

L'extension `witharrows` pour plain-TeX et LaTeX*

F. Pantigny
fpantigny@wanadoo.fr

19 octobre 2024

Résumé

L'extension `witharrows` fournit des environnements `{WithArrows}` et `{DispWithArrows}` similaires aux environnements `{aligned}` et `{align}` de l'extension `amsmath` mais avec la possibilité de dessiner des flèches sur le côté droit. Ces flèches sont habituellement utilisées pour donner des explications concernant le calcul mathématique présenté.

L'extension LaTeX `witharrows` est entièrement contenue dans le fichier `witharrows.sty`. Ce fichier peut être placé dans le répertoire courant ou dans une arborescence `texmf`. Le mieux reste néanmoins d'installer `witharrows` avec une distribution TeX comme MiKTeX, TeX Live ou MacTeX.

En fait, `witharrows` est aussi une extension utilisable avec plain-TeX et dans ce cas, le seul fichier nécessaire est le fichier `witharrows.tex` : voir à ce sujet p. 24. Dans la suite, on décrira l'extension LaTeX.

Cette extension peut être utilisée avec `xelatex`, `lualatex`, `pdflatex` mais aussi avec le cheminement classique `latex-dvips-ps2pdf` (ou Adobe Distiller). Néanmoins, le fichier `witharrows-french.tex` de la présente documentation ne peut être compilé qu'avec `LuaLaTeX`. L'extension `witharrows` charge les extensions `l3keys2e`, `tikz`, `varwidth` ainsi que les bibliothèques Tikz `arrows.meta` et `bending`. L'utilisateur final n'a qu'à charger l'extension `witharrows` avec l'instruction habituelle : `\usepackage{witharrows}`.

Les flèches sont tracées avec Tikz et donc **plusieurs compilations peuvent être nécessaires**.¹

Cette extension fournit un environnement `{WithArrows}` pour construire des alignements d'équations avec des flèches pour les explications sur le côté droit.

```
\begin{WithArrows}
A & = (a+1)^2 \Arrow{on développe} \\
& = a^2 + 2a + 1 % <----- ne pas mettre de \\ ici
\end{WithArrows}
```

$$\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned} \left. \vphantom{\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned}} \right\} \textit{on développe}$$

La flèche a été tracée avec la commande `\Arrow` dans la rangée dont la flèche part. La commande `\Arrow` doit être utilisée dans la seconde colonne (le mieux est de la mettre à la fin de la ligne comme dans l'exemple précédent).

L'environnement `{WithArrows}` est proche de l'environnement `{aligned}` de `amsmath`. L'extension `witharrows` propose aussi des environnements `{DispWithArrows}` et `{DispWithArrows*}` qui sont similaires aux environnements `{align}` et `{align*}` de l'extension `amsmath` : cf. p. 18.

*Ce document correspond à la version 2.9a de `witharrows`, à la date du 2024/10/19.

1. Si vous utilisez Overleaf, Overleaf effectue automatiquement un nombre de compilations suffisant (en utilisant `latexmk`).

1 Options pour la forme des flèches

La commande `\Arrow` a plusieurs options. Ces options peuvent être placées entre crochets, avant, ou après, l'argument obligatoire.

L'option `jump` indique le nombre² de rangées que la flèche doit sauter (la valeur initiale est, bien entendu, 1).

```


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


```

$$\begin{aligned}
 A &= ((a+b)+1)^2 \\
 &= (a+b)^2 + 2(a+b) + 1 \\
 &= a^2 + 2ab + b^2 + 2a + 2b + 1
 \end{aligned}
 \left. \vphantom{\begin{aligned} A &= ((a+b)+1)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \end{aligned}} \right) \textit{on développe}$$

Il est possible de faire partir plusieurs flèches d'une même rangée.

```


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


```

$$\begin{aligned}
 A &= ((a+b)+1)^2 \\
 &= (a+b)^2 + 2(a+b) + 1 \\
 &= a^2 + 2ab + b^2 + 2a + 2b + 1
 \end{aligned}
 \left. \vphantom{\begin{aligned} A &= ((a+b)+1)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \\ &= a^2 + 2ab + b^2 + 2a + 2b + 1 \end{aligned}} \right) \textit{on développe}$$

L'option `xoffset` décale la flèche vers la droite (habituellement, on ne souhaite pas que les flèches soient collées au texte). La valeur initiale de `xoffset` est de 3 mm.

```


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


```

$$\begin{aligned}
 A &= ((a+b)+1)^2 \\
 &= (a+b)^2 + 2(a+b) + 1
 \end{aligned}
 \left. \vphantom{\begin{aligned} A &= ((a+b)+1)^2 \\ &= (a+b)^2 + 2(a+b) + 1 \end{aligned}} \right) \textit{avec } xoffset=1cm$$

Les flèches sont tracées avec Tikz. C'est pourquoi la commande `\Arrow` a une option `tikz` qui peut être utilisée pour donner à la flèche (en fait, à la commande `\path` de Tikz) les options proposées par Tikz pour une telle flèche. L'exemple suivant fournit une flèche en trait épais.

```


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


```

$$\begin{aligned}
 A &= (a+1)^2 \\
 &= a^2 + 2a + 1
 \end{aligned}
 \left. \vphantom{\begin{aligned} A &= (a+1)^2 \\ &= a^2 + 2a + 1 \end{aligned}} \right) \textit{on développe}$$

2. Il n'est pas possible de donner une valeur négative à `jump`. Voir plus loin (p. 3) la manière de tracer une flèche qui remonte.

Il est également possible de changer les pointes de flèche. Par exemple, nous pouvons tracer une flèche qui remonte vers le haut avec l'option Tikz <-.

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

$$A = (a + 1)^2 \quad \nearrow \textit{on factorise}$$

$$= a^2 + 2a + 1$$

Il est aussi possible de supprimer les deux pointes de flèche avec l'option Tikz « - ».

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

$$A = (a + 1)^2 \quad \rightarrow \textit{très classique}$$

$$= a^2 + 2a + 1$$

Pour avoir des flèches droites et non incurvées, il convient d'utiliser l'option Tikz « bend left = 0 ».

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

$$A = (a + 1)^2 \quad \downarrow \textit{on développe}$$

$$= a^2 + 2a + 1$$

En fait, il est possible de modifier de manière plus drastique la forme des flèches avec l'option `tikz-code` (présentée p. 24).

Il est possible d'utiliser l'option « text width » pour contrôler la largeur du texte associé à la flèche.

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

$$A = ((a + b) + 1)^2 \quad \left. \begin{array}{l} \textit{Nous avons développé en deux étapes} \\ \textit{mais il aurait été plus habile} \\ \textit{d'utiliser la formule du multinôme.} \end{array} \right\}$$

$$= (a + b)^2 + 2(a + b) + 1$$

$$= a^2 + 2ab + b^2 + 2a + 2b + 1$$

Dans les environnements `{DispWithArrows}` et `{DispWithArrows*}`, il y a une option `wrap-lines`. Lorsque cette option est utilisée, les lignes des étiquettes sont automatiquement coupées sur la marge droite : voir p. 18.

Si on veut changer la fonte du texte associé à une flèche, on peut, bien entendu, placer une commande comme `\bfseries`, `\large` ou `\sffamily` au début du texte. Mais, par défaut, les étiquettes sont composées avec une combinaison de `\small` et `\itshape`. En ajoutant `\bfseries` au début du texte, on ne va pas supprimer le `\small` et le `\itshape` et, par conséquent, on aura un texte en gras, italique et petite taille.

```

 $\begin{WithArrows}$ 
A & = (a+1)^2 \Arrow{\bfseries on développe} \\
& = a^2 + 2a + 1 \\
\end{WithArrows}

```

$$\begin{aligned}
 A &= (a+1)^2 \\
 &= a^2 + 2a + 1 \quad \left. \vphantom{A} \right\} \textit{on développe}
 \end{aligned}$$

Il est possible de placer des commandes `\` dans le texte pour forcer des retours à la ligne³. Néanmoins, si on utilise des commandes `\`, une instruction de fonte placée au début du texte aura un effet seulement jusqu'à la première occurrence de `\` (comme dans un environnement `{tabular}`). C'est pourquoi Tikz fournit une option `font` pour modifier la fonte de tout le texte de l'étiquette. Cette fois-ci, si on utilise une option `tikz={font={\bfseries}}`, la spécification par défaut constituée par `\small` et `\itshape` va être écrasée.

```

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

```

$$\begin{aligned}
 A &= (a+1)^2 \\
 &= a^2 + 2a + 1 \quad \left. \vphantom{A} \right\} \textit{on développe}
 \end{aligned}$$

Si on veut exactement le même résultat que précédemment, on doit donner à l'option `font` la valeur `\itshape\small\bfseries`.

Presque toutes les options peuvent être données entre crochets à l'environnement `{WithArrows}`. Il ne doit pas y avoir d'espace entre le `\begin{WithArrows}` et le crochet ouvrant (`[`) des options de l'environnement. Ces options s'appliquent à toutes les flèches de l'environnement.⁴

```

 $\begin{WithArrows}[tikz=blue]$ 
A & = \bigl((a+b)+1\bigr)^2 \Arrow{premier développement} \\
& = (a+b)^2 + 2(a+b) + 1 \Arrow{second développement} \\
& = a^2 + 2ab + b^2 + 2a + 2b + 1 \\
\end{WithArrows}

```

$$\begin{aligned}
 A &= ((a+b)+1)^2 \\
 &= (a+b)^2 + 2(a+b) + 1 \quad \left. \vphantom{A} \right\} \textit{premier développement} \\
 &= a^2 + 2ab + b^2 + 2a + 2b + 1 \quad \left. \vphantom{A} \right\} \textit{second développement}
 \end{aligned}$$

L'environnement `{WithArrows}` a une option `displaystyle`. Avec cette option, tous les éléments sont composés en `\displaystyle` (comme dans un environnement `{aligned}` de l'extension `amsmath`).

Sans l'option `displaystyle` :

```

 $\begin{WithArrows}$ 
\int_0^1 (x+1)^2 dx
& = \int_0^1 (x^2+2x+1) dx \\
\Arrow{linéarité de l'intégration} & \\
& = \int_0^1 x^2 dx + 2 \int_0^1 x dx + \int_0^1 dx \\
& = \frac{1}{3} + 2\frac{1}{2} + 1 \\
& = \frac{7}{3} \\
\end{WithArrows}

```

3. Par défaut, ce n'est pas possible dans un nœud Tikz. Néanmoins, dans `witharrows`, les nœuds sont construits avec l'option `align=left`, et, ainsi, cela devient possible.

4. Elles s'appliquent aussi aux environnements imbriqués dans un environnement `{WithArrows}` donné (avec des exceptions attendues pour les options `interline`, `code-before` et `code-after`).

$$\begin{aligned} \int_0^1 (x+1)^2 dx &= \int_0^1 (x^2 + 2x + 1) dx \\ &= \int_0^1 x^2 dx + 2 \int_0^1 x dx + \int_0^1 dx \quad \left. \vphantom{\int_0^1} \right\} \text{linéarité de l'intégration} \\ &= \frac{1}{3} + 2\frac{1}{2} + 1 \\ &= \frac{7}{3} \end{aligned}$$

Le même exemple avec l'option `displaystyle` :

$$\begin{aligned} \int_0^1 (x+1)^2 dx &= \int_0^1 (x^2 + 2x + 1) dx \\ &= \int_0^1 x^2 dx + 2 \int_0^1 x dx + \int_0^1 dx \quad \left. \vphantom{\int_0^1} \right\} \text{linéarité de l'intégration} \\ &= \frac{1}{3} + 2\frac{1}{2} + 1 \\ &= \frac{7}{3} \end{aligned}$$

Presque toutes les options peuvent aussi être fixées au niveau du document avec la commande `\WithArrowsOptions`. Dans ce cas, la portée des déclarations est le groupe TeX courant (de telles déclarations sont parfois qualifiées de « semi-globales »). Par exemple, si nous voulons que tous les environnements `{WithArrows}` soient composés en `\displaystyle` avec des flèches bleues, nous pouvons écrire `\WithArrowsOptions{displaystyle,tikz=blue}`.⁵

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

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

La commande `\Arrow` est reconnue seulement dans les environnements de `witharrows`. Si on dispose d'une commande `\Arrow` précédemment définie, il est possible de continuer à l'utiliser à l'extérieur des environnements de `witharrows`.

