

L'une des motivations majeures à la définition du langage *TeX* et au développement du logiciel correspondant par Donald Knuth était de faciliter la composition de formules mathématiques tout en garantissant une qualité professionnelle à l'impression. Sa réussite dans la réalisation de ces deux buts a été la cause du succès de *TeX*, et plus tard de *LaTeX*, auprès de la communauté scientifique. *TeX* est en effet le langage le plus utilisé pour la composition et la restitution de formules mathématiques complexes à l'écran.

Cependant, le langage reste difficile à acquérir en raison du grand nombre de symboles mathématiques existants. Le but de cet article est de permettre au lecteur d'acquérir les bases de LaTeX.

Les environnements mathématiques

LaTeX doit savoir à l'avance si le texte à traiter contient ou non des éléments mathématiques car il compose les notations mathématiques différemment du texte normal : la police change (bien que par défaut elle ressemble à celle du texte) ainsi que l'espacement (« **ab** » correspond à « a multiplié par b » et non à une syllabe d'un mot), ...

Des environnements spéciaux ont donc été définis à cette fin. Ils peuvent être répartis en deux catégories selon la façon dont ils sont présentés et peuvent apparaître

- dans le texte, au sein d'un paragraphe : les formules mathématiques sont affichées dans une ligne, c'est-à-dire dans le corps du texte là où elles sont écrites, par exemple dans la phrase « nous en déduisons que $a+a=2a$, après simplification. » ;
- de façon isolée (mode mathématique isolé, *displayed* en anglais), en dehors d'un paragraphe : les formules affichées sont séparées du texte principal, elles sont centrées sur la page ou la colonne (ou du moins, s'il y a plusieurs lignes, la ligne la plus grande est centrée).

Les équations hors paragraphe, isolées, peuvent être numérotées ; outre le fait que LaTeX numérote de manière automatique, si l'on place une étiquette `\label{...}` dans l'équation, la référence à cette étiquette (`\ref{...}`) donne le numéro de l'équation.

Il existe donc plusieurs environnements mathématiques, qui ont une forme « abrégée » :

Type	Environnement	Abréviation LaTeX	Abréviation TeX
dans le texte	<code>\begin{math}...\end{math}</code>	<code>\ (...\)</code>	<code>\$...\$</code>
mode hors paragraphe	<code>\begin{displaymath}...\end{displaymath}</code>	<code>\ [...\]</code>	<code>\$\$... \$\$</code>
mode hors paragraphe numéroté	<code>\begin{equation}...\end{equation}</code>	<i>aucune</i>	

Note

On voit dans de nombreux documents l'utilisation de `$$$...$$$`. Cette forme est à proscrire : elle donne un espacement incorrect et ne gère pas les options de classes [1] (<http://www.tuteurs.ens.fr/logiciels/latex/maths.html>). Par exemple, cette forme ne supporte pas les équations de nombre du côté gauche (classe `fleqn`).

Si l'on utilise l'extension **amsmath**, les modes hors paragraphe numéroté et non-numéroté sont équivalents, excepté la présence ou l'absence de numérotation. Si l'on n'utilise pas **amsmath**, les deux modes ne gèrent pas l'espacement vertical de la même manière.

Donnons tout de suite quelques exemples.

Le code `\sqrt{\frac{x^2}{3}}` va insérer la formule mathématique $\sqrt{\frac{x^2}{3}}$ dans un texte.

Autre exemple : `\[\sqrt{\frac{x^2}{3}}\]`. Cet exemple va afficher la même formule, sauf qu'elle va être centrée sur une nouvelle ligne.

```
\begin{document}
  Voici la fonction  $\sin x$ ,
  la fonction  $\cos x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}$ 
  et la fonction  $\tan(x)$  \ldots
\end{document}
```

va donner le résultat suivant :

Voici la fonction **$\sin x$** , la fonction

$$\cos x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}$$

et la fonction tangente **$\tan(x)$** ...

Autre différence avec la balise `\[...\]` par rapport à la balise `$...$` :

- `\sum_{k=0}^2` va donner

$$\sum_{k=0}^2$$

- alors que `\[\sum_{k=0}^2\]` va donner

$$\sum_{k=0}^2$$

Que faut-il écrire dans les balises `$...$` et `\[...\]` ?

