

# Tableaux

## Tableaux

## I. Composer un tableau

### A. Créer des lignes et des colonnes

Pour créer un tableau, aucun *package* supplémentaire n'est chargé. L'environnement *tabular* sera utilisé en spécifiant un argument obligatoire composé d'une des trois lettres suivantes : l, c ou r. On peut inscrire plusieurs de ces lettres et une même lettre peut apparaître plusieurs fois.

Le nombre de lettres inscrites correspond simplement au nombre de colonne que contiendra le tableau. Dans chacune des colonnes, le texte pourra alors être aligné à gauche (l), centré (c) ou aligné à droite (r). Les lettres sont des *spécificateurs de colonnes*.

Ainsi, l'instruction `\begin{tabular}{rcl}` permettra la création d'un tableau contenant trois colonnes, dont le texte sera aligné à droite dans la première colonne, centré dans la deuxième et enfin aligné à gauche dans la troisième. Il est inutile de spécifier le nombre de lignes.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
	Total	40

```

1 \begin{tabular}{rcl}
2   Article & Couleur & Prix en euros \\[0.5cm]
3   Pantalon & bleu & 25 \\
4   Gants & blanc & 15 \\
5   & Total & 40
6 \end{tabular}

```

Comme pour l'environnement *align*, `&` spécifie l'emplacement de l'alignement : autrement dit ce symbole sépare chaque colonne, y compris les colonnes vides.

La commande de changement de ligne `\\` admet un argument optionnel qui sert à indiquer l'espace verticale que l'on veut insérer après cette ligne.

Afin de mieux visualiser les différentes lignes et colonnes, on peut ajouter des *filets* : la commande `\hline` trace un *filet horizontal*. Pour insérer un *filet vertical*, on peut inscrire une barre verticale `|` entre les deux spécificateurs de colonnes concernés.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
	Total	40

```

1 \begin{tabular}{r|cl}
2   Article & Couleur & Prix en euros \\ \hline\hline
3   Pantalon & bleu & 25 \\
4   Gants & blanc & 15 \\
5   & Total & 40
6 \end{tabular}

```

### B. Fusionner des colonnes

L<sup>A</sup>T<sub>E</sub>X offre la commande `\multicolumn{(nombre-colonnes)}{<specificateur>}{<text>}` qui permet de fusionner horizontalement des colonnes.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
Total :		40

```

1 \begin{tabular}{r|cl}
2   \multicolumn{1}{c|}{Article} & & \\
3   Couleur & Prix en euros \\ \hline
4   Pantalon & bleu & 25 \\
5   Gants & blanc & 15 \\
6   \multicolumn{2}{r}{Total :} & 40
7 \end{tabular}

```



On constate dans l'exemple précédent que la commande `\multicolumn` permet non seulement de fusionner plusieurs colonnes mais aussi de modifier ponctuellement le spécificateur de colonne sur une cellule en particulier.


## C. Spécificateurs de colonne supplémentaires

Les trois spécificateurs de base ne permettent pas de changement de ligne au sein d'une même cellule. De plus, on ne peut pas spécifier la largeur de la cellule. Le spécificateur `p{<dim>}` permet de composer une colonne en imposant une largeur : le texte est alors composé comme un paragraphe normal.

Rectangle	Quadrilatère dont les diagonales sont de même longueur et se coupent en leur milieu. Le rectangle est un parallélogramme.	<pre> 1 \begin{tabular}{r p{4.5cm}} 2   Rectangle &amp; Quadrilatère dont les diagonales 3   sont de même longueur et 4   se coupent en leur milieu.\par 5   Le rectangle est un parallélogramme. 6 \end{tabular} </pre>
-----------	--	--

Dans l'exemple précédent, on constate que les deux paragraphes sont alignés sur le haut de la cellule. On pourrait avoir envie de centrer verticalement ces paragraphes. On utilisera simplement `m{<dim>}` :


Rectangle	Quadrilatère dont les diagonales sont de même longueur et se coupent en leur milieu. Le rectangle est un parallélogramme.	<pre> 1 \begin{tabular}{r m{4.5cm}} 2   Rectangle &amp; Quadrilatère dont les diagonales 3   sont de même longueur et 4   se coupent en leur milieu.\par 5   Le rectangle est un parallélogramme. 6 \end{tabular} </pre>
-----------	--	--

 Dans l'exemple précédent, remplacer `m{4.5cm}` par `b{4.5cm}`. Que se passe-t-il ?