Néanmoins, une commande `\Arrow` définie précédemment pourrait encore être utile dans un environnement `{WithArrows}`. Si vous voulez l'utiliser dans un tel environnement, il est possible de changer le nom de la commande `\Arrow` de l'extension `witharrows` grâce à une option `command-name` dédiée. Le nouveau nom de la commande doit être fourni à l'option `sans` la contre-oblique.

```
\NewDocumentCommand {\Arrow} {} {\longmapsto}
$\begin{WithArrows}[-emphase#command-name=Explication@]
f & = \bigl(x \ \Arrow (x+1)^2\bigr)
\Explication{on travaille directement sur les fonctions}\
& = \bigl(x \ \Arrow x^2+2x+1\bigr)
\end{WithArrows}$
```

$$\begin{aligned} f &= (x \longmapsto (x+1)^2) \\ &= (x \longmapsto x^2 + 2x + 1) \quad \left. \vphantom{f} \right\} \text{on travaille directement sur les fonctions} \end{aligned}$$

L'environnement `{WithArrows}` fournit aussi deux options `code-before` et `code-after` pour du code LaTeX qui sera exécuté au début et à la fin de l'environnement. Ces options ne sont pas conçues

5. Il est aussi possible de configurer `witharrows` en modifiant le style Tikz `WithArrows/arrow` qui est le style utilisé par `witharrows` lors du tracé d'une flèche. Par exemple, pour avoir les étiquettes en bleu et en caractères droits, on peut utiliser l'instruction suivante : `\tikzset{WithArrows/arrow/.append style = {blue,font = {}}}`.

pour être utilisées comme des *hooks* (elles sont disponibles uniquement au niveau de l'environnement et ne s'appliquent pas aux environnements imbriqués).

```
\begin{WithArrows}[code-before = \color{blue}]
A & = (a+b)^2 \Arrow{on développe} \\
& = a^2 + 2ab + b^2 \\
\end{WithArrows}$
```

$$\begin{array}{l} A = (a + b)^2 \\ = a^2 + 2ab + b^2 \end{array} \left. \vphantom{\begin{array}{l} A = (a + b)^2 \\ = a^2 + 2ab + b^2 \end{array}} \right\} \textit{on développe}$$

Des commandes spéciales sont disponibles dans le `code-after` : une commande `\WithArrowsNbLines` qui indique le nombre de lignes (=rangées) de l'environnement courant (pour TeX, il s'agit d'une commande et non d'un compteur), une forme spéciale de la commande `\Arrow` et la commande `\MultiArrow` : ces deux dernières commandes sont décrites à partir de la page 14.

2 Nombre et formats des colonnes

Jusqu'à présent, nous n'avons utilisé l'environnement `{WithArrows}` qu'avec deux colonnes. Néanmoins, il est possible de l'utiliser avec un nombre arbitraire de colonnes grâce à l'option `format`. La valeur donnée à cette option est similaire au préambule d'un environnement `{array}` : il s'agit d'une séquence de lettres `r`, `c` et `l` mais aussi `R`, `C` et `L`.

Les lettres `R`, `C` et `L` placent des groupes vides `{}` qui permettent un espacement correct lorsque ces colonnes contiennent des symboles de type `\mathrel` (comme `=`, `≤`, etc.) ou `\mathbin` (comme `+`, `×`, etc.). Ce système est inspiré par l'environnement `{IEEEeqnarray}` de `IEEEtrantools`.

La valeur initiale du paramètre `format` est, en fait, `rL`.

Par exemple, si on veut seulement une colonne alignée à gauche, on peut utiliser l'option `format=l`.

```
\begin{WithArrows}[format = l]
f(x) \ge g(x) \Arrow{on élève les deux membres au carré} \\
f(x)^2 \ge g(x)^2 \Arrow{on fait tout passer à gauche} \\
f(x)^2 - g(x)^2 \ge 0 \\
\end{WithArrows}$
```

$$\begin{array}{l} f(x) \geq g(x) \\ f(x)^2 \geq g(x)^2 \\ f(x)^2 - g(x)^2 \geq 0 \end{array} \left. \vphantom{\begin{array}{l} f(x) \geq g(x) \\ f(x)^2 \geq g(x)^2 \\ f(x)^2 - g(x)^2 \geq 0 \end{array}} \right\} \begin{array}{l} \textit{on élève les deux membres au carré} \\ \textit{on fait tout passer à gauche} \end{array}$$

Dans l'exemple suivant, on utilise cinq colonnes toutes centrées (l'environnement `{DispWithArrows*}` utilisé est présenté p. 18).

```
\begin{DispWithArrows*}[format = cCcCc,
                        wrap-lines,
                        interline=1mm]
k & \; \leq; & t & \; \leq; & k+1 \\
\frac{1}{k+1} & \leq & \frac{1}{t} & \leq & \frac{1}{k} \\
\Arrow{nous pouvons intégrer les inégalités puisque $k \leq k+1$ } \\
\int\limits_k^{k+1} \frac{dt}{k+1} \\
& \leq & \int\limits_k^{k+1} \frac{dt}{t} \\
& \leq & \int\limits_k^{k+1} \frac{dt}{k} \\
\frac{1}{k+1} & \leq & \ln(k+1) - \ln(k) & \leq & \frac{1}{k} \\
\end{DispWithArrows*}
```

$$\begin{array}{rcccl}
k & \leq & t & \leq & k+1 \\
\frac{1}{k+1} & \leq & \frac{1}{t} & \leq & \frac{1}{k} \\
\int_k^{k+1} \frac{dt}{k+1} & \leq & \int_k^{k+1} \frac{dt}{t} & \leq & \int_k^{k+1} \frac{dt}{k} \\
\frac{1}{k+1} & \leq & \ln(k+1) - \ln(k) & \leq & \frac{1}{k}
\end{array}
\left. \vphantom{\begin{array}{rcccl} k & \leq & t & \leq & k+1 \\ \frac{1}{k+1} & \leq & \frac{1}{t} & \leq & \frac{1}{k} \\ \int_k^{k+1} \frac{dt}{k+1} & \leq & \int_k^{k+1} \frac{dt}{t} & \leq & \int_k^{k+1} \frac{dt}{k} \\ \frac{1}{k+1} & \leq & \ln(k+1) - \ln(k) & \leq & \frac{1}{k} \end{array}} \right\} \text{ nous pouvons intégrer les } \\
& & & & \text{ inégalités puisque } k \leq k+1$$

3 Positionnement précis des flèches

L'environnement `{WithArrows}` construit, lors de la composition du tableau, deux séries de nœuds représentés en rouge dans l'exemple suivant.⁶

$$\begin{aligned}
I &= \int_{\frac{\pi}{4}}^0 \ln\left(1 + \tan\left(\frac{\pi}{4} - u\right)\right) (-du) && \cdot \\
&= \int_0^{\frac{\pi}{4}} \ln\left(1 + \tan\left(\frac{\pi}{4} - u\right)\right) du && \cdot \\
&= \int_0^{\frac{\pi}{4}} \ln\left(1 + \frac{1 - \tan u}{1 + \tan u}\right) du && \cdot \\
&= \int_0^{\frac{\pi}{4}} \ln\left(\frac{1 + \tan u + 1 - \tan u}{1 + \tan u}\right) du && \cdot \\
&= \int_0^{\frac{\pi}{4}} \ln\left(\frac{2}{1 + \tan u}\right) du && \cdot \\
&= \int_0^{\frac{\pi}{4}} (\ln 2 - \ln(1 + \tan u)) du && \cdot \\
&= \frac{\pi}{4} \ln 2 - \int_0^{\frac{\pi}{4}} \ln(1 + \tan u) du && \cdot \\
&= \frac{\pi}{4} \ln 2 - I && \cdot
\end{aligned}$$

Les nœuds sur la gauche sont à la fin de chaque ligne de texte. Ces nœuds seront appelés *nœuds de gauche*. Les nœuds du côté droit sont alignés verticalement sur le bord droit de l'alignement d'équations. Ces nœuds seront appelés *nœuds de droite*.

Par défaut, les flèches utilisent les nœuds de droite. Nous dirons que ces flèches sont dans le mode `rr` (*r* for *right* en anglais). Ces flèches sont verticales (nous dirons qu'une flèche est *verticale* lorsque ses deux extrémités sont à la même abscisse).

Néanmoins, il est possible d'utiliser les nœuds de gauche, ou une combinaison de nœuds de gauche et de droite avec l'une des options `lr`, `rl` et `ll` (*l* for *left* en anglais). Ces flèches sont, la plupart du temps, non verticales.

6. L'option `show-nodes` peut être utilisée pour visualiser ces nœuds. Les nœuds sont en fait des nœuds Tikz de forme rectangulaire et de largeur nulle. Une flèche entre deux nœuds part de l'ancre sud (*south anchor*) du premier nœud et arrive à l'ancre nord (*north anchor*) du deuxième nœud.

$$\begin{aligned}
\text{Ainsi } I &= \int_{\frac{\pi}{4}}^0 \ln\left(1 + \tan\left(\frac{\pi}{4} - u\right)\right) (-du) \\
&= \int_0^{\frac{\pi}{4}} \ln\left(1 + \tan\left(\frac{\pi}{4} - u\right)\right) du \\
&= \int_0^{\frac{\pi}{4}} \ln\left(1 + \frac{1 - \tan u}{1 + \tan u}\right) du \\
&= \int_0^{\frac{\pi}{4}} \ln\left(\frac{1 + \tan u + 1 - \tan u}{1 + \tan u}\right) du \\
&= \int_0^{\frac{\pi}{4}} \ln\left(\frac{2}{1 + \tan u}\right) du \\
&= \int_0^{\frac{\pi}{4}} (\ln 2 - \ln(1 + \tan u)) du \\
&= \frac{\pi}{4} \ln 2 - \int_0^{\frac{\pi}{4}} \ln(1 + \tan u) du \\
&= \frac{\pi}{4} \ln 2 - I
\end{aligned}$$

Cette flèche utilise une option `lr`.

Cette flèche utilise une option `ll`
et une option `jump` fixée à 2

Il y a aussi une option nommée `i` (*i* pour *intermédiaire*). Avec cette option, la flèche est verticale et calée à gauche.

```

\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) \ \Arrowright[i]{parce que \$(x-y)(x+y)=x^2-y^2\$} \\
& = a^4-b^4
\end{WithArrows}

```

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

} parce que $(x-y)(x+y) = x^2 - y^2$

L'environnement `{WithArrows}` propose aussi une option `group`. Avec cette option, toutes les flèches de l'environnement sont regroupées sur une même verticale calée à gauche.

```

\begin{WithArrows}[displaystyle,group]
2xy'-3y=\sqrt{x}
& \Longleftarrow 2x(K'y_0+Ky'_0)-3Ky_0 = \sqrt{x} \\
& \Longleftarrow 2xK'y_0 + K(2xy'_0-3y_0) = \sqrt{x} \\
& \Longleftarrow 2x K'y_0 = \sqrt{x} \ \Arrowleft{...} \\
...
\end{WithArrows}

```

$$\begin{aligned}
2xy' - 3y = \sqrt{x} &\iff 2x(K'y_0 + Ky'_0) - 3Ky_0 = \sqrt{x} \\
&\iff 2xK'y_0 + K(2xy'_0 - 3y_0) = \sqrt{x} \\
&\iff 2xK'y_0 = \sqrt{x} \\
&\iff 2xK'x^{\frac{3}{2}} = x^{\frac{1}{2}} \\
&\iff K' = \frac{1}{2x^2} \\
&\iff K = -\frac{1}{2x}
\end{aligned}$$

} on remplace y_0 par sa valeur

} simplification par x

} on primitive

L'environnement `{WithArrows}` fournit encore une autre option, nommée `groups` (avec un *s* dans le nom). Avec cette option, les flèches sont divisées en plusieurs « groupes ». Chaque groupe est un

ensemble de flèches connectées entre elles⁷. Toutes les flèches d'un même groupe sont placées sur une même verticale qui est calée à gauche.

$$\begin{aligned}
A &= B \\
&= C + D && \left. \begin{array}{l} \text{un} \\ \text{deux} \end{array} \right\} \\
&= D' \\
&= E + F + G + H + I \\
&= K + L + M && \left. \begin{array}{l} \text{trois} \\ \text{quatre} \end{array} \right\} \\
&= N \\
&= O
\end{aligned}$$

Dans un environnement qui utilise l'option `group` ou l'option `groups`, il est encore possible de donner une option de position (`ll`, `lr`, `rl`, `rr` ou `i`) à une flèche individuelle⁸. Une telle flèche sera tracée indépendamment des groupes. Il est aussi possible de commencer un nouveau groupe avec l'option `new-group` pour une certaine flèche.

