

CHAPITRE 7

Mathématique

Ce chapitre décrit tous les éléments de base qu'il vous faudra maîtriser pour utiliser des mathématiques dans un document. On va commencer avec les symboles et les constructions de base. Ensuite, on enchaînera avec l'insertion d'équations et de matrices. Enfin, on terminera en voyant comment faire de la mise en page comme des théorèmes par exemple.

7.1 Mode mathématique

Il y a trois modes mathématique différents. On peut insérer une formule dans le texte en utilisant `$... $` ou `\(... \)`. Pour insérer une formule séparée du texte et centrée, on peut utiliser `$$... $$` ou `\[... \]`. Enfin, pour insérer une équation numérotée, on va utiliser l'environnement `equation`.

Lorsque que l'on est en mode mathématique et qu'on souhaite repasser en mode texte, il suffit d'utiliser la commande `\text{rm}`. Dans ce mode, les espaces seront à nouveau pris en compte, contrairement à ce qui se passe en mode mathématique.

Soient x et y deux entiers. Leur somme vaut :

$$z = x + y$$

On a également la propriété suivante :

$$z^2 \leq x^2 + y^2 \tag{7.1}$$

**Code**

```
Soient  $x$  et  $y$  deux entiers. Leur somme vaut :
\[
z = x + y
\]
On a également la propriété suivante :
\begin{equation}
z^2 \leq x^2 + y^2
\end{equation}
```

7.2 Symboles

Cette section rassemble les symboles les plus courants dont vous pourriez avoir besoin. La plupart sont disponibles directement en \LaTeX , tandis que pour d'autres, il faudra utiliser \mathcal{AMS} et donc importer les packages `amsmath` et `amssymb`.

7.2.1 Lettres grecques

Les lettres grecques minuscules sont obtenues avec des commandes, sauf pour la lettre *omicron* qui est obtenue avec le *o* minuscule romain.

α	<code>\alpha</code>	η	<code>\eta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>
β	<code>\beta</code>	θ	<code>\theta</code>	ξ	<code>\xi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	o	<code>o</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	π	<code>\pi</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>

Les lettres grecques majuscules sont obtenues soit avec une lettre majuscule romaine, soit avec une commande.

A	<code>A</code>	H	<code>H</code>	N	<code>N</code>	T	<code>T</code>
B	<code>B</code>	Θ	<code>\Theta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>
Γ	<code>\Gamma</code>	I	<code>I</code>	O	<code>O</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	K	<code>K</code>	Π	<code>\Pi</code>	X	<code>X</code>
E	<code>E</code>	Λ	<code>\Lambda</code>	P	<code>P</code>	Ψ	<code>\Psi</code>
Z	<code>Z</code>	M	<code>M</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>

Enfin, certaines lettres minuscules possèdent une variante. Pour la lettre *kappa*, il faut importer le package `amssymb`.

ε	<code>\varepsilon</code>	\varkappa	<code>\varkappa</code>	ϱ	<code>\varrho</code>	φ	<code>\varphi</code>
ϑ	<code>\vartheta</code>	ϖ	<code>\varpi</code>	ς	<code>\varsigma</code>		

