Extension witharrows, v. 1.0

François Pantigny

2017/08/17

Abstract

The LaTeX package `witharrows` gives an environment `{WithArrows}` which is similar to environment `{aligned}` of `amsmath` (and `mathtools`) but gives the possibility to draw arrows on the right side of the alignment. These arrows are usually used to give explanations concerning the mathematical calculus presented.

This package uses Tikz to draw the arrows. It can be used with `xelatex`, `pdflatex` but also by the classical workflow `latex-dvips-ps2pdf` (or Distiller). Two compilations may be necessary.

This package gives an environment `{WithArrows}` to construct alignments of equations with arrows for the explanations on the right side :

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow{we expand} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \quad \text{we expand}$$

The arrow has been drawn with the command `\Arrow` on the ligne from which it starts. The command `\Arrow` can be used anywhere on the line but the best way is to put it at the end.

The ends of the lines of the tabular can be coded by `\\` but also by `\cr` (like in the command `\halign` of TeX). In fact, in an environment `{WithArrows}`, the command `\\` has no option for the spacing between rows like in many environments of LaTeX.

1 Options for the shape of the arrows

The commande `\Arrow` has several options. These options can be put between square brackets, before, or after the mandatory argument.
The option `jump` gives the number of lines the arrow must jump (the default value is, of course, 1) [1]

```
$\begin{WithArrows}
A & = \bigl((a+b)+1\bigr)^2 \Arrow[jump=2]{we expand} \\
  & = (a+b)^2 + 2(a+b) +1 \\
  & = a^2 + 2ab + b^2 + 2a + 2b +1 \\
\end{WithArrows}$
```

$$\begin{aligned} A &= \bigl((a+b)+1\bigr)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \end{aligned}$$

we expand

It's possible to put several arrows which start from the same line.

```
$\begin{WithArrows}
A & = \bigl((a+b)+1\bigr)^2 \Arrow{}\Arrow{}[jump=2] \\
  & = (a+b)^2 + 2(a+b) +1 \\
  & = a^2 + 2ab + b^2 + 2a + 2b +1
\end{WithArrows}$
```

$$\begin{aligned} A &= \bigl((a+b)+1\bigr)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \end{aligned}$$

The option `xoffset` shift the arrows to the right (we usually don't want the arrows to be stucked on the text). The default value of `xoffset` is 3 mm.

```
$\begin{WithArrows}
A & = \bigl((a+b)+1\bigr)^2
\Arrow[xoffset=1cm]{with \texttt{xoffset=1cm}} \\
  & = (a+b)^2 + 2(a+b) +1
\end{WithArrows}$
```

$$\begin{aligned} A &= \bigl((a+b)+1\bigr)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \end{aligned}$$

with `xoffset=1cm`

The arrows are drawn with Tikz. That's why the command `\Arrow` has an option `tikz` which can be used to give to the arrow (in fact, the command `\path` of Tikz)

[1] It's not possible to give a non-positive value to `jump`. See below the way to draw an arrow which goes backwards.

the options proposed by Tikz for such an arrow. The following example gives an blue thick arrow.

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow[tikz={blue,thick}]{we expand} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \quad \text{we expand}$$

It's also possible to change the arrowheadss. For example, we can draw an arrow which goes backwards with the Tikz option `<-`.

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow[tikz=<-]{we factorize} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \quad \text{we factorize}$$

It's also possible to suppress both tips of the arrow with the Tikz option `-`.

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow[tikz=-]{very classical} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \quad \text{very classical}$$

In order to have straight arrows instead of curved ones, we must use the Tikz option "`bend left = 0`".

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow[tikz={bend left=0}]{we expand} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \quad \text{we expand}$$

One of the most useful options is "`text width`" to control the with of the text associated to the arrow.

```
$\begin{WithArrows}
A & = \bigl((a+b)+1\bigr)^2
\Arrow[jump=2,tikz={text width=5.3cm}]{We have done...} \\
  & = (a+b)^2 + 2(a+b) +1 \\
  & = a^2 + 2ab + b^2 + 2a + 2b +1
\end{WithArrows}$
```

$$\begin{aligned} A &= \bigl((a+b)+1\bigr)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \end{aligned}$$

We have done a two-stages expansion but it would have been clever to expand with the multinomial theorem.