Si on le souhaite, on peut passer l'option `group` ou `groups` à la commande `WithArrowsOptions` de telle sorte qu'elle deviendra la valeur par défaut (jusqu'à la fin du groupe TeX courant). Dans ce cas, il est encore possible de revenir au comportement par défaut pour un environnement `{WithArrows}` donné avec l'option `rr` : `\begin{WithArrows}[rr]`.

Dans l'exemple suivant, nous avons utilisé l'option `groups` pour l'environnement et l'option `new-group` pour la dernière flèche (c'est pourquoi cette dernière flèche n'est pas alignée avec les premières).

$$\begin{aligned}
\sum_{k=0}^n \frac{\cos kx}{\cos^k x} &= \sum_{k=0}^n \frac{\Re(e^{ikx})}{(\cos x)^k} && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} (\cos x)^k \text{ est réel} \\
&= \sum_{k=0}^n \Re\left(\frac{e^{ikx}}{(\cos x)^k}\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \Re(z + z') = \Re(z) + \Re(z') \\
&= \Re\left(\sum_{k=0}^n \left(\frac{e^{ix}}{\cos x}\right)^k\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{somme de termes d'une suite géométrique} \\
&= \Re\left(\frac{1 - \left(\frac{e^{ix}}{\cos x}\right)^{n+1}}{1 - \frac{e^{ix}}{\cos x}}\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{calcul algébrique} \\
&= \Re\left(\frac{1 - \frac{e^{i(n+1)x}}{\cos^{n+1} x}}{1 - \frac{e^{ix}}{\cos x}}\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{réduction au même dénominateur} \\
&= \Re\left(\frac{\frac{\cos^{n+1} x - e^{i(n+1)x}}{\cos^{n+1} x}}{\frac{\cos x - e^{ix}}{\cos x}}\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \Re(kz) = k \cdot \Re(z) \text{ lorsque } k \text{ est réel} \\
&= \frac{1}{\cos^n x} \Re\left(\frac{\cos^{n+1} x - e^{i(n+1)x}}{\cos x - e^{ix}}\right) && \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{forme algébrique des nombres complexes} \\
&= \frac{1}{\cos^n x} \Re\left(\frac{\cos^{n+1} x - (\cos(n+1)x + i \sin(n+1)x)}{\cos x - (\cos x + i \sin x)}\right) \\
&= \frac{1}{\cos^n x} \Re\left(\frac{(\cos^{n+1} x - \cos(n+1)x) - i \sin(n+1)x}{-i \sin x}\right) \\
&= \frac{1}{\cos^n x} \cdot \frac{\sin(n+1)x}{\sin x}
\end{aligned}$$

7. Plus précisément : pour chaque flèche a , notons $i(a)$ le numéro de sa rangée de départ et $f(a)$ le numéro de sa rangée d'arrivée ; pour deux flèches a et b , nous noterons $a \sim b$ lorsque $\llbracket i(a), f(a) \rrbracket \cap \llbracket i(b), f(b) \rrbracket \neq \emptyset$; les groupes sont les classes d'équivalence de la clôture transitive de la relation \sim .

8. Une telle flèche est qualifiée d'*indépendante* (*independent* en anglais) dans la documentation technique.

4 L'option « o » pour des flèches individuelles

Considérons, dans un environnement donné, deux flèches notées formellement a et b .

On notera i_a et i_b les numéros des lignes de départ de a et b et f_a et f_b les numéros de leurs lignes d'arrivée. On a bien entendu $i_a \leq f_a$ et $i_b \leq f_b$

On dira que la flèche a recouvre la flèche b lorsque $i_a \leq i_b \leq f_b \leq f_a$

Dans l'exemple ci-contre, la flèche rouge recouvre la flèche bleue.

$$\begin{array}{l} A = B \\ = C \\ = D \\ = E \end{array}$$

Au niveau local, il existe une option `o`. Cette option n'est disponible que lorsque l'on est en mode `group` ou bien en mode `groups` (cf. p. 8).

Une flèche de type `o` est tracée avec un décalage horizontal (comme celui fixé par `xoffset`) calculé automatiquement en fonction des flèches qu'elle recouvre.⁹

```

 $\begin{WithArrows}[groups]
A & = B & \Arrow{un}\Arrow[o,jump=3]{direct} \ \ \
& = C + C & \Arrow{deux} \ \ \
& = D + D + D & \Arrow{trois} \ \ \
& = E + E \ \ \
& = F + F
\end{WithArrows}$ 

```

$$\begin{array}{l} A = B \\ = C + C \\ = D + D + D \\ = E + E \\ = F + F \end{array}$$

Les flèches de type `o` peuvent elles-mêmes être recouvertes par d'autres flèches de type `o` :

```

 $\begin{WithArrows}[groups]
A & = B & \Arrow{un}\Arrow[o,jump=2]{deux}\Arrow[o,jump=3]{trois}\ \ \
& = C \ \ \
& = D \ \ \
& = E + E + E + E + E + E + E
\end{WithArrows}$ 

```

$$\begin{array}{l} A = B \\ = C \\ = D \\ = E + E + E + E + E + E + E \end{array}$$

L'espace (horizontal) entre une flèche de type `o` et les flèches immédiatement recouvertes est fixé par le paramètre `xoffset-for-o-arrows` que l'on peut régler avec `\WithArrowsOptions` (valeur initiale : 2 mm).

Remarque : La lettre `o` a été choisie car c'est la première lettre du mot anglais *over*. Une flèche de type `o` est *au-dessus* des flèches qu'elle recouvre.

⁹. Parmi les flèches recouvertes, les flèches indépendantes (c'est-à-dire celles qui ont une option explicite `rr`, `ll`, `rl`, `lr`, `i`, `up` ou `down`) ne sont pas prises en compte pour le calcul du `xoffset`.

5 Les options « up » et « down » pour des flèches individuelles

Au niveau local, il y a deux options pour les flèches individuelles, nommées `up` et `down`. L'exemple suivant illustre ces types de flèches :

```
\(\begin{WithArrows}
A & = B
\Arrow[up]{une flèche de type \texttt{up}} \\\
& = C + C + C + C + C + C + C + C \\\
& = C + C + C + C + C + C + C + C
\Arrow[down]{une flèche de type \texttt{down}} \\\
& = E + E
\end{WithArrows}\)
```

$$\begin{array}{l}
 A = B \quad \xrightarrow{\text{une flèche de type up}} \\
 = C + C + C + C + C + C + C + C \quad \downarrow \\
 = C + C + C + C + C + C + C + C \\
 = E + E \quad \xleftarrow{\text{une flèche de type down}}
 \end{array}$$

Les options `up` et `down` nécessitent la bibliothèque `Tikz calc`. Si elle n'a pas été chargée au préalable par l'utilisateur, une erreur sera levée.

Les options `up` et `down` peuvent en fait prendre comme valeur une liste de couples clé-valeur.

- La clé `radius` est le rayon de l'angle arrondi de la flèche¹⁰.
- La clé `width` contrôle la largeur (de la partie horizontale) de la flèche :
 - avec la valeur `max`, la largeur de la flèche est ajustée par rapport à la position des nœuds (c'est le comportement par défaut des flèches `up` et `down` comme illustré dans l'exemple précédent) ;
 - avec une valeur numérique, la largeur de la flèche est directement fixée par cette valeur numérique ;
 - avec la valeur `min`, la largeur de la flèche est ajustée au contenu de son étiquette.

```
\$ \begin{WithArrows}
A & = B
\Arrow[up={width=2cm, radius=0pt}]{essai} \\\
& = C + C + C + C + C + C + C + C
\end{WithArrows}\$
```

$$\begin{array}{l}
 A = B \quad \xrightarrow{\text{essai}} \\
 = C + C + C + C + C + C + C + C \quad \downarrow
 \end{array}$$

```
\$ \begin{WithArrows}
A & = B
\Arrow[up={width=min}]{essai} \\\
& = C + C + C + C + C + C + C + C
\end{WithArrows}\$
```

10. La valeur initiale de ce paramètre est égale à 4 pt, qui est la valeur par défaut du “rounded corners” de Tikz.

$$\begin{array}{l}
 A = B \\
 = C + C + C + C + C + C + C + C
 \end{array}
 \overset{\textit{essai}}{\curvearrowright}$$

Les options relatives aux flèches `up` et `down` peuvent être fixées au niveau global ou environnemental avec la clé `up-and-down`. Cette clé peut aussi être utilisée comme préfixe comme illustré maintenant.

```
\WithArrowsOptions{up-and-down/width=min}
```

6 Comparaison avec l’environnement `{aligned}`

L’environnement `{WithArrows}` présente des similitudes avec l’environnement `{aligned}` de l’extension `amsmath`. Ce sont seulement des similitudes car `{WithArrows}` n’a pas été écrit en s’appuyant sur `{aligned}`.¹¹

Comme dans les environnements de l’extension `amsmath`, il est possible de changer l’espacement entre deux rangées avec l’option de la commande `\` de fin de ligne (il est aussi possible d’utiliser `*` mais cela a exactement le même effet que `\` puisqu’un environnement `{WithArrows}` est toujours insécable). Cette option est conçue pour être utilisée avec des valeurs positives uniquement.

```

\begin{WithArrows}
A & = (a+1)^2 \Arrow{on développe} \\[2ex]
& = a^2 + 2a + 1
\end{WithArrows}

```

$$\begin{array}{l}
 A = (a + 1)^2 \\
 = a^2 + 2a + 1
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right) \textit{on développe}$$

Dans les environnements de l’extension `amsmath` (ou de `mathtools`), l’espace entre deux rangées est fixé par un paramètre appelé `\jot` (il s’agit d’une dimension et non d’un ressort). C’est aussi le cas pour l’environnement `{WithArrows}`. Une option `jot` a été ajoutée à l’environnement `{WithArrows}` dans le but de changer ce paramètre `\jot` pour un environnement donné.¹²

```

\begin{WithArrows}[displaystyle,jot=2ex]
F & = \frac{1}{2}G \Arrow{on développe}\
& = H + \frac{1}{2}K \Arrow{on continue}\
& = K
\end{WithArrows}

```

$$\begin{array}{l}
 F = \frac{1}{2}G \\
 = H + \frac{1}{2}K \\
 = K
 \end{array}
 \left. \begin{array}{l} \\ \\ \end{array} \right) \textit{on développe}$$

$$\left. \begin{array}{l} \\ \\ \end{array} \right) \textit{on continue}$$

Néanmoins, cette nouvelle valeur de `\jot` sera aussi utilisée dans les nouveaux alignements inclus dans l’environnement `{WithArrows}` :

11. En fait, il est possible d’utiliser l’extension `witharrows` sans l’extension `amsmath`.

12. Il est aussi possible de changer `\jot` avec l’environnement `{spreadlines}` de `mathtools`.

```

 $\begin{WithArrows}[jot=2ex]
\varphi(x,y) = 0 \quad \& \quad \Leftrightarrow (x+y)^2 + (x+2y)^2 = 0
\Arrow{\$x\$ et \$y\$ sont réels}\
& \Leftrightarrow \left\{
\begin{aligned}
x+y &= 0 \\
x+2y &= 0
\end{aligned}
\right.
\right.
\end{WithArrows}$ 

```

$$\varphi(x,y) = 0 \Leftrightarrow (x+y)^2 + (x+2y)^2 = 0 \quad \left. \vphantom{\varphi(x,y)} \right\} x \text{ et } y \text{ sont réels}$$

$$\Leftrightarrow \begin{cases} x+y = 0 \\ x+2y = 0 \end{cases}$$

Peut-être que cela ne correspond pas au résultat souhaité. C'est pourquoi une option `interline` est fournie. Il est possible d'utiliser un ressort (*skip*) pour cette option.

```

 $\begin{WithArrows}[interline=2ex]
\varphi(x,y) = 0 \quad \& \quad \Leftrightarrow (x+y)^2 + (x+2y)^2 = 0
\Arrow{\$x\$ et \$y\$ sont réels}\
& \Leftrightarrow \left\{
\begin{aligned}
x+y &= 0 \\
x+2y &= 0
\end{aligned}
\right.
\right.
\end{WithArrows}$ 

```

$$\varphi(x,y) = 0 \Leftrightarrow (x+y)^2 + (x+2y)^2 = 0 \quad \left. \vphantom{\varphi(x,y)} \right\} x \text{ et } y \text{ sont réels}$$

$$\Leftrightarrow \begin{cases} x+y = 0 \\ x+2y = 0 \end{cases}$$

De même que l'environnement `aligned`, `WithArrows` a une option de placement qui peut prendre les valeurs `t`, `c` ou `b`. Néanmoins, la valeur initiale n'est pas `c` mais `t`. Si on le souhaite, il est possible d'avoir la valeur `c` comme valeur par défaut en utilisant l'instruction `\WithArrowsOptions{c}` au début du document.

```

Ainsi\enskip
 $\begin{WithArrows}
A \& = (a+1)^2 \Arrow{\text{on développe}} \
& = a^2 + 2a + 1
\end{WithArrows}$ 

```

$$\text{Ainsi } A = (a+1)^2 \quad \left. \vphantom{A} \right\} \text{on développe}$$

$$= a^2 + 2a + 1$$

La valeur `c` peut être utile, par exemple, si on souhaite mettre une accolade :

```

On pose\enskip  $\left\{
\begin{aligned}
f(x) &= 3x^3+2x^2-x+4 \\
g(x) &= 5x^2-5x+6
\end{aligned}
\right.$ 

```

On pose $\left\{ \begin{array}{l} f(x) = 3x^3 + 2x^2 - x + 4 \\ g(x) = 5x^2 - 5x + 6 \end{array} \right\}$ les deux sont des polynômes

Contrairement à `{aligned}`, l'environnement `{WithArrows}` utilise `\textstyle` par défaut. Là aussi, il est possible de changer ce comportement avec `\WithArrowsOptions` :

`\WithArrowsOptions{displaystyle}`.