7.2.2 Symboles divers

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\exists	<code>\exists</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\neg	<code>\neg</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\natural	<code>\natural</code>
\wp	<code>\wp</code>	\bot	<code>\bot</code>	\sharp	<code>\sharp</code>	\sharp	<code>\sharp</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\clubsuit	<code>\clubsuit</code>	\clubsuit	<code>\clubsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	\diamondsuit	<code>\diamondsuit</code>	\diamondsuit	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>	\spadesuit	<code>\spadesuit</code>
$<$	<code><</code>	$>$	<code>></code>	\gg	<code>\gg</code>	\gg	<code>\gg</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\ll	<code>\ll</code>	\ll	<code>\ll</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	$=$	<code>=</code>	$=$	<code>=</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\neq	<code>\neq</code>	\neq	<code>\neq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\equiv	<code>\equiv</code>	\equiv	<code>\equiv</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\approx	<code>\approx</code>	\approx	<code>\approx</code>
\sqsubset^*	<code>\sqsubset^*</code>	\sqsupset^*	<code>\sqsupset^*</code>	\cong	<code>\cong</code>	\cong	<code>\cong</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\propto	<code>\propto</code>	\propto	<code>\propto</code>
\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>	\bowtie	<code>\bowtie</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>	\notin	<code>\notin</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\sim	<code>\sim</code>	\sim	<code>\sim</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>	\asymp	<code>\asymp</code>
$+$	<code>+</code>	$-$	<code>-</code>	\simeq	<code>\simeq</code>	\simeq	<code>\simeq</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\circ	<code>\circ</code>	\circ	<code>\circ</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\bullet	<code>\bullet</code>	\bullet	<code>\bullet</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\oplus	<code>\oplus</code>	\oplus	<code>\oplus</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>	\ominus	<code>\ominus</code>
\otimes	<code>\otimes</code>	\oslash	<code>\oslash</code>	\odot	<code>\odot</code>	\odot	<code>\odot</code>
\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>	\Rightarrow	<code>\Rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	$*$	<code>*</code>	$*$	<code>*</code>
\diamond	<code>\diamond</code>	\wr	<code>\wr</code>	\star	<code>\star</code>	\star	<code>\star</code>
\lessdot^*	<code>\lessdot^*</code>	\gtrdot^*	<code>\gtrdot^*</code>	\doteq	<code>\doteq</code>	\doteq	<code>\doteq</code>
\bigtriangleup	<code>\bigtriangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\bigcirc	<code>\bigcirc</code>	\bigcirc	<code>\bigcirc</code>
\uplus	<code>\uplus</code>	\amalg	<code>\amalg</code>	\Join^*	<code>\Join^*</code>	\Join^*	<code>\Join^*</code>
\lhd	<code>\lhd</code>	\rhd	<code>\rhd</code>				
\unlhd	<code>\unlhd</code>	\unrhd	<code>\unrhd</code>				

7.2.5 Polices de caractères

On peut obtenir différentes polices de caractères en mode mathématique. Les lettres représentant les ensembles sont obtenues avec la commande `\mathbb` :

\mathbb{A} `\mathbb{A}` \mathbb{B} `\mathbb{B}` \mathbb{C} `\mathbb{C}` ...

On peut également obtenir des lettres rondes (calligraphiées) avec la commande `\mathcal` :

\mathcal{A} `\mathcal{A}` \mathcal{B} `\mathcal{B}` \mathcal{C} `\mathcal{C}` ...

Enfin, on peut également avoir une police de type gothique avec la commande `\mathfrak` :

\mathfrak{A} `\mathfrak{A}` \mathfrak{B} `\mathfrak{B}` \mathfrak{C} `\mathfrak{C}` ...

7.3 Constructions basiques

On va voir toutes les constructions mathématiques basiques dans cette section : fraction, racine carrée, fonctions, indice et exposant ...

Indice	x_{10}	<code>x_{10}</code>
Exposant	x^{10}	<code>x^{10}</code>
Fraction	$\frac{x}{y}$	<code>\frac{x}{y}</code>
Racine nième	$\sqrt[n]{x}$	<code>\sqrt[n]{x}</code>
Trait horizontal	\overline{AB}	<code>\overline{AB}</code>
	\underline{AB}	<code>\underline{AB}</code>
Vecteur	\overrightarrow{AB}	<code>\overrightarrow{AB}</code>
Angle	\widehat{AB}	<code>\widehat{AB}</code>
Tilde	\widetilde{AB}	<code>\widetilde{AB}</code>

7.3.1 Accolade

On peut avoir une accolade qui s'étend en-dessous ou au-dessus d'une formule, tout en ajoutant un texte sur l'accolade avec les commandes `\overbrace` et `\underbrace`.