If we want to change the font of the text associated to the arrow, we can, of course, put a command like `\bfseries`, `\large` or `\sffamily` at the beginning of the text. But, by default, the texts are composed with a combination of `\small` and `\itshape`. When adding `\bfseries` at the beginning of the text, we won't suppress the `\small` and the `\itshape` and we will consequently have a text in a bold, italic and small font.

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow{\bfseries we expand} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned}$$

we expand

If we put commands `\\` in the text to force newlines, a command of font placed in the beginning of the text will have effect only until the first command `\\` (like in a environment `{tabular}`). That's why Tikz gives a option `font` to modify the font of the whole text. Nevertheless, if we use the option `tikz={font={\bfseries}}`, the default specification of `\small` and `\itshape` will be overwritten.

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow[tikz={font={\bfseries}}]{we expand} \\
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned}$$

we expand

If we want exactly the same result as previously, we have to give to the option `font` the value `{\itshape\small\bfseries}`.

Almost all the options can be given directly to the environment `WithArrows` (between square brackets). In this case, they apply to all the arrows of the environment.

```
$\begin{WithArrows}[tikz=blue]
A & = \bigl((a+b)+1\bigr)^2 \Arrow{First expansion.} \\
  & = (a+b)^2 + 2(a+b) +1 \Arrow{Second expansion.} \\
  & = a^2 + 2ab + b^2 + 2a + 2b +1
\end{WithArrows}$
```

$$\begin{aligned} A &= \bigl((a+b)+1\bigr)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \qquad \text{First expansion.} \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \qquad \text{Second expansion.} \end{aligned}$$

The environment `{WithArrows}` has an option `displaystyle`. With this option, all the elements are composed in `\displaystyle` (like in an environment `{aligned}` of `amsmath`).

Without the option `displaystyle` :

```
$\begin{WithArrows}
\int_0^1 (x+1)^2 dx
& = \int_0^1 (x^2+2x+1) dx
\Arrow{linearity of integration}     \\
& = \int_0^1 x^2 dx + 2 \int_0^1 x dx + \int_0^1 dx \\
& = \frac13 + 2\frac12 + 1 \\
& = \frac73
\end{WithArrows}$
```

$$\begin{aligned} \int_0^1 (x+1)^2 dx &= \int_0^1 (x^2+2x+1) dx \\ &= \int_0^1 x^2 dx + 2\int_0^1 x dx + \int_0^1 dx \qquad \text{linearity of integration} \\ &= \tfrac13 + 2\tfrac12 + 1 \\ &= \tfrac73 \end{aligned}$$

The same example with the option `displaystyle` :

$$\begin{aligned} \int_0^1 (x+1)^2 dx &= \int_0^1 (x^2+2x+1) dx \\ &= \int_0^1 x^2 dx + 2\int_0^1 x dx + \int_0^1 dx \qquad \text{linearity of integration} \\ &= \frac13 + 2\frac12 + 1 \\ &= \frac73 \end{aligned}$$

Almost all the options can also be set at the document level with the command `\WithArrowsOptions`. In this case, the scope of the declarations is the current TeX group (these declarations are “semi-global”). For example, if we want all the

environments `{WithArrows}` composed in `\displaystyle` with blue arrows, we can write `\WithArrowsOptions{displaystyle,tikz=blue}`.[2]

```
\WithArrowsOptions{displaystyle,tikz=blue}
$\begin{WithArrows}
\sum_{i=1}^n (x_i+1)^2
& = \sum_{i=1}^n (x_i^2+2x_i+1) \Arrow{by linearity}\\
& = \sum_{i=1}^n x_i^2 + 2\sum_{i=1}^nx_i+ n
\end{WithArrows}$
```

$$\begin{aligned}\sum_{i=1}^{n}(x_i+1)^2 &= \sum_{i=1}^{n}(x_i^2+2x_i+1)\\ &= \sum_{i=1}^{n}x_i^2+2\sum_{i=1}^{n}x_i+n \end{aligned}$$