L'exemple suivant est composé avec `{aligned}` :

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

L'exemple suivant est composé avec `{WithArrows}[c,displaystyle]`. Les résultats sont parfaitement identiques.

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

7 Les flèches dans les environnements imbriqués

Les environnements `{WithArrows}` peuvent être imbriqués. Dans ce cas, les options fournies à l'environnement englobant s'appliquent aussi aux environnements imbriqués (avec des exceptions tout à fait attendues pour `interline`, `code-before` et `code-after`). La commande `\Arrow` peut être utilisée dans chaque environnement `{WithArrows}`.

```

 $\begin{WithArrows}
\varphi(x,y)=0
& \Leftrightarrow (x+2y)^2+(2x+4y)^2 = 0 \Arrow{les nombres sont réels}
& \Leftrightarrow
\left\{ \begin{array}{l} x+2y = 0 \\ 2x+4y = 0 \end{array} \right.
\end{WithArrows}
& \Leftrightarrow
\left\{ \begin{array}{l} x+2y = 0 \\ x+2y = 0 \end{array} \right. \Arrow{la même équation}
& \Leftrightarrow x+2y=0
\end{WithArrows}$ 

```

$$\begin{aligned} \varphi(x,y) = 0 &\Leftrightarrow (x+2y)^2 + (2x+4y)^2 = 0 \\ &\Leftrightarrow \left\{ \begin{array}{l} x+2y = 0 \\ 2x+4y = 0 \end{array} \right. \left. \vphantom{\varphi(x,y) = 0} \right\} \text{les nombres sont réels} \\ &\Leftrightarrow \left\{ \begin{array}{l} x+2y = 0 \\ x+2y = 0 \end{array} \right. \left. \vphantom{\varphi(x,y) = 0} \right\} \text{la même équation} \\ &\Leftrightarrow x+2y = 0 \end{aligned}$$

Néanmoins, on peut souhaiter tracer une flèche entre des rangées qui ne sont pas dans le même environnement. Par exemple, on pourrait souhaiter tracer la flèche suivante :

$$\begin{aligned} \varphi(x, y) = 0 &\Leftrightarrow (x + 2y)^2 + (2x + 4y)^2 = 0 \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ x + 2y = 0 \end{cases} \quad \left. \vphantom{\begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases}} \right) \textit{division par 2} \\ &\Leftrightarrow x + 2y = 0 \end{aligned}$$

Une telle construction est possible en utilisant `\Arrow` dans le `code-after` : dans le `code-after`, une version spéciale de la commande `\Arrow` est disponible (nous l'appellerons « `\Arrow` dans le `code-after` »).

Une commande `\Arrow` dans le `code-after` prend trois arguments :

- une spécification pour la rangée de départ de la flèche ;
- une spécification pour la rangée d'arrivée ;
- une étiquette pour la flèche.

Comme d'habitude, il est possible de donner des options entre crochets avant ou après les trois arguments obligatoires. Néanmoins, ces options sont limitées (voir plus bas).

La spécification de rangée est construite avec la position de l'environnement considéré dans l'arbre des imbrications, suivie, après un trait d'union, par le numéro de la rangée.

Dans l'exemple précédent, il y a deux environnements `{WithArrows}` imbriqués dans l'environnement `{WithArrows}` principal.

$$\begin{aligned} \varphi(x, y) = 0 &\Leftrightarrow (x + 2y)^2 + (2x + 4y)^2 = 0 \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases} \quad \textit{environnement n° 1} \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ x + 2y = 0 \end{cases} \quad \textit{environnement n° 2} \\ &\Leftrightarrow x + 2y = 0 \end{aligned}$$

La flèche que nous voulons tracer part de la rangée 2 du sous-environnement n° 1 (et, par conséquent, la spécification est 1-2) et arrive à la rangée 2 du sous-environnement n° 2 (et, par conséquent, la spécification est 2-2). Nous pouvons tracer la flèche avec une commande `\Arrow` dans le `code-after` de la manière suivante :

```
\begin{WithArrows}[code-after = \Arrow{1-2}{2-2}{division par $2$}]
\varphi(x,y)=0
& \Leftrightarrow (x+2y)^2+(2x+4y)^2 = 0 \\\
.....
\end{WithArrows}
```

$$\begin{aligned} \varphi(x, y) = 0 &\Leftrightarrow (x + 2y)^2 + (2x + 4y)^2 = 0 \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} x + 2y = 0 \\ x + 2y = 0 \end{cases} \quad \left. \vphantom{\begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases}} \right) \textit{division par 2} \\ &\Leftrightarrow x + 2y = 0 \end{aligned}$$

Les options autorisées pour une commande `\Arrow` dans le `code-after` sont : `ll`, `lr`, `rl`, `rr`, `v`, `xoffset`, `tikz` et `tikz-code`. Exceptée `v`, qui est spécifique à `\Arrow` dans le `code-after`, toutes ces options ont leur signification habituelle.

Avec l'option `v`, la flèche tracée est verticale à une abscisse calculée avec la rangée de départ et la rangée d'arrivée uniquement : les lignes intermédiaires ne sont pas prises en compte contrairement à ce qui se

pas avec l'option `i`. Pour le moment, l'option `i` n'est pas disponible pour la commande `\Arrow` dans le `code-after`. Néanmoins, il est toujours possible de déplacer une flèche avec `xoffset` (ou `xshift` de Tikz).

```

\begin{WithArrows}[code-after=\Arrow[v]{1-2}{2-2}{division by $2$}]
\varphi(x,y)=0
& \Leftrightarrow (x+2y)^2+(2x+4y)^2 = 0 \\\
.....
\end{WithArrows}$

```

$$\begin{aligned}
\varphi(x,y) = 0 &\Leftrightarrow (x + 2y)^2 + (2x + 4y)^2 = 0 \\
&\Leftrightarrow \begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases} \\
&\Leftrightarrow \begin{cases} x + 2y = 0 \\ x + 2y = 0 \end{cases} \quad \left. \vphantom{\begin{cases} x + 2y = 0 \\ x + 2y = 0 \end{cases}} \right) \textit{division par 2} \\
&\Leftrightarrow x + 2y = 0
\end{aligned}$$

L'extension `witharrows` fournit une autre commande disponible dans le `code-after` : la commande `\MultiArrow`. Cette commande dessine un « râteau ». La liste des rangées concernées par ce râteau est donnée dans le premier argument de la commande `\MultiArrow`. La syntaxe utilisée pour décrire cette liste est celle de la commande `\foreach` de `pgffor`.

```

\begin{WithArrows}[tikz = rounded corners,
code-after = {\MultiArrow{1,...,4}{text}} ]
A & = B \\\
& = C \\\
& = D \\\
& = E \\\
& = F
\end{WithArrows}$

```

$$\begin{aligned}
A &= B \leftarrow \\
&= C \leftarrow \\
&= D \leftarrow \\
&= E \leftarrow \\
&= F
\end{aligned}
\quad \left. \vphantom{\begin{aligned} A &= B \\ &= C \\ &= D \\ &= E \end{aligned}} \right) \textit{text}$$

Pour le moment, aucune option n'est disponible pour cette commande `\MultiArrow`.

8 Dessiner des flèches depuis l'extérieur des environnements `{WithArrows}`

Si on désire dessiner des flèches depuis l'extérieur des environnements `{WithArrows}`, il est possible de le faire en utilisant les nœuds Tikz créés par ces environnements.

Le nom d'un nœud Tikz créé par `witharrows` est préfixé par `wa-`. Ensuite, on trouve une liste de nombres qui indique quelle est la position de l'environnement dans l'arbre d'imbrication des environnements. À la fin, on a le suffixe `l` pour un nœud de gauche et le suffixe `r` pour un nœud de droite.

À titre d'illustration, nous donnons un exemple d'environnements `{WithArrows}` imbriqués et, pour chaque nœud de droite, le nom de ce nœud.¹³

13. Il existe une option `show-node-names` pour faire afficher les noms de ces nœuds.

$$\begin{aligned}
& A \triangleleft B + B + B + B + B + B + B + B + B + B + B + B + B + B + B + B + B \\
& \triangleleft \left\{ \begin{array}{l} C \triangleleft D \\ E \triangleleft F \end{array} \right. \\
& \triangleleft \left\{ \begin{array}{l} G \triangleleft H + H + H + H + H + H + H \\ I \triangleleft \left\{ \begin{array}{l} J \triangleleft K \\ L \triangleleft M \end{array} \right. \end{array} \right. \\
& \triangleleft \left\{ \begin{array}{l} N \triangleleft O \\ P \triangleleft Q \end{array} \right.
\end{aligned}$$

L'extension `witharrows` fournit quelques outils pour faciliter l'utilisation de ces nœuds :

- la commande `\WithArrowsLastEnv` donne le numéro du dernier environnement de niveau 0 (c'est-à-dire non imbriqué dans un autre environnement de `witharrows`) ;
- un nom peut être donné à un environnement avec l'option `name` et, dans ce cas, les nœuds créés dans l'environnement auront des alias construits en utilisant ce nom ;
- le style Tikz `WithArrows/arrow` est le style utilisé par `witharrows` pour dessiner une flèche¹⁴ ;
- le style Tikz `WithArrows/arrow/tips` est le style utilisé pour la pointe de flèche (ce style est lui-même chargé par le style `WithArrows/arrow`).

Par exemple, nous pouvons tracer une flèche allant de `wa-45-2-1-2-r.south` à `wa-45-3-2-r.north` avec l'instruction Tikz suivante :

```

\begin{tikzpicture}[remember picture,overlay]
\draw [WithArrows/arrow]
  ([xshift=3mm]wa-\WithArrowsLastEnv-2-1-2-r.south)
  to ([xshift=3mm]wa-\WithArrowsLastEnv-3-2-r.north) ;
\end{tikzpicture}

```

$$\begin{aligned}
& A \triangleleft B + B + B + B + B + B + B + B + B + B + B + B + B + B + B + B + B \\
& \triangleleft \left\{ \begin{array}{l} C \triangleleft D \\ E \triangleleft F \end{array} \right. \\
& \triangleleft \left\{ \begin{array}{l} G \triangleleft H + H + H + H + H + H + H \\ I \triangleleft \left\{ \begin{array}{l} J \triangleleft K \\ L \triangleleft M \end{array} \right. \end{array} \right. \\
& \triangleleft \left\{ \begin{array}{l} N \triangleleft O \\ P \triangleleft Q \end{array} \right.
\end{aligned}$$

Dans le cas présent, il aurait été plus facile d'utiliser une commande `\Arrow` dans le `code-after` mais c'était seulement un exemple pour illustrer comment les nœuds Tikz créés par `witharrows` peuvent être utilisés.

