

Rapports techniques

Chapitre 02:

Algorithmes et mathématique avec LaTeX

Préparer par:

LOUNIS NAWAL

mathématique avec

LaTeX

mathématique avec LaTeX

Environnements de base

 **Les trois façons d'écrire des maths : LATEX**
distingue trois manières d'écrire des mathématiques.

- ▶ La première consiste à insérer une formule dans le texte .

- ▶ La deuxième est utilisé pour écrire des formules mathématiques centrées.

- La troisième Utilisé pour écrire des formules mathématiques centrées et numéroté.

Mathématique avec LaTeX

Environnements de base

👍 **Formules mathématiques, écrites dans une ligne de texte standard** : pour insérer une formule dans le texte on utilise l'environnement ***math***

👍 **Syntaxe :**

`\begin{math}équation\end{math}`

ou

`$équation$`

ou

`\(équation\)`

👍 **Exemple :**

On considère la fonction f définie pour tout réel x par $f(x) = 5x^4 + 4x^3 + 3x^2 + 2x + 1$.

On considère la fonction f définie pour tout réel x par $f(x)=5x^4+4x^3+3x^2+2x+1$.

Mathématique avec LaTeX

Environnements de base

✚ **Formules mathématiques centrées** : pour insérer des formules mathématique centrées on utilise l'environnement ***displaymath***

✚ **Syntaxe** :

`\begin{displaymath}équation\end{displaymath}`

&

ou

`$$équation$$`

ou

`\[équation\]`

✚ **Exemple** :

Déterminer la fonction dérivée
de $f(x)$:

```
\begin{displaymath}
  f(x)=\sqrt{\frac{x-1}{x+1}}
\end{displaymath}
si elle existe.
```

Déterminer la fonction dérivée
de $f(x)$:

$$f(x) = \sqrt{\frac{x-1}{x+1}}$$

si elle existe.

Mathématique avec LaTeX

Environnements de base

👍 **Formules mathématiques centrées et numéroté:** pour insérer des formules centrées et numéroté on utilise l'environnement ***equation***.

👍 ***syntaxe*** :

```
\begin{equation}équation\end{equation}
```

👍 **Exemple :**

À retenir : si $a > 0$ et $b > 0$,

```
\begin{equation}
```

```
\ln(ab)=\ln(a)+\ln(b)
```

```
\end{equation}
```

À retenir : si $a > 0$ et $b > 0$,

$$\ln(ab) = \ln(a) + \ln(b) \quad (3.1)$$

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 **Indices et exposants** : `_` et `^` sont les commandes permettant de produire respectivement *indice* et *exposant*. Il est nécessaire de « grouper » les arguments entre accolades pour que ces commandes agissent sur plusieurs symboles.

`_`{*indice*}
et
`^`{*exposant*}

👍 **Exemple :**

<code>x_2</code>	x_2	<code>x_{2y}</code>	x_{2y}	<code>x_{t_0}</code>	x_{t_0}
<code>x^2</code>	x^2	<code>x^{2y}</code>	x^{2y}	<code>x_{t^0}</code>	x_{t^0}
		<code>x^{2y}_{t_0}</code>	$x^{2y}_{t_0}$	<code>x_{t^1}^{2y}</code>	$x_{t^1}^{2y}$

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 **Fraction** : la commande `\frac{num}{denom}` produit une fraction formée par le **numérateur** num et le **dénominateur** denom.

👍 **Exemple** :

Soit a et b deux nombres non nuls. On appelle *moyenne harmonique* de a et b le nombre h défini par
$$\frac{1}{h} = \frac{\frac{1}{a} + \frac{1}{b}}{2}.$$



Soit a et b deux nombres non nuls. On appelle *moyenne harmonique* de a et b le nombre h défini par

$$\frac{1}{h} = \frac{\frac{1}{a} + \frac{1}{b}}{2}.$$

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 Racines :

- ▶ la commande `\sqrt{expr}` affiche la racine de son argument.
- ▶ la commande `\sqrt[n]{expr}` affiche la racine n de son argument.

▶ Exemple :

Soit a un réel positif. On appelle racine carrée de a le nombre positif noté `\sqrt{a}` tel que `(\sqrt{a})^2=a`.