by linearity

The command `\Arrow` is recognized only in the environments `{WithArrows}`. If we have a command `\Arrow` previously defined, it's possible to go on using it outside the environments `{WithArrows}`.
However, a previouly defined command `\Arrow` may still be useful in a environment `{WithArrows}`. If we want to use it in such an environment, it's possible to change the name of the command `\Arrow` of the package `witharrows` : there is an option `CommandName` for this purpose. The new name of the command must be given to the option *without* the leading backslash.

```
\def\Arrow{\longmapsto}
$\begin{WithArrows}[CommandName=Explanation]
f & = \bigl(x \Arrow (x+1)^2\bigr)
\Explanation{we work directly on fonctions}\\
& = \bigl(x \Arrow x^2+2x+1\bigr)
\end{WithArrows}$
```

$$\begin{aligned}f &= \bigl(x \longmapsto (x+1)^2\bigr)\\ &= \bigl(x \longmapsto x^2+2x+1\bigr)\end{aligned}$$

we work directly on fonctions

It's possible to use directly the nodes created by `{WithArrows}` (see below) with explicit Tikz instructions (in order, for example, to draw something that can't be drawn with the command `\Arrow`). That's why a style for the tips of the arrows has be created : `TipsOfWithArrows`. By using this style, we will have homogeneous tips for the arrows of the document.
Therefore, if we want to modify the tips of the arrows of `{WithArrows}`, we have to modify the style `TipsOfWithArrows`.

```
\tikzset{TipsOfWithArrows/.style= { > = {Latex[scale=1.2,bend]}} }
```

[2] It's also possible to give the options directly when loading the package, *i.e.* with the command `\usepackage` in the preamble.

2 Precise positioning of the arrows

The environment `{WithArrows}` defines, during the composition of the array, two series of nodes materialized in red in the following example.[3]

$$
\begin{aligned}
I &= \int_{\frac{\pi}{4}}^{0} \ln\Big(1+\tan\big(\tfrac{\pi}{4}-u\big)\Big)(-du) \\
&= \int_{0}^{\frac{\pi}{4}} \ln\Big(1+\tan\big(\tfrac{\pi}{4}-u\big)\Big)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(1+\frac{1-\tan u}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(\frac{1+\tan u+1-\tan u}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(\frac{2}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \big(\ln 2-\ln(1+\tan u)\big)\,du \\
&= \frac{\pi}{4}\ln 2-\int_{0}^{\frac{\pi}{4}} \ln(1+\tan u)\,du \\
&= \frac{\pi}{4}\ln 2-I
\end{aligned}
$$

The nodes of the left are at the end of each line of text. These nodes will be called *left nodes*. The nodes of the right side are aligned vertically on the right side of the array. These nodes will be called *right nodes*.[4]
By default, the arrows use the right nodes. We will say that they are in `rr` mode (*r* for *right*). These arrows are `vertical` (we will say that an arrow is *vertical* when its two ends have the same abscissa).

However, it's possible to use the left nodes, or a combination of left and right nodes, with one of the options `lr`, `rl` and `ll` (*l* for *left*). Those arrows are, usually, not vertical.

[3] The option `shownodes` can be used to materialize the nodes.

[4] The names of the Tikz nodes created by `{WithArrows}` are `wa-`*n*`-l` and `wa-`*n*`-r` where *n* is the number of the line. It's possible to refer to these Tikz nodes after the environment (one should use the options `remember picture` and `overlay` and also `TipsOfWithArrows` and `->` in order to have the same arrowheads).

$$
\begin{aligned}
\text{Therefore } I &= \int_{\frac{\pi}{4}}^{0} \ln\Big(1+\tan\big(\tfrac{\pi}{4}-u\big)\Big)(-du) \\
&= \int_{0}^{\frac{\pi}{4}} \ln\Big(1+\tan\big(\tfrac{\pi}{4}-u\big)\Big)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(1+\frac{1-\tan u}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(\frac{1+\tan u+1-\tan u}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \ln\left(\frac{2}{1+\tan u}\right)\,du \\
&= \int_{0}^{\frac{\pi}{4}} \big(\ln 2-\ln(1+\tan u)\big)\,du \\
&= \frac{\pi}{4}\ln 2-\int_{0}^{\frac{\pi}{4}} \ln(1+\tan u)\,du \\
&= \frac{\pi}{4}\ln 2 - I
\end{aligned}
$$

