\class`es that are not hit by differentials on pages less than r will be printed, as well as all `\structline`s whose source and target classes are both printed on page r, and all differentials of length exactly r. The special value `page=`*`0`* prints all classes, differentials, and structure lines.

`degree`={⟨*x degree*⟩}{⟨*y degree*⟩} (no default)
`cohomological Serre grading` (no value)
`homological Serre grading` (no value)
`Adams grading` (no value)

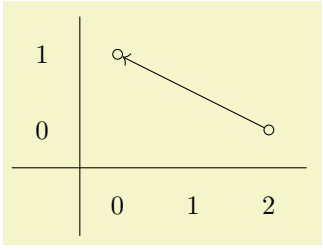
Specifies the degree of differentials. The ⟨*x degree*⟩ and ⟨*y degree*⟩ should both be mathematical expressions in one variable `#1` that evaluate to integers on any input. They specify the x and y displacement of a page `#1` differential. In practice, they will usually be linear expressions with `#1` 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={`⟨*x degree*⟩`}{`⟨*y degree*⟩`}}` or `\sseqset{Adams grading}` outside of the `sseqdata` and `sseqpages` environments.

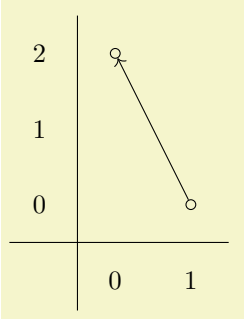
You can make a named grading convention by saying `\sseqset{my grading/.sseq grading={`⟨*x degree*⟩`}{`⟨*y degree*⟩`}}`. Then later passing `my grading` to a spectral sequence is equivalent to saying `degree={`⟨*x degree*⟩`}{`⟨*y degree*⟩`}`. The following grading conventions exist by default:




```
\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`=⟨*x min*⟩⟨*x max*⟩ (no default)
`y range`=⟨*y min*⟩⟨*y max*⟩ (no default)

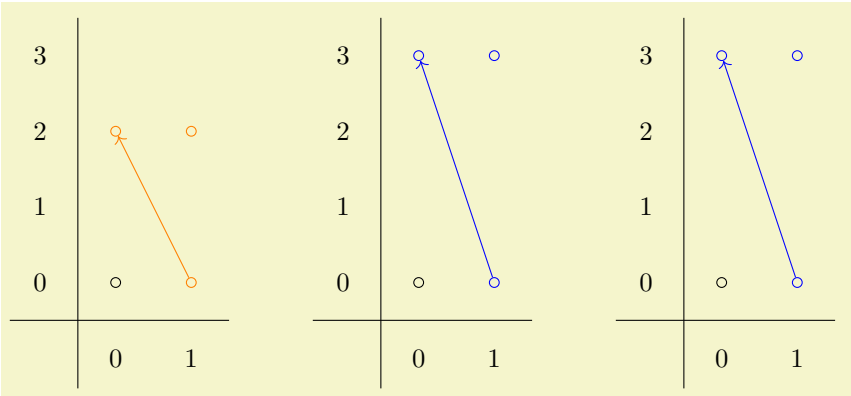
These options force the x and y range to be a specific interval. By default, if no range is specified then the range is chosen to fit all the classes. If an x range is specified but no y range, then the y range is chosen to fit all the classes that lie inside the specified x range, and vice versa.

`update existing` (no value)

This key is only for the `sseqdata` environment. It specifies that the current `sseqdata` environment is adding data to an 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. This is intended to help you avoid accidentally reusing the same name.

`keep changes`=⟨*boolean*⟩ (default `true`)(`initially false`)

This option is only for the `sseqpage` environment, and only works when a `name` is provided. This option specifies that all of the commands in the current `sseqpage` environment should be carried forward to future pages of the same named 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.

`no differentials` (no value)
`draw differentials` (no value)

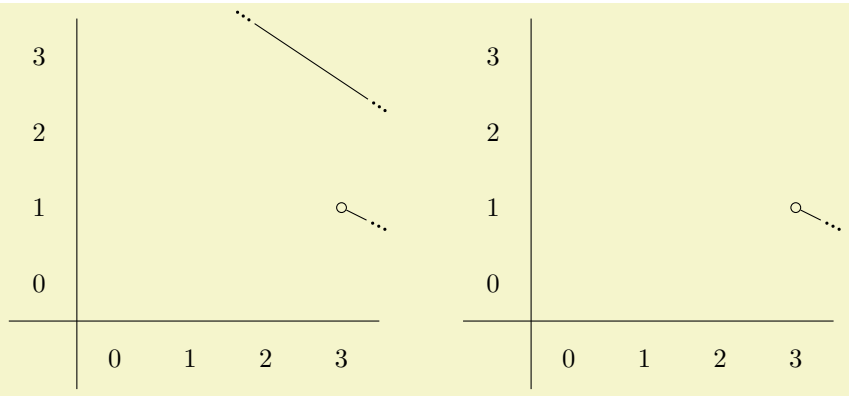
The option `no differentials` suppresses all of the differentials on the current page, whereas `draw differentials` causes the page appropriate differentials to be drawn. This is useful for explaining how the computation of the spectral sequence goes:

`no struct lines` (no value)
`draw struct lines` (no value)

The option `no struct lines` suppresses all of the differentials on the current page, whereas the option `draw struct lines` causes the page appropriate differentials to be drawn.

`no orphan edges` (no value)
`draw orphan edges=`⟨*boolean*⟩ (default `true`) (initially `true`)

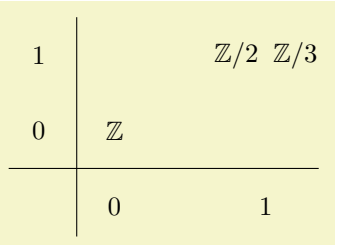
An edge is an "orphan" if both its source and target lie off the page. By default these are drawn, but with the option `no orphan edges` they are not. If the option `no orphan edges` has been set, `draw orphan edges` undoes it.



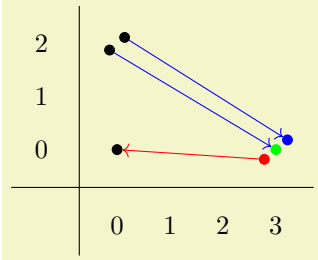
```
\begin{sseqdata}[name=orphan edges example,cohomological Serre grading,
                 x range={0}{3}, y range={0}{3}]
\class(1,4)\class(4,2)
\d3(1,4)
\class(3,1)\class(5,0)
\d2(3,1)
\end{sseqdata}
\printpage[name=orphan edges example]
\hskip1cm
\printpage[name=orphan edges example,no orphan edges]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