$$S = \overbrace{\pi}^{3,14} \cdot \underbrace{r}_{\text{rayon}}^2$$



Code

```
\[
  S = \overbrace{\pi}^{3,14} \cdot \underbrace{r}_{\text{rayon}}^2
\]
```

7.3.2 Opérateurs de taille variable avec indice et exposant

On peut insérer des opérateurs de taille variable, pouvant prendre une valeur en indice et en exposant. Voyons tout d'abord comment insérer une somme :

$$\sum_{i=0}^{n-1} x_i^2 + 2$$

On utilise pour cela la commande `\sum` :



Code

```
\[
  \sum_{i = 0}^{n - 1} {x_i^2 + 2}
\]
```

On peut également obtenir le même effet en utilisant les commandes `\limits` et `\nolimits` qui permettent de spécifier les valeurs en exposant et indice à placer sur le symbole obtenu avec la commande qui précède.

$$\int_0^2 x^2 - 1 \, dx \quad \int\limits_0^2 x^2 - 1 \, dx \quad \int\nolimits_0^2 x^2 - 1 \, dx$$

Vous remarquerez qu'avec `\limits`, les valeurs sont positionnées au-dessus et en-dessous du symbole alors qu'avec `\nolimits`, ces dernières sont placées à côté du symbole.

**Code**

```
\[
  \int_{0}^{2} {x^2 - 1} \, dx
  \qquad \int\limits_{0}^{2} {x^2 - 1} \, dx
  \qquad \int\limits_{0}^{2} {x^2 - 1} \, dx
\]
```

Voici les opérateurs les plus courants :

\prod	<code>\prod</code>	\coprod	<code>\coprod</code>	\sum	<code>\sum</code>	\int	<code>\int</code>
\bigcap	<code>\bigcap</code>	\bigcup	<code>\bigcup</code>	\bigsqcup	<code>\bigsqcup</code>	\oint	<code>\oint</code>
\bigwedge	<code>\bigwedge</code>	\bigvee	<code>\bigvee</code>	\biguplus	<code>\biguplus</code>	\iiint	<code>\iiint</code>
\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>	\bigodot	<code>\bigodot</code>		

7.3.3 Fonctions

Il y a toute une série de fonctions prédéfinies qui sont disponibles via des commandes. Celles-ci sont rendues dans une police de caractère droite, et suivies d'un espace. On peut par exemple écrire :

$$p = \Pr \left[|X| \leq \sin \frac{\pi}{4} \right]$$

**Code**

```
\[
  p = \Pr\left[ |X| \leq \sin \frac{\pi}{4} \right]
\]
```

Voici les fonctions prédéfinies :

<code>\arccos</code>	<code>\cot</code>	<code>\exp</code>	<code>\lim</code>	<code>\min</code>	<code>\tan</code>
<code>\arcsin</code>	<code>\coth</code>	<code>\gcd</code>	<code>\liminf</code>	<code>\Pr</code>	<code>\tanh</code>
<code>\arctan</code>	<code>\csc</code>	<code>\hom</code>	<code>\limsup</code>	<code>\sec</code>	
<code>\arg</code>	<code>\deg</code>	<code>\inf</code>	<code>\ln</code>	<code>\sin</code>	
<code>\cos</code>	<code>\det</code>	<code>\ker</code>	<code>\log</code>	<code>\sinh</code>	
<code>\cosh</code>	<code>\dim</code>	<code>\lg</code>	<code>\max</code>	<code>\sup</code>	

On peut utiliser ces fonctions avec `\limits` et `\nolimits` :

$$\lim_{x \rightarrow \infty} \frac{1}{x} \quad \max_{x \in [0,5]} x^2$$

**Code**

```
\[
  \lim\nolimits_{x \to \infty} \frac{1}{x}
\quad \max_{x \in [0,5]} x^2
\]
```

Il y a également la fonction `mod` utilisée pour l'arithmétique modulaire. On peut l'obtenir avec deux commandes différentes : `\bmod` et `\pmod`.