### Exercice 1

Comment composer le tableau suivant sachant que les deux colonnes extérieures mesurent 2 cm alors que la colonne centrale mesure 8 cm ?

1 <sup>re</sup> s	Jeudi 13 novembre 2014	Étude de fonctions
CONTRÔLE DE MATHÉMATIQUES		
Nom :		
Prénom :		
Note et observations :		

 On pourra regarder avec beaucoup d'attention la documentation du *package* **tabularx** qui offre l'environnement *tabularx* qui permet de spécifier la largeur totale du tableau et donc de calculer ensuite automatiquement la largeur de colonnes spécifiques.

## II. Les tableaux en mathématiques

### A. L'environnement *array*

Comme nous allons rapidement le voir, il peut être utile de composer des tableaux en mode mathématique (notamment en mode *hors texte*). Dans ce cas, on utilise dans le mode l'environnement *array* qui fonctionne comme *tabular* pour les commandes expliquées dans la section précédente.

$  \begin{array}{lcl}  f: \mathbb{R} & \rightarrow & \mathbb{R}_+ \\  x & \mapsto & x^2  \end{array}  $	<pre> 1 \[ \begin{array}{rcl} 2   f \colon \mathbb{R} &amp; \rightarrow &amp; \mathbb{R}_+ \\ 3   x &amp; \mapsto &amp; x^2 4 \end{array} \] </pre>
---	---

## B. Les systèmes

On avait constaté l'utilité des symboles étirables horizontalement. Servons-nous de `\left\{` pour créer une grande accolade pour les systèmes :

$$\begin{cases} 2x - y - 3z = 1 \\ 3x + 2y = -4 \\ -x + 6z = 22 \end{cases}$$

```
1 \left\{
2   \begin{array}{cccccl}
3     2x & - & y & - & 3z & = & 1 \\
4     3x & + & 2y & & & = & -4 \\
5     -x & & & + & 6z & = & 22
6   \end{array}
7 \right.\]
```



`\left` et `\right` fonctionne toujours ensemble. Ici, on ne veut qu'une accolade à gauche. On utilise donc `\left{` mais la commande `\right.` est nécessaire pour que la compilation ait lieu sans erreur.

$$|x| = \begin{cases} -x & \text{si } x < 0 \\ x & \text{sinon} \end{cases}$$

```
1 \left\{
2   \begin{array}{cl}
3     -x & \text{si } x < 0 \\
4     x & \text{sinon}
5   \end{array}
6 \right.\]
```

## C. Les matrices

Cette fois-ci, on peut penser à utiliser `\left(` (et `\right)`)...

$$M = \begin{pmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{pmatrix}$$

```
1 \left(
2   \begin{array}{ccc}
3     2 & -1 & 0 \\
4     3 & 4 & 1 \\
5     0 & 2 & 3
6   \end{array}
7 \right)
```

... mais en réalité des environnements spécifiques existent pour l'écriture des matrices. Entre autre, on retiendra les environnements `pmatrix` et `vmatrix` ainsi que la commande `\bordermatrix`.

$$M = \begin{pmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{pmatrix}$$

$$\det(M) = \begin{vmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{vmatrix} = 29$$

$$\begin{matrix} f(e_1) & f(e_2) \\ e_1 & \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix} \\ e_2 & \end{matrix}$$

```
1 \[M =
2   \begin{pmatrix}
3     2 & -1 & 0 \\
4     3 & 4 & 1 \\
5     0 & 2 & 3
6   \end{pmatrix}
7 \]
8 \det(M) =
9   \begin{vmatrix}
10    2 & -1 & 0 \\
11    3 & 4 & 1 \\
12    0 & 2 & 3
13  \end{vmatrix} = 29
14 \]
15 \[
16   \bordermatrix{
17     & f(e_1) & f(e_2) \\
18   e_1 & 1 & 2 \\
19   e_2 & 0 & 3
20 }
```

### III. Tableaux de signes et tableaux de variations

Ici du allons utiliser l'environnement *tikzpicture* du *package* **TikZ**. Nous ne dirons rien sur ce *package* pour le moment mais nous reviendrons dessus lorsque nous traiterons les graphiques. Cependant, pour créer des tableaux de signes et des tableaux de valeurs, Alain MATTHES a créé le *package* **tkz-tab** qui répond parfaitement bien à notre problème. Il faut donc penser dès à présent à ajouter `\usepackage{tkz-tab}` au préambule.