This arrow uses the `lr` *option.*

This arrow uses a `ll` *option and a* `jump` *equal to* 2

There is also an option called `i` (*i* for *intermediate*). With this option, the arrow is vertical and at the leftmost position.

```
$\begin{WithArrows}
(a+b)(a+ib)(a-b)(a-ib)
& = (a+b)(a-b)\cdot(a+ib)(a-ib) \\
& = (a^2-b^2)(a^2+b^2) \Arrow[i]{because $(x-y)(x+y)=x^2-y^2$}\\
& = a^4-b^4
\end{WithArrows}$
```

$$
\begin{aligned}
(a+b)(a+ib)(a-b)(a-ib) &= (a+b)(a-b)\cdot(a+ib)(a-ib) \\
&= (a^2-b^2)(a^2+b^2) \\
&= a^4-b^4
\end{aligned}
$$

because $(x-y)(x+y)=x^2-y^2$

The environment `{WithArrows}` gives also a `group` option. With this option, *all* the arrows of the environment are grouped on a same vertical line and at a leftmost position.

```
$\begin{WithArrows}[displaystyle,group]
2xy'-3y=\sqrt x
& \Longleftrightarrow 2x(K'y_0+Ky_0')-3Ky_0 = \sqrt x \\
& \Longleftrightarrow 2xK'y_0 + K(2xy_0'-3y_0) = \sqrt x \\
& \Longleftrightarrow 2x K'y_0 = \sqrt x \Arrow{...}\\
...
\end{WithArrows}$
```

$$
\begin{aligned}
2xy' - 3y = \sqrt{x} &\iff 2x(K'y_0 + Ky_0') - 3Ky_0 = \sqrt{x} \\
&\iff 2xK'y_0 + K(2xy_0' - 3y_0) = \sqrt{x} \\
&\iff 2xK'y_0 = \sqrt{x} \qquad \text{We remplace } y_0 \text{ by its value.} \\
&\iff 2xK'x^{\frac{3}{2}} = x^{\frac{1}{2}} \qquad \text{simplification of the } x \\
&\iff K' = \tfrac{1}{2x\check{s}} \qquad \text{antiderivation} \\
&\iff K = -\tfrac{1}{2x}
\end{aligned}
$$

If desired, the option `group` can be given to the command `WithArrowsOptions` so that it will become the default value.

3 Comparison with the environment {aligned}

`{WithArrows}` bears similarities with the environment `{aligned}` of the extension `amsmath`. These are only similarities because `{WithArrows}` has not been written upon the environment `{aligned}`.
In paticular, the command `\\` of the end of line has no option to change the row spacings. In fact, this command `\\` can be replaced, in a environment `{WithArrows}` by a `\cr` of TeX. That's what we will do in the balance of this document.

If desired, it's possible to change the spacing between two given lines by putting a command `\vspace` in a `\noalign` (which is a low level command of TeX to insert extraordinary elements between the lines of an array).

```
$\begin{WithArrows}
A & = (a+1)^2 \Arrow{we expand} \cr
\noalign{\vspace{3mm}}
  & = a^2 + 2a + 1
\end{WithArrows}$
```

$$
\begin{aligned}
A &= (a+1)^2 \\
&= a^2 + 2a + 1 \qquad \text{we expand}
\end{aligned}
$$

In the environments of `amsmath`, the spacing between lines is fixed by a parameter called `\jot` and that's also the case for the environment `{WithArrows}`. An option `jot` has been given to the environment `{WithArrows}` in order to change the value of this parameter `\jot` for an given environment.

```
$\begin{WithArrows}[displaystyle,jot=2mm]
F & = \frac12G     \Arrow{we expand}\cr
  & = H + \frac12K \Arrow{we go on}\cr
```

```
  & = K
\end{WithArrows}$
```

$$\begin{WithArrows}
F &= \frac{1}{2}G \\
&= H + \frac{1}{2}K \Arrow{we expand} \\
&= K \Arrow{we go on}
\end{WithArrows}$$

Like the environment `{aligned}`, `{WithArrows}` has an option of placement which can assume the values `t`, `c` or `b`. However, the default value is not `c` but `t`. If desired, it's possible to have the `c` value as the default with the command `WithArrowsOptions{c}` at the beginning of the document.

```
Et donc\enskip
$\begin{WithArrows}
A & = (a+1)^2 \Arrow{we expand} \cr
  & = a^2 + 2a + 1
\end{WithArrows}$
```

Et donc $\begin{WithArrows} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{WithArrows}$ *we expand*