`class placement transform`={⟨*transform keys*⟩} (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}
```

3.2 Styles

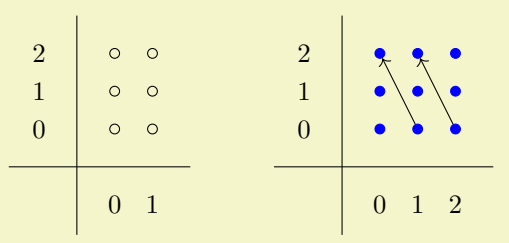
The `sseqpages` package has a large number of "styles" which control the appearance of specific components (classes, differentials, or structlines) of a spectral sequence. These are named so that each command has a plural variant (e.g., `classes`) and a "style" variant (e.g., `class style`). The difference between these is always that `classes=\marg{keys}` adds the keys to the list of options used to style every class, whereas `class style=\marg{keys}` overwrites the list of options. It's important to be aware when using the style variants that some of the styles are not empty when `sseqpages` is loaded, so for instance saying `class style={}` will change the appearance of the diagram. Generally, the plural versions are more useful, but in very large diagrams it can be noticeably faster to use the style variants.

In cases where the same drawing feature is affected by multiple of these styles, the more specific style takes precedence.

Throughout, "class" and "cycle" are synonyms.

`sseqs`={⟨*keys*⟩} (no default)
`sseq style`={⟨*keys*⟩} (no default)

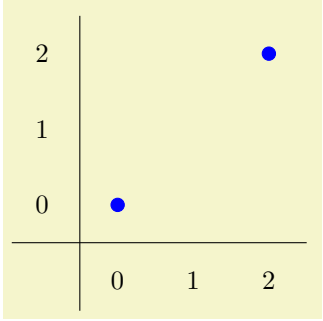
This passes global options to all future spectral sequences in the current scope. It is only useful to use this command with `\sseqset`. This is only really important for Ti*k*Z options, because for most options from `sseqpages` you can set a default directly by saying `\sseqset`{key={⟨*value*⟩}}.




