Package `pointbox`

Alice M.

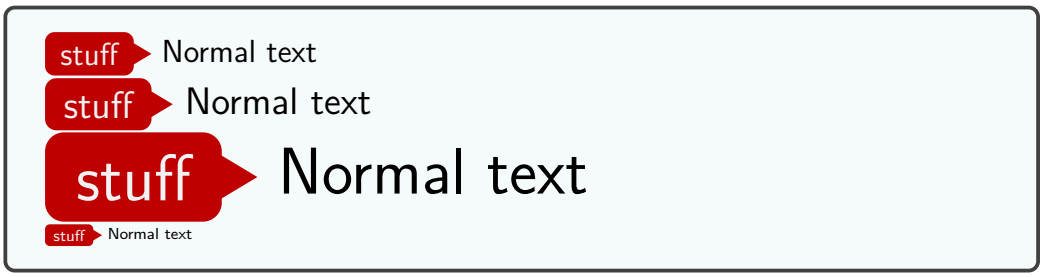
12 June 2017

(Since this package was initially thought for slide-shows, this documentation will be mostly using a sans-serif font from here onwards. You do not necessarily have to do the same.)

1 Basics

The main command is `\pointbox[⟨options⟩]{⟨content⟩}`.

By default, the content uses the current font size. Also, most default dimensions are proportional to the font size. Here is a bunch of
`\pointbox{stuff} Normal text`
in different contexts:

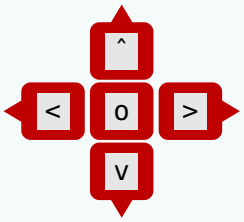


2 Options

The optional parameter of `\pointbox` is parsed using the `xkeyval` package, which means that it is a set of comma-separated key-value pairs, with some keys that can be used without specifying a value.

The options `^`, `>`, `v` and `<` can be used to control the position of the point. `o` (this is the letter "o") removes the point altogether.

`\pointbox[<]{Text}`, etc.



Colours can be managed via the `fg` ("foreground") and `bg` ("background") parameters:

`fg = green, bg = cyan` Colours example

The font can be altered by passing commands via the `font` parameter:

`font = \ttfamily\bfseries\itshape` ***Font example***

The position of the point can be altered relatively to the normal position using the `shift` parameter:

One Two One Two

One, two One, two

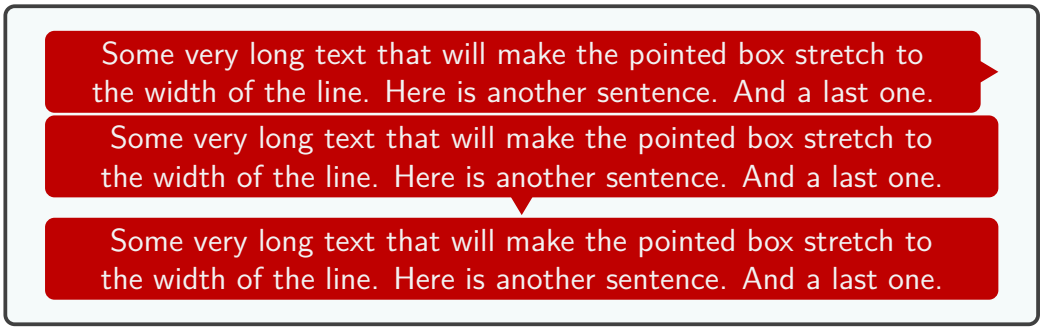
A minimal width can be enforced using `minwidth`, mostly to make sure several boxes have the same width (it might look better this way):

A ABCDEF

Note that the minimal width is fixed on the box, without taking the point into account:

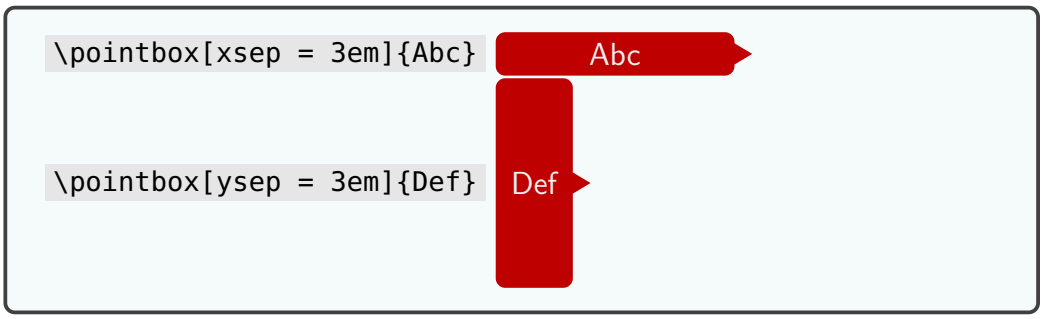
A point No point

The default maximal width is `\linewidth`. This takes the point into account to prevent overfull hboxes, but only if the point is drawn and if it is placed on the side, as shown in the following examples:



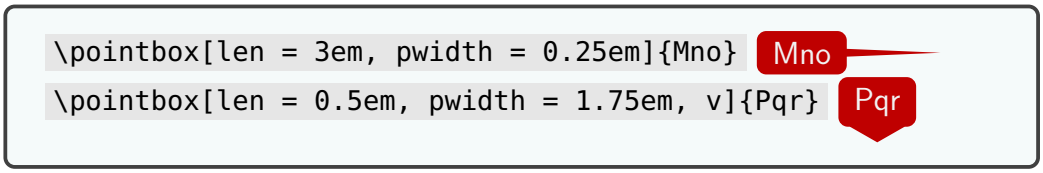
A different maximal width can be enforced using `maxwidth`. Here are two boxes using `maxwidth = 0.5\linewidth` in order to fit nicely on a line:

Inner horizontal and vertical padding can be controlled via `xsep` and `ysep`, respectively:

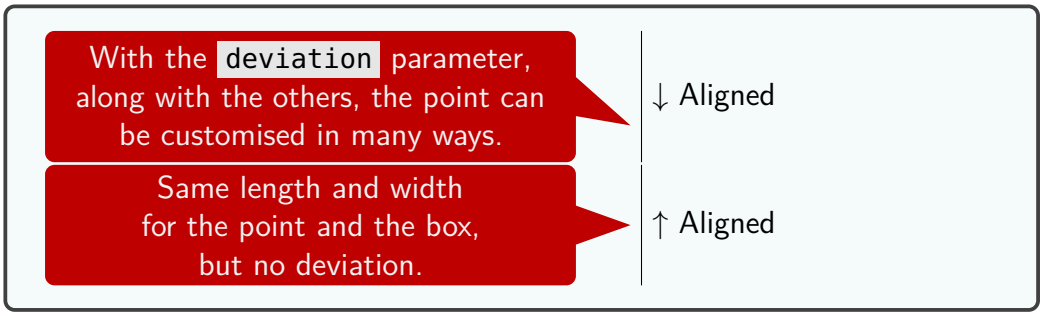


The roundedness of the corners is set with `radius`:

The length and width of the point are set via `len` and `pwidth` respectively:



The `deviation` parameter can be used to skew the point. Note that the length is computed perpendicularly from the edge of the block.



To set a parameter for all subsequent pointboxes, you can use: `\pbxset{⟨options⟩}`. Note that if this command is enclosed in a group[1], the changes are lost at the end of the group (which is generally what people want): `{\pbxset{bg = blue, o}\pointbox{1}\pointbox{2}}\pointbox{3}`
Output: 1 2 3

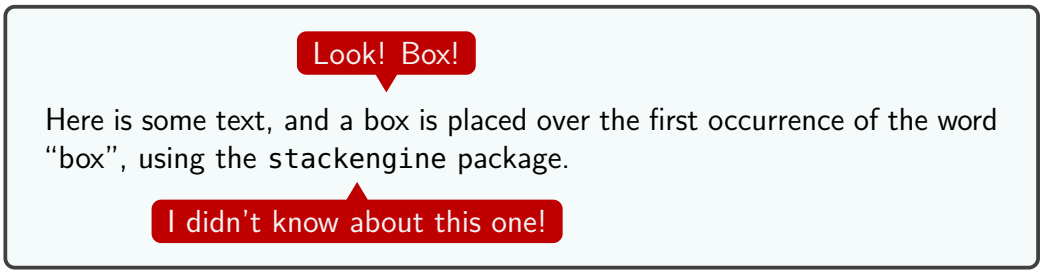
3 Placement

Pointed boxes might look good when placed in the margin using the standard `marginpar` command, but you might get warnings if you override the maximal width in order to get bigger boxes. The command for this example is:

Margin pointbox

`\marginpar{\pointbox[<, maxwidth = 7em]{Margin pointbox}}`

Here are a few other placement possibilities:



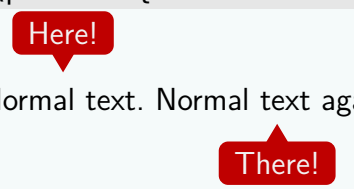
Since this might be annoying to use, I provide two commands for the most common cases:

[1] `{ ... }` or `\begingroup ... \endgroup`

`\pointtop[⟨options⟩]{⟨outer text⟩}{⟨box text⟩}` and `\pointbot[⟨options⟩]{⟨outer text⟩}{⟨box text⟩}`.

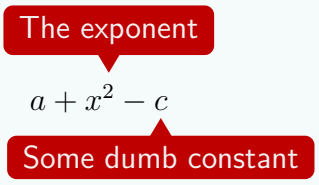
Here is a simple example:

```
\pointtop{Normal text.}{Here!}
\pointbot{Normal text again.}{There!}
```

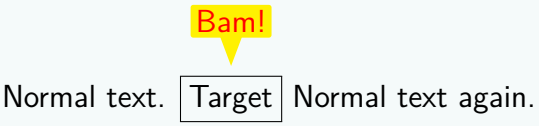


Mixing it with maths and stuff might be tricky depending on what you are trying to achieve, but it is possible:

```
x\pointtop{\ensuremath{^2}}{The exponent}, etc.
```



Parameters can still be used through the optional argument to change the aspect of the box:



By using the `wrapfig` package, you can place boxes in the text like this. The alignment and width can be chosen; just take a look at the code to see how it can be done. The placement of such "figures" can be tricky, though, so check the documentation of `wrapfig` if you run into trouble. Now here is some more text just to make the example more realistic. Here is an additional sentence, which has been artificially prolonged. And here are the final words.