Regardons premièrement, les puissances (exposants) et les indices des variables. Ceci est très simple, vous allez voir. Exemple, pour insérer x_i^2 , il suffit d'entrer le code LaTeX suivant : `$x_{i^2}`. Le symbole « \wedge » met le caractère suivant en exposant, alors que le caractère « $_$ » (tiret de soulignement) le met en indice.

Pour mettre un groupe de caractères ou toute une fonction en exposant (ou indice) il suffit d'utiliser les balises de groupes `{` et `}` (accolades). Par exemple, `x^{x^2}` va afficher x^{x^2}

Comment encadrer une équation ?

Pour encadrer une équation, on utilise l'extension `empheq`. Elle fournit l'environnement `empheq`, qui peut prendre plusieurs paramètres, en particulier :

- le type d'encadrement (paramètre optionel), avec `[box=...]` ; par défaut (paramètre absent), il n'y a pas d'encadrement ;
- le type d'environnement de base : `{equation}` pour une équation hors texte numérotée, `{equation*}` pour une équation non numérotée.

Par exemple :

```
\usepackage{empheq}
...
\begin{empheq}[box=\fbox]{equation*}
  \|\ \vec{\Cmath}_{3/1}\ \| = \unit{8\,670}{\newton}\text{.}
\end{empheq}
```

Les symboles

Les mathématiques utilisent beaucoup de symboles. S'il y a bien une difficulté en LaTeX c'est de se rappeler la façon dont les produire. Il y a bien sûr un ensemble de symboles directement accessibles à partir du clavier :

`+ - = ! / () [] < > | ' :`

En dehors de ceux-ci, différentes commandes doivent être appelées pour afficher les symboles désirés. Et il existe beaucoup de lettres grecques, de symboles relationnels et ensemblistes, de flèches, d'opérateurs binaires, etc. Un très grand nombre de symboles à apprendre, et en fait, ils surchargeraient ce cours d'instruction si j'essayais de les énumérer tous. Par conséquent, vous pouvez consulter un document complet de référence à [tutorial9/symbols.pdf symbols.pdf]. Nous verrons naturellement certains de ces symboles utilisés dans tout ce cours.

Multiplication

Plusieurs symboles sont habituellement utilisés pour représenter les produits.

Il est possible de placer un point entre les deux nombres comme dans l'exemple : `$x \cdot y$` ce qui donne $x \cdot y$. Notez bien que la multiplication utilise le point centré (`\cdot`) et non pas un point normal.

La multiplication peut aussi être représentée par le symbole « `\times` » qui signifie « fois ». Par exemple, `$x \times y$` donne le résultat suivant : $x \times y$.

Fractions

Pour créer une fraction, vous devez utiliser la commande `\frac{num\acute{e}rateur}{d\acute{e}nominateur}`. Ce qui donne :
$$\frac{\textit{numérateur}}{\textit{dénominateur}}$$

(Pour ceux qui ont besoin d'un moyen mnémotechnique, il suffit de retenir de *haut* en *bas* tout simplement.) Pour pouvez également imbriquer des fractions dans des fractions, comme le montrent les exemples ci-dessous :

$$\frac{x+y}{y-z}$$

$$\frac{\frac{1}{x} + \frac{1}{y}}{y-z}$$

Remarque : TeX fournissait une variante, la commande `\over`. Cependant, son usage est relativement malheureux car il place *tout* ce qui le précède au numérateur et *tout* ce qui le suit au dénominateur. Les cas où il peut être utile sont les fractions simples dans le texte, du type :

$$1+x \over 2 \frac{1+x}{2}$$

Puissances et indices

Les puissances et les indices sont les équivalents mathématiques des indices supérieurs et inférieurs dans un texte normal. Le caractère chapeau « ^ » est utilisé pour élever quelque chose, et celui de soulignement « _ » pour abaisser. Voyons un exemple qui montre comment les utiliser :

Puissance

$$x^n \quad x^n$$

$$x^{2n}$$

$$x^{2n}$$

Indice

$$n_i$$

$$n_i$$

$$n_{ij}$$

$$n_{ij}$$

Remarque : s'il y a plus d'un caractère à élever (ou à abaisser) alors vous devez les grouper en utilisant les accolades (`{...}`).