Dans l'exemple suivant, on crée des environnements `{WithArrows}` nommés « `premier` » et « `second` » et on dessine une flèche entre un nœud du premier et un nœud du second.

```

$\begin{WithArrows}[name=premier]
A & = B \\
& = C
\end{WithArrows}$

```

14. Plus précisément, ce style est passé à Tikz via l'option « `every path` » avant le dessin de la flèche (qui se fait avec l'instruction Tikz stockée dans `tikz-code`). Ce style est modifié par l'option `tikz` de `witharrows` (avec une portée correspondant au groupe TeX courant).

```

\bigskip
$\begin{WithArrows}[name=second]
A' & = B' \\
& = C'
\end{WithArrows}$

\begin{tikzpicture}[remember picture,overlay]
\draw [WithArrows/arrow]
    ([xshift=3mm]premier-1-r.south)
    to ([xshift=3mm]second-1-r.north) ;
\end{tikzpicture}

```

$$\begin{array}{l}
 A = B \\
 = C \\
 A' = B' \\
 = C'
 \end{array}
 \begin{array}{l}
 \curvearrowright \\
 \downarrow
 \end{array}$$

9 L'environnement `{DispWithArrows}`

Comme dit précédemment, l'environnement `{WithArrows}` présente des similitudes avec l'environnement `{aligned}` de l'extension `amsmath` (et de `mathtools`). L'extension `witharrows` fournit aussi un environnement `{DispWithArrows}` qui est similaire aux environnements `{align}` et `{flalign}` de l'extension `amsmath`.

L'environnement `{DispWithArrows}` doit être utilisé *hors* du mode mathématique. Comme `{align}`, il doit être utilisé en mode horizontal.

```

\begin{DispWithArrows}
A & = (a+1)^2 \Arrow{on développe} \\
& = a^2 + 2a + 1
\end{DispWithArrows}

```

$$\begin{array}{l}
 A = (a + 1)^2 \\
 = a^2 + 2a + 1
 \end{array}
 \begin{array}{l}
 \curvearrowright \textit{on développe} \\
 \downarrow
 \end{array}
 \begin{array}{l}
 (1) \\
 (2)
 \end{array}$$

Il est possible d'utiliser la commande `\notag` (ou `\nonumber`) pour supprimer un numéro d'équation. Il est possible d'utiliser la commande `\tag` pour placer une étiquette spéciale (par ex. \star). Il est aussi possible de placer un label pour une ligne de l'environnement avec la commande `\label`. Ces commandes doivent être utilisées dans la dernière colonne de l'environnement.

```

\begin{DispWithArrows}
A & = (a+1)^2 \Arrow{on développe} \notag \\
& = a^2 + 2a + 1 \tag{\star} \label{mon-équation}
\end{DispWithArrows}

```

$$\begin{array}{l}
 A = (a + 1)^2 \\
 = a^2 + 2a + 1
 \end{array}
 \begin{array}{l}
 \curvearrowright \textit{on développe} \\
 \downarrow
 \end{array}
 \begin{array}{l}
 \\
 (\star)
 \end{array}$$

Un lien vers l'équation (\star) .¹⁵

15. Dans ce document, les références ont été personnalisées avec `\labelformat{equation}{(#1)}`

Si `amsmath` (ou `mathtools`) est chargée, il est possible d'utiliser `\tag*` qui compose le label sans les parenthèses. Par exemple, il est possible de l'utiliser pour mettre le symbole `\square` de `amssymb`. Ce symbole est souvent utilisé pour marquer la fin d'une démonstration.¹⁶

```
\begin{DispWithArrows}
A & = (a+1)^2 \Arrow{on développe} \notag \\
& = a^2 + 2a + 1 \tag*{\square}
\end{DispWithArrows}
```

$$\begin{aligned}
 A &= (a+1)^2 \\
 &= a^2 + 2a + 1 \quad \left. \vphantom{A} \right\} \textit{on développe}
 \end{aligned}
 \quad \square$$

Il est également possible de supprimer tous les numéros d'équations avec l'option booléenne `notag` (ou `nonumber`), au niveau global ou bien au niveau d'un environnement.

Enfin, il existe aussi un environnement `{DispWithArrows*}` qui supprime tous les numéros.¹⁷

```
\begin{DispWithArrows*}
A & = (a+1)^2 \Arrow{on développe} \\
& = a^2 + 2a + 1
\end{DispWithArrows*}
```

$$\begin{aligned}
 A &= (a+1)^2 \\
 &= a^2 + 2a + 1 \quad \left. \vphantom{A} \right\} \textit{on développe}
 \end{aligned}$$

En fait, il y a une autre option, nommée `tagged-lines`, qui peut être utilisée pour contrôler quelles lignes seront numérotées. La valeur de cette option est la liste des numéros de lignes (dans l'environnement) qui seront numérotées (par le système de numérotage des équations). Par exemple, avec l'option `tagged-lines = {first,3,last}`, seulement la première, la troisième et la dernière lignes seront numérotées. Il y a aussi une valeur spéciale `all` qui signifie que toutes les lignes doivent être numérotées.

```
\begin{DispWithArrows}[tagged-lines = last]
A & = A_1 \Arrow{première étape} \\
& = A_2 \Arrow{deuxième étape} \\
& = A_3
\end{DispWithArrows}
```

$$\begin{aligned}
 A &= A_1 \\
 &= A_2 \\
 &= A_3
 \end{aligned}
 \left. \vphantom{A} \right\} \begin{array}{l} \textit{première étape} \\ \textit{deuxième étape} \end{array}
 \tag{3}$$

Avec l'option `fleqn`, l'environnement est composé calé à gauche (d'une manière similaire à l'option `fleqn` des classes standard de LaTeX). Dans ce cas, la marge gauche peut être réglée avec l'option `mathindent` (qui a un nom inspiré du paramètre `\mathindent` de LaTeX). La valeur initiale pour cette option est de 25 pt. On peut y mettre comme valeur un ressort (= `skip = glue`).

```
\begin{DispWithArrows}[fleqn,mathindent = 1cm]
A & = (a+1)^2 \Arrow{on développe} \\
& = a^2 + 2a + 1
\end{DispWithArrows}
```

16. Signalons que l'environnement `{DispWithArrows}` est compatible avec la commande `\qedhere` de `amsthm`.

17. Même dans ce cas, il est possible de mettre une étiquette manuellement avec la commande `\tag`.

$$A = (a + 1)^2 \tag{4}$$

$$= a^2 + 2a + 1 \tag{5}$$

on développe

Remarque : Par conception, l’option `fleqn` de `witharrows` est indépendante de l’option de classe `fleqn` de LaTeX. En effet, puisque les environnements de `witharrows` sont conçus pour être utilisés avec des flèches sur le côté droit, l’utilisateur peut souhaiter utiliser `witharrows` avec l’option `fleqn` (de manière à avoir plus de place sur la droite des équations pour les flèches) tout en continuant à centrer les équations classiques.

Si l’option `leqno` est utilisée comme option de classe, les labels seront composés à gauche également pour les environnements `{DispWithArrows}`.¹⁸

Si l’extension `amsmath` est chargée, il est possible d’utiliser la commande `\intertext` dans les environnements `{DispWithArrows}`. Il est également possible d’utiliser l’environnement `{subequations}`. Néanmoins, il existe pour les environnements `{DispWithArrows}` une option `subequations` qui demande d’encapsuler l’environnement dans un environnement `{subequations}`.

Dans l’exemple suivant, l’option `subequations` est fixée via la commande `\WithArrowsOptions`. Chacun des environnements suivants sera alors sous-numéroté (dans la portée de la commande `WithArrowsOptions`).

`\WithArrowsOptions{subequations}`

Premier environnement.

```
\begin{DispWithArrows}
A & = B \\
& = C
\end{DispWithArrows}
```

Deuxième environnement.

```
\begin{DispWithArrows}
D & = E \\
& = F
\end{DispWithArrows}
```

Premier environnement.

$$A = B \tag{6a}$$

$$= C \tag{6b}$$

Deuxième environnement.

$$D = E \tag{7a}$$

$$= F \tag{7b}$$

S’il n’y a pas suffisamment de place pour faire apparaître le numéro d’équation à la fin de la ligne, il n’y a pas de positionnement automatique de ce numéro sur la ligne suivante (comme dans les environnements de l’extension `amsmath`) : il y aura surimpression de texte. Néanmoins, dans `{DispWithArrows}`, l’utilisateur peut utiliser la commande `\tagnextline` pour demander manuellement la composition du numéro sur la ligne suivante.

```
\begin{DispWithArrows}[displaystyle]
S_{2(p+1)}
& = \sum_{k=1}^{2(p+1)} (-1)^k k^2 \\
& \smash[b]{= \sum_{k=1}^{2p} (-1)^k k^2} \\
& \quad + (-1)^{2p+1} (2p+1)^2 + (-1)^{2p+2} (2p+2)^2 \tagnextline \\
\end{DispWithArrows}
```

¹⁸ L’extension `amsmath` a une option `leqno` mais `witharrows`, bien entendu, ne tient pas compte de cette option : `witharrows` vérifie seulement l’option `leqno` de la classe du document.

```

&= S_{2p}-(2p+1)^2+(2p+2)^2\\
&=p(2p+1)-(2p+1)^2+(2p+2)^2\\
&= 2p^2+5p+3
\end{DispWithArrows}

```

$$\begin{aligned}
S_{2(p+1)} &= \sum_{k=1}^{2(p+1)} (-1)^k k^2 & (8) \\
&= \sum_{k=1}^{2p} (-1)^k k^2 + (-1)^{2p+1} (2p+1)^2 + (-1)^{2p+2} (2p+2)^2 & (9) \\
&= S_{2p} - (2p+1)^2 + (2p+2)^2 & (10) \\
&= 2p^2 + p - 4p^2 - 4p - 1 + 4p^2 + 8p + 4 & (11) \\
&= 2p^2 + 5p + 3 & (12)
\end{aligned}$$

Les environnements `{DispWithArrows}` et `{DispWithArrows*}` fournissent également une option `wrap-lines`. Avec cette option, les lignes de l'étiquette sont automatiquement coupées à droite.

```

\begin{DispWithArrows*}[displaystyle,wrap-lines]
S_n
& = \frac{1}{n} \operatorname{Re} \left( \sum_{k=0}^{n-1} \operatorname{bigl}(e^{i\frac{\pi}{2n}}\operatorname{bigr})^k \right)
\Arrow{somme des termes d'une suite géométrique de raison $e^{i\frac{2\pi}{n}}$}
& = \frac{1}{n} \operatorname{Re} \left( \frac{1-\operatorname{bigl}(e^{i\frac{\pi}{2n}}\operatorname{bigr})^n}{1-e^{i\frac{\pi}{2n}}} \right)
\Arrow{cette ligne de texte a été coupée automatiquement}
& = \frac{1}{n} \operatorname{Re} \left( \frac{1-i}{1-e^{i\frac{\pi}{2n}}} \right)
\end{DispWithArrows*}

```

$$\begin{aligned}
S_n &= \frac{1}{n} \Re \left(\sum_{k=0}^{n-1} \left(e^{i\frac{\pi}{2n}} \right)^k \right) & \left. \begin{array}{l} \text{somme des termes d'une suite} \\ \text{géométrique de raison } e^{i\frac{2\pi}{n}} \end{array} \right\} \\
&= \frac{1}{n} \Re \left(\frac{1 - \left(e^{i\frac{\pi}{2n}} \right)^n}{1 - e^{i\frac{\pi}{2n}}} \right) & \left. \begin{array}{l} \text{cette ligne de texte a été coupée} \\ \text{automatiquement} \end{array} \right\} \\
&= \frac{1}{n} \Re \left(\frac{1 - i}{1 - e^{i\frac{\pi}{2n}}} \right)
\end{aligned}$$

L'option `wrap-lines` ne s'applique pas aux environnements `{WithArrows}` imbriqués dans un environnement `{DispWithArrows}` ou `{DispWithArrows*}`. Néanmoins, elle s'applique aux instructions `\Arrow` et `\MultiArrow` du `code-after` de l'environnement englobant.