```
\sseqset{sseqs={scale=0.5}}% Applies to both of the two following sseqs
\begin{sseqpage}
\foreach \x in {0,1} \foreach \y in {0,1,2}{
    \class(\x,\y)
}
\end{sseqpage}
\hskip1cm
\begin{sseqpage}[Adams grading,classes={fill,blue}]
\foreach \x in {0,1,2} \foreach \y in {0,1,2}{
    \class(\x,\y)
}
\d2(1,0)
\d2(2,0)
\end{sseqpage}
```

`classes={`⟨*keys*⟩`}` (no default)
`cycles={`⟨*keys*⟩`}` (no default)
`class style={`⟨*keys*⟩`}` (no default)
`cycle style={`⟨*keys*⟩`}` (no default)

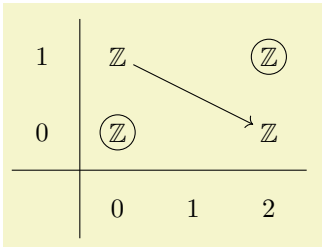
The keys `classes` and `cycles` are synonyms as are `class style` and `cycle style`. These options change the appearance of all classes. The options `classes` and `cycles` append whatever keys you give to the list of class style options, whereas `class style` and `cycle style` overwrite the list of styles.




```
\begin{sseqpage}[classes={blue,fill,minimum width=0.5em}]
\class(0,0)
\class(2,2)
\end{sseqpage}
```

`permanent classes={`⟨*keys*⟩`}` (no default)
`permanent cycles={`⟨*keys*⟩`}` (no default)
`permanent class style={`⟨*keys*⟩`}` (no default)
`permanent cycle style={`⟨*keys*⟩`}` (no default)

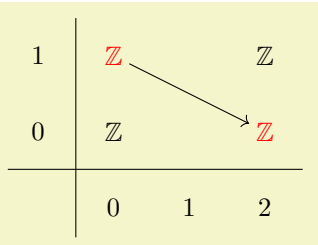
These options change the appearance of all permanent cycles (e.g., those classes which never support or are hit by a differential). For instance, we can circle the permanent cycles automatically. Note that because `permanent cycles` is more specific than `classes`, the `permanent cycles={draw}` command wins out over the `class={draw=none}` command to insure that the permanent cycle nodes are drawn.




```
\begin{sseqpage}[cohomological Serre grading,
                 classes={draw=none},permanent cycles={draw}]
\foreach \x in {0,2} \foreach \y in {0,1}{
    \class["$\mathbb{Z}$"](\x,\y)
}
\d2(0,1)
\end{sseqpage}
```

`transient classes={⟨keys⟩}` (no default)
`transient cycles={⟨keys⟩}` (no default)
`transient class style={⟨keys⟩}` (no default)
`transient cycle style={⟨keys⟩}` (no default)

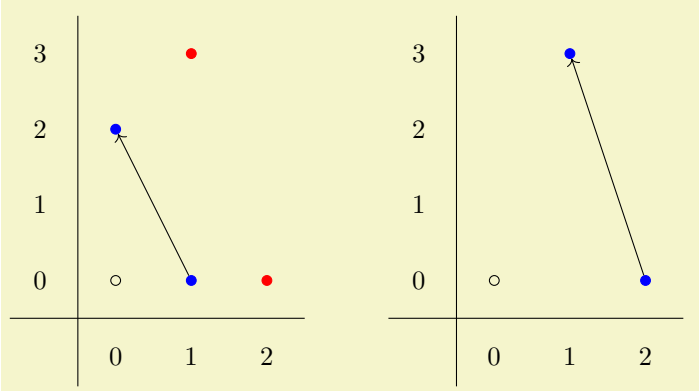
These options change the appearance of all transient cycles (e.g., those classes which eventually support or are hit by a differential). Again, this takes precedence over the `classes` option.




```
\begin{sseqpage}[cohomological Serre grading,
                 classes={draw=none}, transient cycles=red]
\foreach \x in {0,2} \foreach \y in {0,1}{
    \class["$\mathbb{Z}$"](\x,\y)
}
\d2(0,1)
\end{sseqpage}
```

`this page classes={⟨keys⟩}` (no default)
`this page cycles={⟨keys⟩}` (no default)
`this page class style={⟨keys⟩}` (no default)
`this page cycle style={⟨keys⟩}` (no default)

These options change the appearance of all cycles which support or are hit by a differential on this page. Any class that is hit on the current page is also a transient cycle, and so `this page classes` takes precedence over `transient cycles`