Soit a un réel positif. On appelle racine carrée de a le nombre positif noté \sqrt{a} tel que $(\sqrt{a})^2 = a$.

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 Fonctions mathématiques :

<code>\arccos</code>	arccos	<code>\arcsin</code>	arcsin	<code>\arctan</code>	arctan	<code>\arg</code>	arg
<code>\cos</code>	cos	<code>\cosh</code>	cosh	<code>\exp</code>	exp	<code>\inf</code>	inf
<code>\lim</code>	lim	<code>\ln</code>	ln	<code>\log</code>	log	<code>\max</code>	max
<code>\min</code>	min	<code>\sin</code>	sin	<code>\sinh</code>	sinh	<code>\sup</code>	sup
<code>\tan</code>	tan						

👍 Exemples :

Pour tout réel x ,

$$\cos 2x = 2 \cos^2 x - 1.$$

Pour tout réel x ,

$$\cos 2x = 2 \cos^2 x - 1.$$

Pour tout réel x ,

$$\sin 2x = 2 \sin x \cos x.$$

Pour tout réel x ,

$$\sin 2x = 2 \sin x \cos x.$$

Mathématique avec LaTeX

Saisie de formules mathématiques

- 👍 **Flèches** : les commandes permettant de générer des flèches :
- toutes les commandes finissent par **arrow** ;
 - le préfixe obligatoire **left** ou **right** indique la direction ;
 - le préfixe facultatif long donne une version longue ;
 - la première lettre de la commande mise en majuscule rend la flèche double ;
 - on peut mettre des flèches aux deux extrémités en collant les deux mots left et right.

👍 **Exemples de commandes :**

<code>\rightarrow</code>	donne	\rightarrow		<code>\Longleftarrow</code>	donne	\Longleftarrow
<code>\Leftarrow</code>	donne	\Leftarrow		<code>\Longleftrightarrow</code>	donne	\Longleftrightarrow

👍 **Exemples:**

Mathématique avec LaTeX

Saisie de formules mathématiques

L'opérateur `\rightarrow` produit la flèche \rightarrow

On montre que $\lim_{h \rightarrow 0} \frac{\sin h}{h} = 1$ et

$$\lim_{h \rightarrow 0} \frac{\cos h - 1}{h} = 0.$$

On montre que

`\lim_{h \rightarrow 0}`

`\frac{\sin h}{h}=1` et

`\lim_{h \rightarrow 0}`

`\frac{\cos h-1}{h}=0.`

mathématique avec LaTeX

Saisie de formules mathématiques

👍 Opérateurs de somme et de produit :

<code>\sum</code>	\sum	<code>\prod</code>	\prod	<code>\int</code>	\int	<code>\bigcap</code>	\bigcap	<code>\bigcup</code>	\bigcup
-------------------	--------	--------------------	---------	-------------------	--------	----------------------	-----------	----------------------	-----------

👍 Exemple :

Pour tout réel q distinct de 1,

$$\sum_{k=0}^n q^k = \frac{1 - q^{n+1}}{1 - q}$$

Pour tout réel q distinct de 1,

$$\sum_{k=0}^n q^k = \frac{1 - q^{n+1}}{1 - q}$$

👍 Lettres grecques :

<code>\alpha</code>	α	<code>\beta</code>	β	<code>\gamma</code>	γ	<code>\delta</code>	δ	<code>\epsilon</code>	ϵ
<code>\varepsilon</code>	ε	<code>\theta</code>	θ	<code>\lambda</code>	λ	<code>\mu</code>	μ	<code>\pi</code>	π
<code>\rho</code>	ρ	<code>\sigma</code>	σ	<code>\phi</code>	ϕ	<code>\varphi</code>	φ	<code>\psi</code>	ψ
<code>\omega</code>	ω	<code>\Gamma</code>	Γ	<code>\Sigma</code>	Σ	<code>\Psi</code>	Ψ	<code>\Delta</code>	Δ
<code>\Omega</code>	Ω	<code>\Pi</code>	Π	<code>\Phi</code>	Φ				

mathématique avec LaTeX

Saisie de formules mathématiques

👍 Symboles et opérateurs :