Nous avons dit que les environnements `{DispWithArrows}` et `{DispWithArrows*}` devaient être utilisés en mode horizontal et non en mode vertical. Il y a néanmoins une exception. Ces environnements peuvent être utilisés directement après un `\item` d'une liste LaTeX. Dans ce cas, aucun espace vertical n'est ajouté avant l'environnement.¹⁹

Voici un exemple où, avec `{DispWithArrows}`, on peut insérer l'équation dans un environnement `{enumerate}` tout en utilisant l'option `wrap-lines`.

19. Il est possible de désactiver cette fonctionnalité avec l'option `standard-behaviour-with-items`.

```

\begin{enumerate}
\item
\begin{DispWithArrows}%
[displaystyle, wrap-lines, tagged-lines = last, fleqn, mathindent = 0 pt]
S_n
& = \frac{1}{n} \Re \left( \sum_{k=0}^{n-1} \bigl( e^{i \frac{\pi}{2n}} \bigr)^k \right)
\Arrow{on utilise la formule pour la somme de termes d'une suite géométrique de raison
$e^{i \frac{2\pi}{n}}$} \\
& = \frac{1}{n} \Re \left( \frac{1 - \bigl( e^{i \frac{\pi}{2n}} \bigr)^n}{1 - e^{i \frac{\pi}{2n}}} \right)
\Arrow{$\bigl( e^{i \frac{\pi}{2n}} \bigr)^n = e^{i \frac{\pi}{2}} = i$} \\
& = \frac{1}{n} \Re \left( \frac{1 - i}{1 - e^{i \frac{\pi}{2n}}} \right)
\end{DispWithArrows}
\end{enumerate}

```

$$\begin{aligned}
1. S_n &= \frac{1}{n} \Re \left(\sum_{k=0}^{n-1} \left(e^{i \frac{\pi}{2n}} \right)^k \right) \\
&= \frac{1}{n} \Re \left(\frac{1 - \left(e^{i \frac{\pi}{2n}} \right)^n}{1 - e^{i \frac{\pi}{2n}}} \right) \\
&= \frac{1}{n} \Re \left(\frac{1 - i}{1 - e^{i \frac{\pi}{2n}}} \right)
\end{aligned}
\begin{array}{l}
\left. \begin{array}{l}
\text{on utilise la formule pour la somme de termes d'une suite} \\
\text{géométrique de raison } e^{i \frac{2\pi}{n}}
\end{array} \right\} \\
\left. \begin{array}{l}
\left(e^{i \frac{\pi}{2n}} \right)^n = e^{i \frac{\pi}{2}} = i
\end{array} \right\}
\end{array}
\tag{13}$$

L'environnement `{DispWithArrows}` est similaire à l'environnement `{align}` de l'extension `amsmath`. Néanmoins, `{DispWithArrows}` n'est pas construit en utilisant `{align}` (en fait, il est possible d'utiliser `witharrows` sans `amsmath`).

Il y a quelques différences entre `{DispWithArrows}` et `{align}`.

- L'environnement `{DispWithArrows}` ne peut pas être inséré dans un environnement `{gather}` de l'extension `amsmath`.
- Un environnement `{DispWithArrows}` est toujours insécable (même avec `\allowdisplaybreaks` de l'extension `amsmath`).
- Les commandes `\label`, `\tag`, `\notag` et `\nonumber` ne sont autorisées que dans la dernière colonne.
- Après un `\item` d'une liste LaTeX, aucun espace vertical n'est ajouté (cela peut être modifié avec l'option `standard-behaviour-with-items`).
- **Enfin, par défaut, les éléments d'un environnement `\{DispWithArrows\}` sont composés en `textstyle` et non en `displaystyle` (il est possible de modifier ce point avec l'option `displaystyle`).**

En ce qui concerne les références, l'extension `witharrows` est compatible avec les extensions `autonom`, `cleveref`, `fancyref`, `hyperref`, `listbls`, `prettyref`, `refcheck`, `refstyle`, `showlabels`, `smartref`, `typedref` et `varioref` et avec les options `showonlyrefs` et `showmanualtags` de `mathtools`.²⁰

Elle n'est pas compatible avec `showkeys` (certains labels ne sont pas affichés).

L'option `<...>` de `DispWithArrows`

L'environnement `{DispWithArrows}` propose une option `left-brace`. Si cette option est présente, son contenu est composé à gauche, suivi d'une accolade (d'où le nom) puis du corps de l'environnement.²¹

Pour la lisibilité, cette option `left-brace` est aussi disponible avec une syntaxe spéciale : on peut la rentrer en la mettant entre chevrons (`<` et `>`) juste après le `{DispWithArrows}` (et donc avant les éventuels arguments optionnels entre crochets).

²⁰ Rappelons que `varioref`, `hyperref`, `cleveref` et `autonom` doivent être chargés dans cet ordre. L'extension `witharrows` peut être chargée n'importe où.

²¹ L'option `left-brace` peut aussi être utilisée sans valeur : dans ce cas, seule l'accolade est tracée.

On peut alors faire des constructions avec distinctions de cas, comme dans l'exemple suivant.²²

```
\begin{DispWithArrows}< \binom{n}{p} = >[format = ll,fleqn,displaystyle]
0 & \quad \text{si } p > n
\Arrow{ce cas peut en fait rentrer\\ dans le suivant} \\
\frac{n(n-1)\cdots(n-p+1)}{p!} & \quad \text{si } 0 \leq p \leq n \\
0 & \quad \text{si } p < 0
\end{DispWithArrows}
```

$$\binom{n}{p} = \begin{cases} 0 & \text{si } p > n \\ \frac{n(n-1)\cdots(n-p+1)}{p!} & \text{si } 0 \leq p \leq n \\ 0 & \text{si } p < 0 \end{cases} \left. \begin{array}{l} \text{ce cas peut en fait rentrer} \\ \text{dans le suivant} \end{array} \right) \quad (14)$$

$$(15)$$

$$(16)$$

Dans l'exemple suivant, on sous-numérote les équations d'un système avec l'option `subequations` (disponible quand l'`amsmath` est chargé).

```
\begin{DispWithArrows}< \label{système} \ref*{système} \Leftrightarrow >%
[format = l, subequations]
x+y+z = -3 \Arrow{tikz=-,jump=2}{3 equations} \\
xy+xz+yz=-2 \\
xyz = -15 \label{dernière-équation}
\end{DispWithArrows}
```

$$(17) \Leftrightarrow \left\{ \begin{array}{l} x + y + z = -3 \\ xy + xz + yz = -2 \\ xyz = -15 \end{array} \right. \left. \begin{array}{l} (17a) \\ (17b) \\ (17c) \end{array} \right) 3 \text{ équations}$$

Le système complet est l'équation (17) (cette référence a été obtenue avec `\ref{système}`) et la dernière équation est l'équation (17c) (référence obtenue avec `\ref{dernière-équation}`). Notons que `\ref*`, utilisé dans le code ci-dessus, est une variante de `ref` qui compose une référence sans créer de lien hypertexte (même quand `hyperref` est chargé).

On peut souhaiter remplacer l'accolade gauche par un autre délimiteur extensible (gauche). On peut le faire avec l'option `replace-left-brace-by`. Par exemple, « `replace-left-brace-by = [\enskip]` » va composer avec un crochet et ajouter aussi un espace de 0.5 em après le crochet.

10 Fonctionnalités avancées

10.1 Utilisation avec Beamer

Nouveau 2.9

Si l'extension `witharrows` est utilisée avec Beamer, la commande `\Arrow` accepte un argument entre chevrons (après l'argument optionnel entre crochets) pour indiquer les « *overlays* » concernés (en interne, `witharrows` utilise simplement la commande `\only` de Beamer).

```
\Arrow[jump=2]<3->{Exemple}
```

²² L'environnement `{cases}` de l'extension `amsmath` permet de réaliser de telles distinctions de cas mais on ne peut pas utiliser la numérotation automatique des équations. L'environnement `{numcases}` de l'extension `cases` (de Donald Arseneau) permet d'utiliser ces numéros d'équations, mais, bien entendu, il n'y a pas de composition de flèches.

10.2 Utilisation avec plain-TeX

L'extension `witharrows` peut être utilisée avec plain-TeX. Pour cela, on doit charger l'extension avec `\input` :

```
\input{witharrows.tex}
```

En plain-TeX, il n'y a pas d'environnements comme dans LaTeX.

Au lieu de `\begin{WithArrows}` et `\end{WithArrows}` comme en LaTeX, on doit utiliser un pseudo-environnement délimité par `\Witharrows` et `\endWithArrows` (de même pour `{DispWithArrows}`).

```
$$\WithArrows
A & = (a+1)^2 \Arrow{on développe} \\
& = a^2 + 2a + 1
\endWithArrows$
```

La version pour plain-TeX de `witharrows` ne propose pas toutes les fonctionnalités proposées pour la version LaTeX. En particulier, les fonctionnalités concernant les numéros d'équations ne sont pas disponibles (puisqu'elles reposent sur le système de numérotation de LaTeX).

10.3 L'option `tikz-code` : comment changer la forme des flèches

L'option `tikz-code` permet à l'utilisateur de changer la forme des flèches.²³

Par exemple, les options « `up` » et « `down` » décrites précédemment (cf. p. 11) sont programmées en interne avec `tikz-code`.

La valeur de cette option doit être une instruction de tracé Tikz valide (avec le point-virgule final) où les trois marqueurs `#1`, `#2` et `#3` représentent le point de départ, le point d'arrivée et l'étiquette de la flèche.

La valeur initiale de `tikz-code` est la suivante :

```
\draw (#1) to node {#3} (#2) ;
```

Dans l'exemple suivant, nous remplaçons le chemin par défaut par un chemin avec trois segments (et un nœud recouvrant le deuxième segment).

```
\begin{WithArrows}[format=c,ygap=5pt,interline=4mm,
  tikz-code = {\draw[rounded corners]
    (#1) -- ([xshift=5mm]#1)
    -- node[circle,
      draw,
      auto = false,
      fill = gray!50,
      inner sep = 1pt] {\tiny #3}
    ([xshift=5mm]#2)
    -- (#2) ; }]}
3 (2x+4) = 6 \Arrow{$\div 3$} \\
2x+4 = 2 \Arrow{$-4$} \\
2x = -2 \Arrow{$\div 2$} \\
x = -1
\end{WithArrows}
```

²³ Si l'option `wrap-lines` est utilisée dans un environnement `{DispWithArrows}` ou `{DispWithArrows*}`, l'option `tikz-code` n'aura pas d'effet sur les flèches de cet environnement mais uniquement sur les flèches des environnements `{WithArrows}` imbriqués.

$$\begin{array}{l}
3(2x + 4) = 6 \\
2x + 4 = 2 \\
2x = -2 \\
x = -1
\end{array}
\begin{array}{l}
\div 3 \\
-4 \\
\div 2
\end{array}$$

L'environnement `{DispWithArrows}` et sa version étoilée `{DispWithArrows*}` fournissent une commande `\WithArrowsRightX` qui peut être utilisée dans l'option `tikz-code`. Cette commande fournit l'abscisse de la marge droite de la boîte de composition courante (en prenant en compte les éventuels numéros des équations). Pour un exemple d'utilisation, voir p. 31.

10.4 La commande `\WithArrowsNewStyle`

L'extension `witharrows` fournit une commande `\WithArrowsNewStyle` pour définir des styles d'une manière similaire aux « styles » de Tikz.

La commande `\WithArrowsNewStyle` prend deux arguments obligatoires. Le premier est le nom du style et le second est une liste de couples clé-valeur. La portée de la définition ainsi faite par `\WithArrowsNewStyle` est le groupe TeX courant.²⁴

Le style peut être utilisé comme une clé au niveau du document (avec `\WithArrowsOptions`), au niveau d'un environnement (via les arguments optionnels) ou bien encore dans une autre commande `\WithArrowsNewStyle`.

Pour un exemple d'utilisation, voir p. 31.

Il n'existe pas de style pour les flèches individuelles. Néanmoins, il reste possible de définir de nouvelles commandes s'appuyant sur la commande `\Arrow`. Par exemple :

```
\newcommand{\ThickArrow}{\Arrow[tikz=thick]}
```

Cette nouvelle commande `\ThickArrow` garde la possibilité²⁵ d'accepter des clés entre crochets. On peut écrire `\ThickArrow[jump=2]` car en fait, `\Arrow[tikz=thick][jump=2]` est une syntaxe autorisée pour la commande `\Arrow` (on peut mettre un nombre arbitraire d'arguments optionnels entre crochets après la commande `\Arrow`).