Selon l'installation effectuée, il est possible que **tkz-tab** ne soit pas présent dans votre distribution. Vous vous en rendrez rapidement compte en compilant le premier exemple. La documentation de ce *package* est également une source à consulter absolument.

#### A. Tableaux de signes

```

1 \textbf{Tableau de signes :}\par
2 \begin{tikzpicture}
3   \tkzTabInit[nocadre,espc1=1.5]%
4     {\$x\$/0.75,Signe de \$\$x+2\$/1.5,Signes de \$\$x^2-1\$/1.5,%
5       signe du \$\$ produit\$/1.5,signe du \$\$ quotient\$/1.5}%
6     {\$-\infty\$, \$-2\$, \$-1\$, \$1\$, \$+\infty\$}
7   \tkzTabLine{,-,z,+,t,+,t,+}
8   \tkzTabLine{+,t,+,z,-,z,+}
9   \tkzTabLine{-,z,+,z,-,z,+}
10  \tkzTabLine{-,z,+,d,-,d,+}
11 \end{tikzpicture}

```

Tableau de signes :							
x	$-\infty$	$-2$	$-1$	$1$	$+\infty$		
Signe de $x + 2$	$-$	0	$+$	$+$	$+$		
Signes de $x^2 - 1$	$+$	$+$	0	$-$	0	$+$	
signe du produit	$-$	0	$+$	0	$-$	0	$+$
signe du quotient	$-$	0	$+$	$-$	$+$		

Un tableau de signes ou de variations commencera toujours par `\tkzTabInit` dont voilà la syntaxe :

`\tkzTabInit[(options)]{\langle première colonne \rangle}{\langle première ligne \rangle}`

En option est spécifié que le cadre autour du tableau ne doit pas être dessiné et que l'espace entre les valeurs de la première ligne est réglée par le paramètre `espc1`.

Ensuite, on énumère les différentes lignes de la première colonne : *\langle nom de la ligne \rangle* / *\langle hauteur de la ligne \rangle*. Le *\langle nom de la ligne \rangle* accepte des changements de lignes à l'aide de `\\` et on passe d'une ligne à l'autre en utilisant la virgule.

Enfin, dans le dernier argument, on écrit les valeurs de la première ligne en les séparant par une virgule.

Pour finir, pour chaque ligne, on écrit les signes par la commande `\tkzTabLine`. La lettre `t` crée un filet vertical en pointillés et la lettre `z` fait la même chose en ajoutant un zéro. La lettre `d` insère une double-barre pour les valeurs interdites.



Si on a besoin d'écrire des nombres décimaux, on prendra soin de les écrire entre accolades pour que la virgule ne crée pas de conflit : `\$ {1,5} \$`.

## B. Tableaux de variations

```

1 \textbf{Tableau de variations :}\par
2 \begin{tikzpicture}
3   \tkzTabInit[nocadre,espc1=2]{\textit{x}/0.75,Variations \textit{de f(x)}/1.75}{\$-3\$,\$-1\$,\$1\$,\$4\$}
4   \tkzTabVar{+/\$6\$,-D+/\$-\infty\$/\$+\infty\$,-/,-/,\$-5\$}
5 \end{tikzpicture}

```

**Tableau de variations :**

$x$	$-3$	$-1$	$1$	$4$
Variations de $f(x)$	6 ↘ $-\infty$	$+\infty$ ↘ $-\infty$	$+\infty$ ↗ $-\infty$	$-5$

Il nous suffit ici de commenter la commande `\tkzTabVar` : pour chaque valeurs de  $x$  indiquée sur la première ligne, on peut préciser une valeur précédée du signe  $+/$  pour dire que la valeur sera écrite « en haut » ou bien  $-/$  pour dire que la valeur sera écrite « en bas ». Des flèches relieront alors automatiquement les différentes valeurs :

`\tkzTabVar{(+ ou -)/\langle valeur_1 \rangle , (+ ou -)/\langle valeur_2 \rangle , ...}`



Pour la double-barre avec des valeurs indiquées avant et après celle-ci, on notera `-D+/\langle valeur_1 \rangle /\langle valeur_2 \rangle` ou bien `+D-/\langle valeur_1 \rangle /\langle valeur_2 \rangle`.