<code>\infty</code>	∞	<code>\emptyset</code>	\emptyset	<code>\backslash</code>	\backslash	<code>\times</code>	\times	<code>\cdot</code>	\cdot
<code>\circ</code>	\circ	<code>\div</code>	\div	<code>\cap</code>	\cap	<code>\cup</code>	\cup	<code>\frown</code>	\frown

👍 Opérateurs binaires :

<code>\leq</code> ou <code>\le</code>	\leq	<code>\geq</code> ou <code>\ge</code>	\geq	<code>\equiv</code>	\equiv	<code>\sim</code>	\sim
<code>\simeq</code>	\simeq	<code>\approx</code>	\approx	<code>\perp</code>	\perp	<code>\subset</code>	\subset
<code>\in</code>	\in						
<code>\not\in</code>	\notin	<code>\not\subset</code>	$\not\subset$	<code>\not\equiv</code>	$\not\equiv$	<code>\not=</code> ou <code>\neq</code>	\neq

👍 Superposition de symboles:

`\stackrel{dessus}{\underset{dessous}{}}`

👍 Exemple :

$$f : x \mapsto f(x)$$

`$$f:x\stackrel{f}{\longmapsto}f(x)$$`

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 **Array** : L'environnement **array** permet de produire la grande majorité de vos formules. Comme son nom l'indique il range des objets en **ligne** et **colonne**. En fait c'est le pendant de l'environnement **tabular** du mode texte. Et comme tabular, array ne passe pas à la ligne.

👍 **Syntaxe** : La syntaxe rappelle celle de **tabular** :

```
\begin{array}[vpos]{format} ... \end{array}
```

- ▶ **format** : précise pour chaque colonne l'alignement : **c** pour centré, **l** pour aligné à gauche et **r** pour aligné à droite ;
- ▶ **vpos** : spécifie quant à lui le positionnement vertical du tableau.
- ▶ Comme dans les tableaux, on notera l'utilisation des commandes :
 - & comme séparateur de colonne ;
 - \\ pour passer à la ligne.

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 Exemples :

```
Soit $A=\begin{array}{rc}
-1 & 1 \\
3 & 4
\end{array}$ la matrice ...
```

3.14 Soit $A = \begin{pmatrix} -1 & 1 \\ 3 & 4 \end{pmatrix}$ la matrice ...

Exemple utilisant les points de suspensions :

```
\begin{displaymath} A=\left[\begin{array}{ccc}
a_{00} & \dots & a_{0n} \\
\vdots & \ddots & \vdots \\
a_{n0} & \dots & a_{nn}
\end{array}\right]
```

3.15 $A = \begin{bmatrix} a_{00} & \dots & a_{0n} \\ \vdots & \ddots & \vdots \\ a_{n0} & \dots & a_{nn} \end{bmatrix}$

- `\dots`, `\vdots` et `\ddots` produisent respectivement : \dots , \vdots et \ddots .

Mathématique avec LaTeX

Saisie de formules mathématiques

👍 **Array et les délimiteurs** : On utilise couramment l'environnement `array` pour produire des matrices. Il faut alors avoir recours à des *délimiteurs*. Ces délimiteurs sont de la famille des parenthèses et permettent d'englober un objet mathématique entre crochets, accolades, etc.

👍 **Syntaxe** : `\leftdelim1 objet \rightdelim2`

👍 `delim1` et `delim2` sont deux délimiteurs et `objet` un objet mathématique.

👍 Parmi les délimiteurs, voici les plus usités :

<code>(et)</code>	<code>(\Pi)</code>	<code>[et]</code>	<code>[\Pi]</code>
<code>\{ et \}</code>	<code>\{\Pi\}</code>	<code>\lfloor et \rfloor</code>	<code>\lfloor \Pi \rfloor</code>
<code>\lceil et \rceil</code>	<code>\lceil \Pi \rceil</code>	<code>\angle et \rangle</code>	<code>\angle \Pi \rangle</code>
<code> </code>	<code> \Pi </code>	<code>\ </code>	<code>\ \Pi\ </code>

Ecrire des Algorithmes

avec LaTeX

Insérer des Algorithmes avec LaTeX

- ✚ Quatre façons pour écrire des algorithmes avec LaTeX:
 - 👍 l'environnement **lstlisting** du package **listings**.
 - 👍 l'environnement **algorithm** du package **algorithm2e**.
 - 👍 l'environnement **algorithmic** du package **algorithmicx**.
 - 👍 l'environnement **program** du package **program**.