10.5 La clé `right-overlap`

La clé `right-overlap` est une clé booléenne dont la valeur initiale est `true`. Elle concerne uniquement les environnements `{WithArrows}`.

Quand la clé `right-overlap` est active, les flèches (et leurs étiquettes) sont tracées en débordement et n'entrent donc pas dans le calcul de la boîte TeX contenant l'environnement `{WithArrows}`.

Quand la clé `right-overlap` est mise à `false` (dans `\WithArrowsOptions` ou bien dans un environnement `{WithArrows}` individuel) le débord vers la droite est pris en compte dans le calcul des dimensions de la boîte englobante.

24. On rappelle que, en particulier, tout environnement LaTeX est un groupe TeX.

25. Dans la version de `witharrows` pour LaTeX mais pas dans la version pour plain-TeX.

```

 $\left\{\begin{array}{l} 2x + 3y = 5 \\ -2x - 5y = 2 \end{array}\right. \xrightarrow{\text{on ajoute } L_1 \text{ à } L_2} \left\{\begin{array}{l} 2x + 3y = 5 \\ -2y = 7 \end{array}\right.$ 

```

Le réglage `right-overlap = false` peut aussi être utile en conjonction avec la classe `standalone`.

10.6 Positionnement vertical des flèches

Il y a quatre paramètres pour ajuster finement la position verticale des flèches : `ygap`, `ystart`, `start-adjust` et `end-adjust`.

Nous expliquons d'abord le comportement quand les paramètres `start-adjust` et `end-adjust` sont égaux à zéro :

- l'option `ystart` représente la distance verticale entre la ligne de base du texte et le début de la flèche (valeur initiale : 0.4 ex) ;
- l'option `ygap` représente la distance verticale entre deux flèches consécutives (valeur initiale : 0.4 ex).

$$\begin{aligned}
 (\cos x + \sin x)^2 &= \cos^2 x + 2 \cos x \sin x + \sin^2 x \xrightarrow{\text{ystart}} \\
 &= \cos^2 x + \sin^2 x + 2 \sin x \cos x \xrightarrow{\text{ygap}} \\
 &= 1 + \sin(2x)
 \end{aligned}$$

Néanmoins, pour des raisons esthétiques, quand il est possible, `witharrows` commence la flèche un peu plus haut (d'une quantité `start-adjust`) et termine la flèche un peu plus bas (par une quantité `end-adjust`). Par défaut, ces deux paramètres `start-adjust` et `end-adjust` sont égaux à 0.4 ex.

Voici par exemple le comportement sans le mécanisme de `start-adjust` et `end-adjust` :

```

 $\begin{array}{l} A = (a+1)^2 \\ = a^2 + 2a + 1 \end{array} \xrightarrow{\text{on développe}}$ 

```

$$\begin{array}{l} A = (a+1)^2 \\ = a^2 + 2a + 1 \end{array} \xrightarrow{\text{on développe}}$$

Voici le comportement par défaut (les paramètres `start-adjust` et `end-adjust` sont utilisés avec la valeur initiale 0.4 ex). La flèche est plus longue et le résultat est plus esthétique.

$$\begin{array}{l} A = (a+1)^2 \\ = a^2 + 2a + 1 \end{array} \xrightarrow{\text{on développe}}$$

Il est possible d'utiliser l'option `adjust` qui fixe à la fois `start-adjust` et `end-adjust`.

Une flèche de `jump` égal à 1 a une longueur²⁶ maximale égale au paramètre `max-length-of-arrow`. La valeur initiale de ce paramètre est égale à 2 cm.

26. On appelle *longueur* de la flèche la différence entre l'ordonnée du point de départ et l'ordonnée du point d'arrivée.

Dans l'exemple suivant, la valeur de `max-length-of-arrow` a été fixée à 1.5 cm.

```
\[\begin{WithArrows}[max-length-of-arrow = 1.5cm]
A
& =
\begin{vmatrix}
1 & a & a^2 & a^3 & a^4 \\
1 & b & b^2 & b^3 & b^4 \\
1 & c & c^2 & c^3 & c^4 \\
1 & d & d^2 & d^3 & d^4 \\
1 & e & e^2 & e^3 & e^4
\end{vmatrix}
\Arrow{
$L_2 \gets L_2-L_1$ \\
$L_3 \gets L_3-L_1$ \\
$L_4 \gets L_4-L_1$ \\
$L_5 \gets L_5-L_1$ % ne pas mettre de \\ ici
} \\
& =
\begin{vmatrix}
1 & a & a^2 & a^3 & a^4 \\
0 & b-a & b^2-a^2 & b^3-a^3 & b^4-a^4 \\
0 & c-a & c^2-a^2 & c^3-a^3 & c^4-a^4 \\
0 & d-a & d^2-a^2 & d^3-a^3 & d^4-a^4 \\
0 & e-a & e^2-a^2 & e^3-a^3 & e^4-a^4
\end{vmatrix}
\end{WithArrows}\]
```

$$\begin{aligned}
 A &= \begin{vmatrix} 1 & a & a^2 & a^3 & a^4 \\ 1 & b & b^2 & b^3 & b^4 \\ 1 & c & c^2 & c^3 & c^4 \\ 1 & d & d^2 & d^3 & d^4 \\ 1 & e & e^2 & e^3 & e^4 \end{vmatrix} \\
 &= \begin{vmatrix} 1 & a & a^2 & a^3 & a^4 \\ 0 & b-a & b^2-a^2 & b^3-a^3 & b^4-a^4 \\ 0 & c-a & c^2-a^2 & c^3-a^3 & c^4-a^4 \\ 0 & d-a & d^2-a^2 & d^3-a^3 & d^4-a^4 \\ 0 & e-a & e^2-a^2 & e^3-a^3 & e^4-a^4 \end{vmatrix} \begin{array}{l} \left. \begin{array}{l} L_2 \leftarrow L_2 - L_1 \\ L_3 \leftarrow L_3 - L_1 \\ L_4 \leftarrow L_4 - L_1 \\ L_5 \leftarrow L_5 - L_1 \end{array} \right\} \end{array}
 \end{aligned}$$

10.7 Notes de pied de page dans les environnements de `witharrows`

Si vous voulez mettre des notes de pied de page dans un environnement de `witharrows`, vous pouvez utiliser une paire `\footnotemark-\footnotetext`.

Il est aussi possible d'extraire les notes de pieds de page avec l'extension `footnote` ou bien l'extension `footnotehyper`.

Si `witharrows` est chargée avec l'option `footnote` (avec `\usepackage[footnote]{witharrows}` ou avec `\PassOptionsToPackage`), l'extension `footnote` est chargée (si elle ne l'est pas déjà) et elle est utilisée pour extraire les notes de pied de page.

Si `witharrows` est chargée avec l'option `footnotehyper`, l'extension `footnotehyper` est chargée (si elle ne l'est pas déjà) et elle est utilisée pour extraire les notes de pied de page.

Attention : Les extensions `footnote` et `footnotehyper` sont incompatibles. L'extension `footnotehyper` est le successeur de l'extension `footnote` et devrait être utilisée préférentiellement. L'extension `footnote` a quelques défauts ; en particulier, elle doit être chargée après l'extension `xcolor` et elle n'est pas parfaitement compatible avec `hyperref`.

Dans ce document, l'extension `witharrows` a été chargée avec l'option `footnotehyper` et nous donnons un exemple avec une note de pied de page dans l'étiquette d'une flèche :

$$A = (a + b)^2 \\ = a^2 + b^2 + 2ab \quad \left. \vphantom{A} \right\} \textit{on développe}^{27}$$

10.8 L'option `no-arrows`

L'option `no-arrows` est une facilité donnée à l'utilisateur. Avec cette option, les flèches ne sont pas tracées. Néanmoins, une analyse des flèches est effectuée et certaines erreurs sont détectées (par exemple, si une flèche arrive après la dernière ligne de l'environnement).

10.9 Note pour les utilisateurs de AUCTeX

Dans un éditeur de texte paramétré pour éditer du LaTeX, les environnements `{DispWithArrows}` et `{DispWithArrows*}` devraient être formatés comme l'environnement `{equation}` de LaTeX c'est-à-dire avec un formatage adapté au mode mathématique.

Dans l'éditeur Emacs avec AUCTeX, cela se fait en ajoutant les chaînes "`DispWithArrows`" et "`DispWithArrows*`" à la variable `font-latex-math-environments`. On peut le faire en utilisant l'interface de personnalisation de Emacs :

```
M-x customize > [Text] > [TeX] > [Font LaTeX]
```

10.10 Note pour les développeurs

Si vous définissez un environnement s'appuyant sur un environnement de `witharrows`, nous recommandons de faire appel à l'environnement de `witharrows` avec `\WithArrows-\endWithArrows` ou `\DispWithArrows-\endDispWithArrows` (et non `\begin{WithArrows}-\end{WithArrows}`, etc.).

En faisant ainsi, les messages d'erreur affichés par `witharrows` mentionneront (le cas échéant) le nom de votre environnement et seront plus faciles à comprendre pour l'utilisateur final.

Par exemple, vous pouvez définir un environnement `{DWA}` qui va être un alias de `{DispWithArrows}` :

```
\NewDocumentEnvironment {DWA} {} {\DispWithArrows}{\endDispWithArrows}
```

Si vous utilisez cet environnement `{DWA}` en mode mathématique, vous aurez le message d'erreur suivant :

```
The environment {DWA} should be used only outside math mode.
```

Une autre exemple est donné en interne par la définition de l'environnement `{DispWithArrows*}` dans l'extension `witharrows` avec le code suivant :

```
\NewDocumentEnvironment {DispWithArrows*} {}
  {\WithArrowsOptions{notag}%
   \DispWithArrows}
  {\endDispWithArrows}
```

11 Exemples

11.1 `\MoveEqLeft`

Il est possible d'utiliser `\MoveEqLeft` de `mathtools`. Dans ce cas, on doit se souvenir que la commande `\MoveEqLeft` a aussi la valeur d'une esperluette (&). Il faut en tenir compte pour le positionnement d'une éventuelle commande `\Arrow`.

27. Une note de pied de page.

```


$$\arccos(x) = \arcsin \frac{4}{5} + \arcsin \frac{5}{13}$$


```

parce que les deux sont dans $[-\frac{\pi}{2}, \frac{\pi}{2}]$

```


$$\Leftrightarrow x = \sin \left( \arcsin \frac{4}{5} + \arcsin \frac{5}{13} \right)$$


$$\Leftrightarrow x = \frac{4}{5} \cos \arcsin \frac{5}{13} + \frac{5}{13} \cos \arcsin \frac{4}{5}$$


$$\Leftrightarrow x = \frac{4}{5} \sqrt{1 - \left(\frac{5}{13}\right)^2} + \frac{5}{13} \sqrt{1 - \left(\frac{4}{5}\right)^2}$$


```

$$\arccos(x) = \arcsin \frac{4}{5} + \arcsin \frac{5}{13}$$

$$\Leftrightarrow x = \sin \left(\arcsin \frac{4}{5} + \arcsin \frac{5}{13} \right)$$

$$\Leftrightarrow x = \frac{4}{5} \cos \arcsin \frac{5}{13} + \frac{5}{13} \cos \arcsin \frac{4}{5}$$

$$\Leftrightarrow x = \frac{4}{5} \sqrt{1 - \left(\frac{5}{13}\right)^2} + \frac{5}{13} \sqrt{1 - \left(\frac{4}{5}\right)^2}$$

) $\forall x \in [-1, 1], \cos(\arcsin x) = \sqrt{1 - x^2}$

11.2 Une commande `\DoubleArrow`

En utilisant l'option `o` (cf. p. 10) disponible au niveau local, on peut facilement écrire une commande `\DoubleArrow` pour deux flèches allant en sens inverse l'une de l'autre.

```

\NewDocumentCommand \DoubleArrow { 0 {} m m }
{
  \Arrow[tikz=->,#1]{#2}%
  \Arrow[o,tikz=<-,#1]{#3}
}

```

Exemple d'utilisation :

```


$$(a+b)^2 \xleftrightarrow{\text{développement}} a^2 + 2ab + b^2 \xleftrightarrow{\text{factorisation}} (a+b)^2$$


```

$$A = (a + b)^2$$

$$= a^2 + 2ab + b^2$$

) développement) factorisation

11.3 Modifier la forme des nœuds

Il est possible de changer la forme des étiquettes, qui sont des nœuds Tikz, en modifiant la clé « `every node` » de Tikz.

```

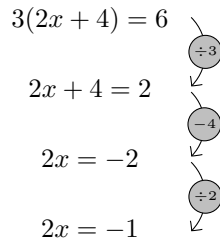
\begin{WithArrows}%
  [format = c,
   interline = 4mm,
   tikz = {every node/.style = {circle,
                                draw,
                                auto = false,
                                fill = gray!50,
                                inner sep = 1pt,
                                font = \tiny}}]
  3 (2x+4) = 6 \Arrow{\$/div 3$} \\  

  2x+4 = 2 \Arrow{\$/-4$} \\  

  2x = -2 \Arrow{\$/div 2$} \\  

  2x = -1
\end{WithArrows}

```



11.4 Exemples avec l'option tikz-code

Rappelons que l'option `tikz-code` est le code Tikz utilisé par `witharrows` pour tracer les flèches.²⁸

La valeur initiale de `tikz-code` est `\draw (#1) to node {#3} (#2)` ; où les trois marqueurs `#1`, `#2` et `#3` représentent le point de départ, le point d'arrivée et l'étiquette de la flèche.