$$a \bmod b \quad x \equiv a \pmod{b}$$

**Code**

```
\[
  a \bmod b
\quad x \equiv a \pmod{b}
\]
```

Définir une fonction

Pour définir une nouvelle fonction, il faut utiliser la commande `\DeclareMathOperator` du package `amsmath`. Si on souhaite redéfinir une fonction qui existait déjà, on va utiliser la commande `\operatorname`. Ces définitions doivent se faire dans le préambule du document.

$$\operatorname{acos} x = \operatorname{Arc} \cos x$$

**Code****amsmath**

```
\DeclareMathOperator{\acos}{acos}
\renewcommand{\arccos}{\operatorname{Arc-cos}}

\begin{document}
$\operatorname{acos} x = \operatorname{arccos} x$
```

7.3.4 Superposer deux symboles

On peut superposer deux symboles avec la commande `\stackrel`. Celle-ci prend deux paramètres. Le premier est le symbole qui va au-dessus du second.

$$P(f) \stackrel{\triangle}{=} \{x \mid f(x) \geq 0\}$$


$$\backslash [\quad P(f) \quad \backslash \stackrel{\text{rel}}{\triangle} \{=\} \quad \backslash \{ \quad x \quad \backslash \text{mid} \quad f(x) \quad \backslash \geq 0 \quad \backslash]$$

Il existe toute une série de délimiteurs qu'on va pouvoir utiliser avec les commandes `\left` et `\right` afin qu'ils s'adaptent automatiquement à leur contenu. On peut par exemple écrire :

$$\left\langle \frac{1}{x^2 + 1}, \frac{2}{x^2} \right\rangle$$

Code

$$\left\langle \frac{1}{x^2 + 1}, \frac{2}{x^2} \right\rangle$$
$$\left(\left(\left(\left(\right)\right)\right)\right)$$

$$\backslash \Bigg(\backslash \bigg(\backslash \Big(\backslash \big(\backslash \big) \backslash \Big) \backslash \bigg) \backslash \Bigg)$$


Voici les délimiteurs les plus courants :

(())	↑	\uparrow	↗	\Uparrow
[[]]	↓	\downarrow	↘	\Downarrow
{	\{	}	\}	↕	\updownarrow	↕	\Updownarrow
<	\langle	>	\rangle				\
⌊	\lfloor	⌋	\rfloor	⌈	\lceil	⌋	\rceil
/	/	\	\backslash				

7.4.1 Définition directe

En utilisant le package `delarray`, on va pouvoir définir directement les délimiteurs. Il faut par contre faire très attention aux conflits avec d'autres packages, dont notamment `dcolum`, `colortbl`, `makecell` ou `tabularx`.

$$\begin{pmatrix} y & 3x^2 + x - 1 \\ 2y & -x + 3 \end{pmatrix}$$

 **Code** delarray

```

\begin{array}({cc})
  y & 3 x^2 + x - 1 \\
  2 y & -x + 3
\end{array}

```


7.5 Tableaux, matrices et équations

Dans cette section, on va voir comment définir des tableaux en mode mathématique. On verra également comment définir des matrices, des systèmes d'équations et des listes d'équations.

7.5.1 Tableaux

On peut définir des tableaux en mode mathématique en utilisant l'environnement `array`. Ce dernier fonctionne comme l'environnement `tabular` (section 5.1).

$$\begin{array}{lcl} 1 & = & 1 \\ 11 & = & 11 \\ 111 & = & 111 \end{array}$$


 **Code**

```
\[
  \begin{array}{rc1}
    1 & & = & & 1 \\
    11 & & = & & 11 \\
    111 & & = & & 111
  \end{array}
\]
```

7.5.2 Matrices et systèmes d'équations

On peut facilement construire des matrices et systèmes d'équations en utilisant l'environnement `array` et les délimiteurs. Pour une matrice, il suffit d'utiliser des parenthèses à gauche et à droite. Pour un système d'équations, on place une accolade à gauche et aucun délimiteur à droite.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{cases} a = b \\ c = d \end{cases}$$