Insérer des Algorithmes avec LaTeX

✚ L'utilisation de l'environnement **algorithm** du package **algorithm2e**.

Instructions prédéfinies: De nombreuses instructions sont prédéfinies. Voici les plus courantes :

- Entrées/Sorties : `\KwData`, `\KwResult`, `\KwIn` et `\KwOut`
- Intervalle : `\KwTo`
- Renvois de valeur : `\KwRet` ou `\Return`
- Conditions : `\If`, `\ElseIf`, `\Else` et `\eIf`
- Choix : `\Switch`, `\Case` et `\Other`
- Boucles : `\For`, `\While`, `\ForEach`, `\ForAll` et `\Repeat`

Insérer des Algorithmes avec LaTeX

✚ L'environnement **algorithm** du package **algorithm2e**.

Exemples:

```
\begin{algorithm}
\caption{Un exemple d'algorithme.}
\KwIn{$N$ un entier positif}
\BlankLine
$sum$ \gets 0$ \;
\For{$x$ \gets 0$ \KwTo $N$}{
  \eIf{$x$ est pair}{
    $sum$ \gets $sum$ / 2$ \;
  }{
    $sum$ \gets $sum$ + 1$ \;
  }
}
\Return{$x$}
\end{algorithm}
```



Algorithme 1: Un exemple d'algorithme.

Input : N un entier positif

$sum \leftarrow 0$;

for $x \leftarrow 0$ to N do

 if x est pair then

$sum \leftarrow sum/2$;

 else


$sum \leftarrow sum + 1$;

 end

end

return x

Insérer des Algorithmes avec LaTeX

 **Style :** On peut modifier le style des algorithmes via les options du package lorsqu'on l'importe. Les options principales sont :

- `boxed` ou `boxruled` pour avoir un cadre autour de l'algorithme, avec la légende dedans ou pas ;
- `ruled` ou `algoruled` pour avoir des traits horizontaux avant et après l'algorithme, avec plus ou moins d'espace ;
- `lined`, `vlined` ou `noline` pour avoir des traits verticaux pour les instructions composées, ou rien du tout ;
- `linesnumbered` pour avoir les numéros de ligne ;
- `longend`, `shortend` ou `noend` pour des fins d'instructions longues ou courtes, ou aucune fin.

Insérer des Algorithmes avec LaTeX

✚ L'environnement **algorithm** du package **algorithm2e**.

Exemple (style):

```
\RestyleAlgo{boxed}

\begin{algorithm}
\ DontPrintSemicolon
\ SetAlgoVlined
\ LinesNumbered

\ KwIn{$n$ un entier positif}
\ KwOut{la valeur de la somme  $1 + 2 + \cdots + n$ }
\ BlankLine
$sum \gets 0$ \;
\ For{$i \gets 1$ \ KwTo $n$}{
    $sum \gets sum + i$ \;
}
\ Return{$sum$} \;
\ end{algorithm}
```

Input : n un entier positif

Output : la valeur de la somme $1 + 2 + \cdots + n$

```
1  $sum \leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $n$  do
3    $sum \leftarrow sum + i$ 
4 return  $sum$ 
```

l'environnement algorithm du package algorithm2e.

Syntaxe des commandes de base

- 👍 `\If{condition}{Then block} % in a block`
- 👍 `\elf{condition}{Then Block}{Else block} % in blocks`
- 👍 `\Switch{Condition}{Switch block}`
- 👍 `\Case{a case}{case block} % in a block`
- 👍 `\Other{otherwise block} % in a block`
- 👍 `\For{condition}{text loop} % in a block`
- 👍 `\While{condition}{text loop} % in a block`
- 👍 `\Repeat{End condition}{text loop} % in a block`
- 👍 pour plus d'information et des exemples
voir `_algorithm2e.sty`

 **`\listofalgorithms`**