

2 – Coordonnées

Il y a plusieurs façons d'écrire les coordonnées d'un point (ou d'un vecteur) ; les deux plus courantes sont $A(3, -2)$ et $A(3; -2)$. Les anglo-saxons qui utilisent le point comme séparateur décimal se contentent de la virgule entre l'abscisse et l'ordonnée ; nous sommes obligés d'utiliser le point-virgule pour séparer des nombres qui contiennent déjà une virgule : $B(2,5;3)$.

Mais il peut être pratique pour plus de lisibilité, et pour faciliter les calculs entre les abscisses et les ordonnées (voire les cotes), d'écrire les coordonnées « en colonnes » : $A\begin{pmatrix} 3 \\ -2 \end{pmatrix}$.

Certains préféreront un trait vertical : $A\left| \begin{array}{c} 3 \\ -2 \end{array} \right.$.

Il existe en L^AT_EX différents environnements qui permettent de faire ce que l'on veut.

1 Première méthode avec `pmatrix`

Une écriture comme $A\begin{pmatrix} 3 \\ -2 \end{pmatrix}$ s'obtient en utilisant l'environnement `pmatrix` :

```
$A
\begin{pmatrix}
3 \\
-2
\end{pmatrix}$
```

On peut retenir que le `p` de `pmatrix` signifie « parenthèses ».

L'environnement `pmatrix` n'est valide qu'en mode mathématique, d'où la présence du `$` avant d'appeler l'environnement, et la présence du même `$` quand on en sort.

On crée en fait une matrice d'une colonne et de deux lignes, le passage d'une ligne à une autre se faisant par `\\`.

Pas de problème dans un repère de l'espace où l'on aura $A\begin{pmatrix} 3 \\ -2 \\ 7 \end{pmatrix}$ en rajoutant une troisième ligne.

2 Deuxième méthode avec `array`

Une écriture comme $A\left| \begin{array}{c} 3 \\ -2 \end{array} \right.$ s'obtient en utilisant l'environnement `array`.

Cet environnement permet d'écrire des tableaux en mode mathématique.

Si on veut écrire du texte présenté sous forme de tableau, on utilisera l'environnement `tabular`.

Voici ce qu'il faut entrer :

```
$A
\begin{array}{|c}
3 \\
-2
\end{array}$
```

Le `c` en paramètre signifie que les nombres seront centrés ; on peut les aligner à gauche (en entrant `l` comme `left` à la place du `c`) ou à droite (avec `r` comme `right`).

Le `|` devant le `c` permet de tracer le trait vertical à gauche : si on veut également un trait à droite, on entrera `{|c|}`.

En fait, si on entre exactement ce qui est au dessus, on obtient $A \left| \begin{array}{c} 3 \\ -2 \end{array} \right.$ ce qui n'est pas très beau!

Il faut donc séparer le nom du point (ou du vecteur) et le trait vertical par une espace (en matière d'imprimerie, le mot « espace » est du genre féminin).

Il existe plusieurs espacements possibles en mode mathématique; en voici trois, du plus petit au plus grand : `\`, `\;`, `\l` (attention : le `\l` signifie « appui sur la barre d'espacement » !)

	<code>\$A\;</code>		
	<code>\begin{array}{ c}</code>		
Donc en tapant	<code>3 \\\</code>	on obtient	$A \left \begin{array}{c} 3 \\ -2 \end{array} \right.$
	<code>-2</code>		
	<code>\end{array}\$</code>		

3 Autres environnements

Il existe d'autres environnements matriciels qui peuvent être intéressants en d'autres circonstances :

En tapant	<code>\$\$\begin{matrix}</code>		
	<code>3 \\\</code>	on obtient	$\begin{matrix} 3 \\ -2 \end{matrix}$
	<code>-2</code>		
	<code>\end{matrix}\$</code>		
En tapant	<code>\$\$\begin{bmatrix}</code>		
	<code>3 \\\</code>	on obtient	$\begin{bmatrix} 3 \\ -2 \end{bmatrix}$
	<code>-2</code>		
	<code>\end{bmatrix}\$</code>		
En tapant	<code>\$\$\begin{vmatrix}</code>		
	<code>3 \\\</code>	on obtient	$\begin{vmatrix} 3 \\ -2 \end{vmatrix}$
	<code>-2</code>		
	<code>\end{vmatrix}\$</code>		
En tapant	<code>\$\$\begin{Vmatrix}</code>		
	<code>3 \\\</code>	on obtient	$\begin{Vmatrix} 3 \\ -2 \end{Vmatrix}$
	<code>-2</code>		
	<code>\end{Vmatrix}\$</code>		
Par exemple pour écrire	$\begin{vmatrix} 3 & -5 & 9 \\ -2 & 8 & 7 \\ 1 & -6 & 11 \end{vmatrix}$	on entrera	<code>\$\$\begin{vmatrix}</code>
			<code>3 & -5 & 9\\\</code>
			<code>-2 & 8 & 7\\\</code>
			<code>1 & -6 & 11</code>
			<code>\end{vmatrix}\$</code>

Au passage, on voit que `&` est le séparateur de colonnes dans un tableau ou une matrice.

4 Automatisation

C'est bien joli tout ça, mais s'il faut taper des choses du style

```
$A
\begin{pmatrix}
3 \\\
-2
\end{pmatrix}$
```

chaque fois qu'on veut écrire les coordonnées d'un point, on n'a pas fini (et on regrette de s'être mis à L^AT_EX!!!).

Mais non !

Justement L^AT_EX permet de créer des commandes qui automatisent ce que l'on écrit souvent.

On va donc créer une commande que l'on va appeler `coo`, qui va nécessiter deux paramètres (l'abscisse et l'ordonnée) et qui va automatiquement écrire les coordonnées comme on le souhaite. Il faut donc écrire dans le préambule (avant le `\begin{document}`) :

```
\newcommand{\coo}[2]
{
\begin{pmatrix}
#1 \\
#2
\end{pmatrix}
}
```

On a ainsi défini une nouvelle commande `coo` à deux paramètres (c'est ce que veut dire [2]), qui s'appellent #1 et #2 ; on affectera ces paramètres en mettant les valeurs souhaitées entre accolades. On utilise cette commande en écrivant par exemple `$A\coo{3}{-2}$`, ce qui donne le résultat espéré $A \begin{pmatrix} 3 \\ -2 \end{pmatrix}$.

Pour l'espace, on va définir une nouvelle commande (à trois paramètres) `cooe` ainsi :

```
\newcommand{\cooe}[3]
{
\begin{pmatrix}
#1 \\
#2 \\
#3
\end{pmatrix}
}
```

qui permettra d'obtenir $A \begin{pmatrix} 3 \\ -2 \\ 7 \end{pmatrix}$ en tapant `$A\cooe{3}{-2}{7}$`.

La définition d'une commande peut se faire sur une seule ligne

```
\newcommand{\cooe}[3]{\begin{pmatrix} #1 \\ #2 \\ #3 \end{pmatrix} }
```

mais c'est nettement moins lisible.

On peut créer des commandes sans paramètre, comme la commande `\R` pour écrire rapidement \mathbb{R} :

```
\newcommand{\R}{\mathbb R}
```

Personnellement, dans mes textes, j'écris **R** (en gras) pour l'ensemble des réels, réservant la double barre à l'écriture manuscrite ; ma commande `\R` est donc :

```
\newcommand{\R}{\mathbf R}
```

Enfin il faut savoir que, dans une nouvelle commande, on ne peut pas passer plus de 9 paramètres, mais on travaille rarement dans un espace de dimension supérieure à 9...