 **Code**

```
\[
  \left(\begin{array}{cc}
    a & b \\
    c & d
  \end{array}\right)
  \quad
  \left\{\begin{array}{l}
    a = b \\
    c = d
  \end{array}\right.
\]
```

7.5.3 Listes d'équations

Parfois, on souhaite faire une liste d'équations successives, de la forme $Y = X$. Pour ce faire, on peut utiliser l'environnement `eqnarray`.

$$y = x^2 - 1 \tag{7.2}$$

$$2y = -x^2 + x + 1 \tag{7.3}$$

Cet environnement ne doit pas être utilisé en mode mathématique, c'est implicite. Il correspond en fait à l'environnement `array` avec l'alignement `rc1`. De plus, chaque équation est automatiquement numérotée.

**Code**

```
\begin{eqnarray}
y &= & x^2 - 1 \\
2y &= & -x^2 + x + 1 \\
\end{eqnarray}
```

Pour ne pas avoir de numérotation sur les équations, il faut plutôt utiliser l'environnement `eqnarray*`. On peut également supprimer les numéros de certaines des lignes en utilisant la commande `\nonumber`.

$$y = x^2 - 1 \quad (7.4)$$

$$2y = -x^2 + x + 1$$

$$y = -x + 4 \quad (7.5)$$

**Code**

```
\begin{eqnarray}
y &= & x^2 - 1 \\
2y &= & -x^2 + x + 1 \nonumber \\
y &= & -x + 4 \\
\end{eqnarray}
```

7.6 Théorèmes

On peut très facilement insérer des nouveaux théorèmes (ou des définitions, propositions, lemmes ...) en définissant des nouveaux environnements. Pour cela, il faut utiliser la commande `\newtheorem` au début du document. Celle-ci prend deux paramètres qui sont le nom de l'environnement et l'étiquette à afficher. En option, on peut spécifier le niveau de sectionnement pour la numérotation et on peut spécifier un compteur pour la numérotation. Voici un exemple d'un tel théorème :

Théorème 7.1 (Pythagore). *Le carré de la longueur de l'hypothénuse d'un triangle rectangle est égal à la somme des carrés des longueurs des autres côtés.*

On a défini ici un environnement `theorem` dont les numéros dépendront du chapitre. De plus, on a donné un nom au théorème via l'option de l'environnement. Enfin, la numérotation des propositions se fera avec le même compteur que celui utilisé pour les théorèmes.

**Code**

```
\newtheorem{theorem}{Théorème}[chapter]
\newtheorem{proposition}[theorem]{Proposition}

\begin{theorem}[Pythagore]
Le carré de la longueur de l'hypothénuse d'un triangle rectangle
est égal à la somme des carrés des longueurs des autres côtés.
\end{theorem}
```

7.6.1 Démonstration

On peut également insérer une démonstration en utilisant l'environnement `proof` disponible dans le package `amsthm`. Le symbole QED¹ est automatiquement ajouté grâce à cet environnement.

Démonstration. Soient x et y deux entiers tels que $x^2 + y^2 \geq 0$. De plus, soit l'entier positif z défini comme $\frac{x}{y}$. On peut écrire :

$$zy = x \tag{7.6}$$

□

Parfois, le symbole QED n'est pas placé comme il faut. C'est notamment le cas lorsque la preuve se termine par une équation. Dans ce cas, on peut utiliser la commande `\qedhere` pour indiquer où l'on souhaite le placer.

**Code**

```
\begin{proof}
Soient  $x$  et  $y$  deux entiers tels que  $x^2 + y^2 \geq 0$ . De
plus, soit l'entier positif  $z$  défini comme  $\frac{x}{y}$ .
On peut écrire :

\begin{equation}
zy = x \ \text{\texttt{\textbackslash qedhere}}
\end{equation}
\end{proof}
```

Notez qu'on peut également indiquer en option à l'environnement `proof` le nom du théorème qui est prouvé.

1. Quod erat demonstrandum, synonyme de CQFD.