## C. Un mélange

Et voilà maintenant ce que l'on peut obtenir :

```

1 \begin{tikzpicture}
2   \tkzTabInit[nocadre,espc1=2]%
3     {\textit{x}/0.75,signe de \textit{f'(x)}/1.5,Variations \textit{de f(x)}/1.75}%
4     {\$-3\$,\$-1\$,\$1\$,\$4\$}
5   \tkzTabLine{+,z,-,z,+}
6   \tkzTabVar{-/\$-\infty\$,\$+/\$-\infty\$,\$+/\$+\infty\$}
7 \end{tikzpicture}

```

x	-3	-1	1	4	
signe de f'(x)	+	0	-	0	+
Variations de f(x)	<div><div><div></div><div><math>-\infty</math></div></div><div><div></div><div></div></div><div><div></div><div><math>+\infty</math></div></div></div>				

## IV. Exercices

### Exercice 2

Comment obtenir le Q.C.M. suivant ?

	Réponse A	Réponse B	Réponse C
<b>Question 1.</b>	1a	1b	1c
<b>Question 2.</b>	2a	2b	2c

### Exercice 3

Composer le code source de l'énoncé suivant puis taper la correction.

**Exercice.**

On considère la fonction  $f$  définie de la façon suivante :

$$\begin{aligned} f: \mathbb{R} &\rightarrow \mathbb{R} \\ x &\mapsto \frac{1}{3}x^3 + x^2 - 3x + 4 \end{aligned}$$

- 1°) Déterminer la dérivée  $f'$  de  $f$  sur  $\mathbb{R}$ .
- 2°) Déterminer les racines de  $f'$ .
- 3°) À l'aide du tableau de signes de  $f'$ , dresser le tableau de variations de  $f$  sur  $\mathbb{R}$ .

## V. Complément : le tableur

Le *package* **pas-tableur** de Stéphane PASQUET permet d'imiter l'apparence d'un tableur. Cependant, il n'effectue pas de calcul automatique comme dans un tableur (pour cela, on pourra jeter un œil sur le *package* **spreadtab**). Une fois encore, ce *package* utilise **TikZ** et son environnement *tikzpicture*.

La première commande à retenir est la suivante :

$$\backslash\text{tableur}[\langle\text{nombre-lignes}\rangle][\langle\text{colonnes}\rangle]$$

L'argument  $\langle\text{colonnes}\rangle$  permet de spécifier les lettres des colonnes utilisées soit une par une, soit en utilisant un « intervalle ».

Ensuite,

$$\backslash\text{celttext}[\langle\text{options}\rangle][\langle\text{colonne}\rangle][\langle\text{ligne}\rangle][\langle\text{texte}\rangle]$$

permet d'écrire un texte dans la cellule définie par  $\langle\text{colonne}\rangle$  et  $\langle\text{ligne}\rangle$ . Parmi les options, *l*, *c* et *r* sont utilisées pour l'alignement du texte. Des formules commençant par le signe = peuvent être écrites et le texte peut être mis en forme en utilisant les commandes correspondantes.

Et enfin, les commandes  $\backslash\text{selecCell}$  et  $\backslash\text{multiSelec}$  permettent de mettre en couleur des cellules sélectionnées. Les exemples suivants vous montrent comment :

	A	D	T
1			

	A	B	C
1			
2			
3			

	A	B	C	D
1	$x$	$f(x)$		
2	0	$=A2*A2$		
3	1			

```

1 \begin{tikzpicture}
2   \tabcolwidth{1cm}
3   \tableur{A,D,T}
4 \end{tikzpicture}\bigskip
5
6 \begin{tikzpicture}
7   \tabcolwidth{2cm}
8   \tableur[3]{A,B,C}
9   \multiSelec{A-2}{B-3}
10 \end{tikzpicture}\bigskip
11
12 \begin{tikzpicture}
13   \tabcolwidth{1.5cm}
14   \tableur[3]{A-D}
15   \celttext[c]{A}{1}{\itshape x}
16   \celttext[r]{A}{2}{0}
17   \celttext[r]{A}{3}{1}
18   \celttext[c]{B}{1}{\itshape f(x)}
19   \celttext{B}{2}{=A2*A2}
20   \selecCell{B}{2}
21 \end{tikzpicture}

```