11.4.1 Exemple 1

Dans l'exemple suivant, nous définissons la valeur de `tikz-code` avec deux instructions `\path` : la première instruction trace la flèche elle-même et la seconde place l'étiquette dans un nœud Tikz dans le rectangle délimité par la flèche.

```

\begin{DispWithArrows*}%
  [displaystyle,
   ygap = 2mm,
   ystart = 0mm,
   tikz-code = {\draw (#1) -- ++(4.5cm,0) |- (#2) ;
                \path (#1) -- (#2)
                  node[text width = 4.2cm, right, midway] {#3} ;}]
S_n
& = \frac{1}{n} \sum_{k=0}^{n-1} \cos\bigl(\tfrac{\pi}{2}\cdot\tfrac{k}{n}\bigr)
.....

```

| | |
|--|--|
| $S_n = \frac{1}{n} \sum_{k=0}^{n-1} \cos\left(\frac{\pi}{2} \cdot \frac{k}{n}\right)$ | $\cos x = \Re(e^{ix})$ |
| $= \frac{1}{n} \sum_{k=0}^{n-1} \Re\left(e^{i \frac{k\pi}{2n}}\right)$ | $\Re(z + z') = \Re(z) + \Re(z')$ |
| $= \frac{1}{n} \Re\left(\sum_{k=0}^{n-1} e^{i \frac{k\pi}{2n}}\right)$ | <i>exp est un morphisme pour</i> \times et $+$ |
| $= \frac{1}{n} \Re\left(\sum_{k=0}^{n-1} \left(e^{i \frac{\pi}{2n}}\right)^k\right)$ | <i>somme de termes d'une suite</i> <i>géométrique de raison $e^{i \frac{2\pi}{n}}$</i> |
| $= \frac{1}{n} \Re\left(\frac{1 - \left(e^{i \frac{\pi}{2n}}\right)^n}{1 - e^{i \frac{\pi}{2n}}}\right)$ | |
| $= \frac{1}{n} \Re\left(\frac{1 - i}{1 - e^{i \frac{\pi}{2n}}}\right)$ | |

²⁸ Si un environnement `{DispWithArrows}` ou `{DispWithArrows*}` est utilisé avec l'option `wrap-lines`, la valeur de l'option `tikz-code` n'est pas utilisée pour cet environnement (mais est utilisée pour les environnements imbriqués à l'intérieur).

11.4.2 Exemple 2

Il est possible de modifier l'exemple précédent pour avoir « `text width` » calculé automatiquement en fonction de la marge droite (d'une manière similaire à l'option `wrap-lines`) dans les environnements `{DispWithArrows}` et `{DispWithArrows*}`. Dans la définition de `tikz-code`, nous utilisons la commande `\WithArrowsRightX` qui donne l'abscisse de la marge droite de la boîte de composition (pour TeX, il s'agit d'une commande et non d'une dimension). Pour la lisibilité, nous avons utilisé un style de `witharrows`. Cet exemple nécessite l'extension Tikz calc.

```
\WithArrowsNewStyle{MonStyle}
  {displaystyle,
   ygap = 2mm,
   xoffset = 0pt,
   ystart = 0mm,
   tikz-code = {\path let \p1 = (#1)
                 in (#1)
                 -- node [anchor = west,
                          text width = {\WithArrowsRightX - \x1 - 0.5 em}]
                          {#3}
                 (#2) ;
   \draw let \p1 = (#1)
          in (#1) -- ++(\WithArrowsRightX - \x1,0) |- (#2) ; }}

\begin{DispWithArrows}[MonStyle]
  S_n
  & = \frac{1}{n} \sum_{k=0}^{n-1} \cos\bigl(\tfrac{\pi}{2}\cdot\tfrac{k}{n}\bigr)
  \Arrow{$\cos x = \Re(e^{ix})$}\
  .....
```

$$S_n = \frac{1}{n} \sum_{k=0}^{n-1} \cos\left(\frac{\pi}{2} \cdot \frac{k}{n}\right) \quad (18)$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} \Re\left(e^{i \frac{k\pi}{2n}}\right) \quad (19)$$

$$= \frac{1}{n} \Re\left(\sum_{k=0}^{n-1} e^{i \frac{k\pi}{2n}}\right) \quad (20)$$

$$= \frac{1}{n} \Re\left(\sum_{k=0}^{n-1} \left(e^{i \frac{\pi}{2n}}\right)^k\right) \quad (21)$$

$$= \frac{1}{n} \Re\left(\frac{1 - \left(e^{i \frac{\pi}{2n}}\right)^n}{1 - e^{i \frac{\pi}{2n}}}\right) \quad (22)$$

$$= \frac{1}{n} \Re\left(\frac{1 - i}{1 - e^{i \frac{\pi}{2n}}}\right) \quad (23)$$

11.4.3 Exemple 3

Dans l'exemple suivant, nous changeons la forme de la flèche selon que la rangée de départ est plus longue ou plus courte que la rangée d'arrivée. Cet exemple nécessite l'extension `calc` de Tikz.

```
\begin{WithArrows}[ll,interline=5mm,xoffset=5mm,
  tikz-code = {\draw[rounded corners,
                    every node/.style = {circle,
                                          draw,
                                          auto = false,
```

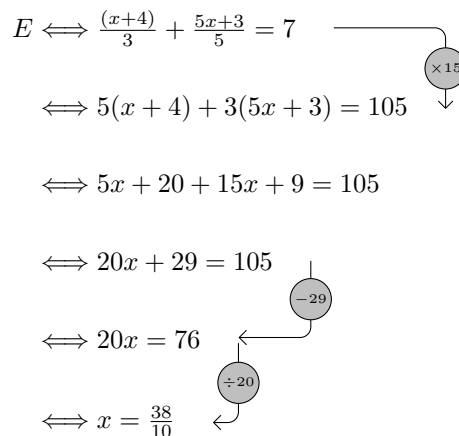
```

inner sep = 1pt,
fill = gray!50,
font = \tiny }]

let \p1 = (#1),
    \p2 = (#2)
and \ifdim \x1 > \x2
    (\p1) -- node {#3} (\x1,\y2) -- (\p2)
\else
    (\p1) -- (\x2,\y1) -- node {#3} (\p2)
\fi ;}]

E & \Longleftarrow \frac{(x+4)}{3} + \frac{5x+3}{5} = 7
\Arrow{$\times 15$}\
& \Longleftarrow 5(x+4) + 3(5x+3) = 105 \
& \Longleftarrow 5x+20 + 15x+9 = 105 \
& \Longleftarrow 20x+29 = 105
\Arrow{$-29$}\
& \Longleftarrow 20x = 76
\Arrow{$\div 20$}\
& \Longleftarrow x = \frac{38}{10}
\end{WithArrows}

```



11.5 Boucle numérotée automatique

Supposons que nous voulions tracer une boucle avec des flèches numérotées. Dans ce but, il est possible d'écrire une commande dédiée `\NumberedLoop` qui fera le travail quand elle sera utilisée dans le `code-after`. Dans l'exemple suivant, nous avons écrit cette commande avec `\foreach` de `pgffor` (cette extension est chargée quand `witharrows` est chargé.).

```

\NewDocumentCommand \NumberedLoop {}
  {\foreach \j in {2,...,\WithArrowsNbLines}
    { \pgfmathtruncatemacro{\i}{\j-1}
      \Arrow[rr]{\i}{\j}{\i} }
  \Arrow[rr,xoffset=1cm,tikz=<-]{1}{\WithArrowsNbLines}{\WithArrowsNbLines}}

```

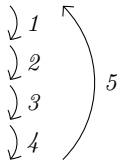
La commande `\WithArrowsNbLines` est une commande disponible dans le `code-after` qui donne le nombre total de lignes (=rangées) dans l'environnement courant (pour TeX, il s'agit d'une commande et non d'un compteur).


```

 $\begin{WithArrows}[code-after = \NumberedLoop]
a.\;& f \text{ est continue sur } E \\
b.\;& f \text{ est continue en } 0 \\
c.\;& f \text{ est bornée sur la sphère unité} \\
d.\;& \exists K > 0 \quad \forall x \in E \quad \|f(x)\| \leq K \|x\| \\
e.\;& f \text{ est lipschitzienne}
\end{WithArrows}$ 

```

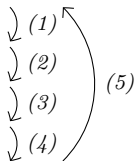
a. f est continue sur E
b. f est continue en 0
c. f est bornée sur la sphère unité
d. $\exists K > 0 \quad \forall x \in E \quad \|f(x)\| \leq K \|x\|$
e. f est lipschitzienne



Comme d'habitude, il est possible de changer les caractéristiques des flèches et des nœuds avec l'option `tikz`. Néanmoins, si nous voulons changer le style pour avoir, par exemple, les numéros entre parenthèses, le moyen le plus rapide est de changer la valeur de `tikz-code` :

```
tikz-code = {\draw (#1) to node {\footnotesize (#3)} (#2) ;}
```

a. f est continue sur E
b. f est continue dans 0
c. f est bornée sur la sphère unité
d. $\exists K > 0 \quad \forall x \in E \quad \|f(x)\| \leq K \|x\|$
e. f est lipschitzienne



Autre documentation

Le document `witharrows.pdf` (fourni avec l'extension `witharrows`) contient une traduction anglaise de la documentation ici présente, ainsi que le code source commenté et un historique des versions.

Table des matières

| | | |
|---|--|----|
| 1 | Options pour la forme des flèches | 2 |
| 2 | Nombre et formats des colonnes | 6 |
| 3 | Positionnement précis des flèches | 7 |
| 4 | L'option « o » pour des flèches individuelles | 10 |
| 5 | Les options « up » et « down » pour des flèches individuelles | 11 |
| 6 | Comparaison avec l'environnement <code>{aligned}</code> | 12 |
| 7 | Les flèches dans les environnements imbriqués | 14 |
| 8 | Dessiner des flèches depuis l'extérieur des environnements <code>{WithArrows}</code> | 16 |
| 9 | L'environnement <code>{DispWithArrows}</code> | 18 |

| | | |
|-----------|--|-----------|
| 10 | Fonctionnalités avancées | 23 |
| 10.1 | Utilisation avec Beamer | 23 |
| 10.2 | Utilisation avec plain-TeX | 24 |
| 10.3 | L'option tikz-code : comment changer la forme des flèches | 24 |
| 10.4 | La commande <code>\WithArrowsNewStyle</code> | 25 |
| 10.5 | La clé <code>right-overlap</code> | 25 |
| 10.6 | Positionnement vertical des flèches | 26 |
| 10.7 | Notes de pied de page dans les environnements de <code>witharrows</code> | 27 |
| 10.8 | L'option <code>no-arrows</code> | 28 |
| 10.9 | Note pour les utilisateurs de AUCTeX | 28 |
| 10.10 | Note pour les développeurs | 28 |
| 11 | Exemples | 28 |
| 11.1 | <code>\MoveEqLeft</code> | 28 |
| 11.2 | Une commande <code>\DoubleArrow</code> | 29 |
| 11.3 | Modifier la forme des nœuds | 29 |
| 11.4 | Exemples avec l'option tikz-code | 30 |
| 11.4.1 | Exemple 1 | 30 |
| 11.4.2 | Exemple 2 | 31 |
| 11.4.3 | Exemple 3 | 31 |
| 11.5 | Boucle numérotée automatique | 32 |