The value `c` may be useful, for example, if we want to add curly braces :

```
On pose\enskip $\left\{
\begin{WithArrows}[c]
f(x) & = 3x^3+2x^2-x+4
\Arrow[tikz=-]{both are polynoms}\cr
g(x) & = 5x^2-5x+6
\end{WithArrows}
\right.$
```

On pose $\left\{\begin{WithArrows}[c] f(x) &= 3x^3+2x^2-x+4 \\ g(x) &= 5x^2-5x+6 \end{WithArrows}\right.$ *both are polynoms*

Unlike `{aligned}`, the environment `{WithArrows}` uses `\textstyle` by default. Once again, it's possible to change this behaviour with `WithArrowsOptions` : `WithArrowsOptions{displaystyle}`.

The following example is composed with `{aligned}` :

$$\left\{\begin{aligned}
\sum_{i=1}^{n}(x_i+1)^2 &= \sum_{i=1}^{n}(x_i^2+2x_i+1) \\
&= \sum_{i=1}^{n}x_i^2 + 2\sum_{i=1}^{n}x_i + n
\end{aligned}\right.$$

The following is composed with `{WithArrows}[c,displaystyle]`. The results are stricly identical.

$$\begin{cases} \displaystyle\sum_{i=1}^{n}(x_i+1)^2 = \sum_{i=1}^{n}(x_i^2+2x_i+1) \\ \qquad\qquad\quad = \displaystyle\sum_{i=1}^{n}x_i^2 + 2\sum_{i=1}^{n}x_i + n \end{cases}$$

4 Examples

It's possible to use the environment `{WithArrows}` with making use of the left column only, or the right column only.

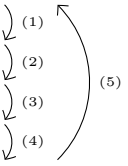
```
$\begin{WithArrows}
&f(x) \ge g(x) \Arrow{by squaring both sides} \cr
& f(x)^2 \ge g(x)^2 \Arrow{by moving to left side} \cr
& f(x)^2 - g(x)^2 \ge 0
\end{WithArrows}$
```

$f(x) \ge g(x)$
$f(x)^2 \ge g(x)^2$ ⤵ *by squaring both sides*
$f(x)^2 - g(x)^2 \ge 0$ ⤵ *by moving to left side*

Here is an example with a loop flow.

```
$\begin{WithArrows}[tikz={font={\tiny}}]
a.\;& f \text{ est continuous on } E
\Arrow{(1)}\Arrow[tikz=<-,jump=4,xoffset=1cm]{(5)}\cr
b.\;& f \text{ est continuous in } 0
\Arrow{(2)}\cr
c.\;& f \text{ is bounded on the unit sphere}
\Arrow{(3)}\cr
d.\;& \exists K > 0\quad \forall x \in E\quad \|f(x)\| \le K \|x\|
\Arrow{(4)}\cr
e.\;& f \text{ is lipschitzian}
\end{WithArrows}$
```

a. f est continuous on E
b. f est continuous in 0
c. f is bounded on the unit sphere
d. $\exists K > 0 \quad \forall x \in E \quad \|f(x)\| \le K\|x\|$
e. f is lipschitzian



The option `font` of Tikz contains in fact a list of tokens which will be placed at the beginning of the text.

These tokens can be true commands for a changement of font (like `\bfseries` or `\sffamily`) but can also be, in fact, any TeX command.
In the following example, the argument of `font` is the token list `\tiny\counter` where `\counter` is a command which increment a counter previously defined and display its new value. Thus, the arrows are automatically numbered.

```
\newcounter{MyCounter}
\newcommand{\counter}{\stepcounter{MyCounter}\theMyCounter.}
$\begin{WithArrows}[tikz={font={\tiny\counter}}]
A(x)
& = B(x) \Arrow{} \cr
& = C(x) \Arrow{} \cr
& = C(x) \Arrow{} \cr
& = E(x) \Arrow{} \cr
& = F(x) \Arrow{} \cr
& = G(x)
\end{WithArrows}$
```

$$\begin{aligned} A(x) &= B(x) \\ &= C(x) \quad \overset{1.}{\downarrow} \\ &= C(x) \quad \overset{2.}{\downarrow} \\ &= E(x) \quad \overset{3.}{\downarrow} \\ &= F(x) \quad \overset{4.}{\downarrow} \\ &= G(x) \quad \overset{5.}{\downarrow} \end{aligned}$$