```
\begin{sseqdata}[name=this page cycles example,Adams grading,
                 transient cycles={red,fill},this page cycles={blue}]
\class(0,0)
\class(0,2)\class(1,0)
\class(1,3)\class(2,0)
\d2(1,0)\d3(2,0)
\end{sseqdata}
\printpage[name=this page cycles example,page=2]
\hskip1cm
\printpage[name=this page cycles example,page=3]
```

`edges={⟨keys⟩}` (no default)
`edge style={⟨keys⟩}` (no default)

This style applies to both differentials and structure lines. The `differentials` and `struct lines` keys both take precedence over `edges`.

`differentials={⟨keys⟩}` (no default)
`differential style={⟨keys⟩}` (no default)

`struct lines={⟨keys⟩}` (no default)

`struct line style={`⟨*keys*⟩`}` (no default)

`this page struct lines={`⟨*keys*⟩`}` (no default)
`this page struct line style={`⟨*keys*⟩`}` (no default)

This style applies to structure lines whose source or target is hit on the current page. It takes precedence over `struct lines`.

`labels` (no value)
`label style` (no value)

`class labels` (no value)
`class label style` (no value)

`inner class labels` (no value)
`inner class label style` (no value)

`outer class labels` (no value)
`outer class label style` (no value)

`edge labels` (no value)
`edge label style` (no value)

`differential labels` (no value)
`differential label style` (no value)

`struct line labels` (no value)
`struct line label style` (no value)

3.3 Global Coordinate Transformations

Of the normal Ti*k*Z coordinate transformations, only the following can be applied to a sseq diagram:

`scale=`⟨*factor*⟩ (no default)
`xscale=`⟨*factor*⟩ (no default)
`yscale=`⟨*factor*⟩ (no default)
`xmirror` (no value)
`ymirror` (no value)

Scale the diagram by ⟨*factor*⟩. Under normal circumstances, you can tell Ti*k*Z to mirror a diagram by saying, for instance, `xscale=-1`, but `sseqpages` needs to be aware that the diagram has been mirrored in order to draw the axes correctly. Thus, if you want to mirror a spectral sequence, use the `xmirror` and `ymirror` options as appropriate.

`rotate=`⟨*angle*⟩ (no default)

It probably won't look great if you pick an angle that isn't a multiple of 90 degrees.

3.4 Layout

`custom clip=`⟨*clip path*⟩ (no default)

`clip=`⟨*boolean*⟩ (default `true`)(`initially true`)

`x axis gap=`⟨*dimension*⟩ (no default, initially `0.5cm`)
`y axis gap=`⟨*dimension*⟩ (no default, initially `0.5cm`)
`axes gap=`⟨*dimension*⟩ (no default, initially `0.5cm`)

`x label gap=`⟨*dimension*⟩ (no default, initially `0.5cm`)
`y label gap=`⟨*dimension*⟩ (no default, initially `0.5cm`)

`x axis start extend=`⟨*dimension*⟩ (no default, initially `0.5cm`)
`y axis start extend=`⟨*dimension*⟩ (no default, initially `0.5cm`)
`x axis end extend=`⟨*dimension*⟩ (no default, initially `0.9cm`)
`y axis end extend=`⟨*dimension*⟩ (no default, initially `0.9cm`)

`x clip axis padding=`⟨*dimension*⟩ (no default, initially `0.1cm`)
`y clip axis padding=`⟨*dimension*⟩ (no default, initially `0.1cm`)

`right clip padding=`⟨*dimension*⟩ (no default, initially `0.1cm`)
`left clip padding=`⟨*dimension*⟩ (no default, initially `0.4cm`)
`top clip padding=`⟨*dimension*⟩ (no default, initially `0.1cm`)
`bottom clip padding=`⟨*dimension*⟩ (no default, initially `0.4cm`)

3.5 Axes Style

`x axis style=`a (no default, initially `border`)
`y axis style=`a (no default, initially `border`)
`axes style=` (no default, initially `border`)

`x axis origin=` (no default, initially `0`)
`y axis origin=` (no default, initially `0`)

`no x axis` (no value)
`no y axis` (no value)
`no axes` (no value)
`draw x axis` (no value)
`draw y axis` (no value)
`draw axes` (no value)

`no x axis labels` (no value)
`no y axis labels` (no value)
`no axes labels` (no value)
`draw x axis labels` (no value)
`draw y axis labels` (no value)
`draw axes labels` (no value)

`x label step=` (no default, initially `1`)
`y label step=` (no default, initially `1`)
`label step=` (no default, initially `1`)

`rotate labels=`⟨*boolean*⟩ (default `true`)(`initially false`)

4 Commands

\sseqset{⟨*keys*⟩}

\sseqnewcmd

\sseqnewgroup