De plus, si vous avez besoin d'assigner en même temps, à la même entité, une puissance et un indice, alors cela peut être réalisé de la manière suivante :

$$x^{2i}_{3j}$$

ou

$$x_{3j}^{2i}$$

l'ordre n'ayant pas d'importance. Les deux écritures donneront x_{3j}^{2i}

Racines

Typiquement, et presque tout le temps, vous êtes amenés à utiliser la racine carrée, qui est obtenue facilement en utilisant la commande suivante : `\sqrt{x}`. Bien que `\sqrt` vienne de *square root* qui signifie « racine carrée », la commande peut être généralisée pour produire une racine de n'importe quelle ordre :

`\sqrt[n]{param}` $\sqrt[n]{x}$

LaTeX s'assure automatiquement que la taille du symbole racine s'ajuste à la taille du contenu.

Si le paramètre n est omis, LaTeX produira une racine carrée.

Parenthèses et autres délimiteurs

Les différents délimiteurs

L'utilisation des parenthèses devient très rapidement importante même dans les relations les plus insignifiantes. Sans les parenthèses, les formules peuvent devenir ambiguës. En outre, les types particuliers de structures mathématiques, telles que les matrices, se fondent typiquement sur des parenthèses pour les englober.

Vous pouvez vous rappeler que vous avez déjà les symboles du `()` et `[]` à votre disposition, qui devraient être adaptés aux besoins de la plupart des rédacteurs.

Il existe en fait beaucoup plus de symboles possibles qui peuvent être employés à la place des crochets, mais sont quelque peu rares. Consultez par exemple le tableau 5 de la référence de symboles ([tutorial9/symbols.pdf symbols.pdf]) pour les autres caractères. En voici quelques-uns :

- `()` (parenthèses simples)
- `[]` (crochets)
- `|` (barre verticale)
- `\|` (barre verticale double)
- `\{ \}` (accolade ouvrante et fermante).
- `.` (délimiteur nul, voir *Remarque* ci-dessous)

Grands délimiteurs

Considérons l'exemple suivant :

`\[(\frac{x^2}{y^3}) \]` $(\frac{x^2}{y^3})$

`\[\left(\frac{x^2}{y^3}\right) \]` $\left(\frac{x^2}{y^3}\right)$

Le premier exemple montre ce qui se produirait si vous employiez les caractères standards de parenthèses. Comme vous pouvez le voir, ils conviendraient à une simple équation qui tiendrait sur une seule ligne (par exemple, $(3 + 2) \times (10 - 3) = 35$) mais pas à des équations ayant une plus grande taille verticale, comme celles comportant des fractions. Le deuxième exemple illustre la manière dont LaTeX fait face à ce problème.

Les commandes `\left...` et `\right...` permettent d'adapter automatiquement la taille des parenthèses. Vous devez faire précéder les deux délimiteurs de ces commandes. Les points après la commande devront être remplacés par un des caractères selon le modèle de parenthèse voulu.

Une autre solution est d'utiliser le paquet *nath*, qui fournit des délimiteurs s'adaptant automatiquement à la taille.

Attention, en LaTeX, `{` et `}` sont interprétés : écrire `{x+1}` produit le résultat $x + 1$: pas d'accolades. Pour les caractères `{` et `}`, il faut utiliser `\{` et `\}`, par exemple `\{x+1\}` produit le résultat $\{x + 1\}$.

Remarque : pour utiliser un délimiteur d'un seul côté (par exemple mettre une accolade devant deux ou trois équations empilées), il faut utiliser le *délimiteur nul* qu'est le point : `\left.` `\right..` Par exemple :

```
\[ \left\{ \begin{array}{rcl} x+2y & = & -1 \\ -x+4y & = & 0 \end{array} \right.
```

(voir la section [Matrices](#) pour plus d'explications sur l'environnement `array`)

Espacement horizontal

LaTeX modifie l'espacement horizontal selon le contexte. La plupart des délimiteurs sont reconnus comme tels : l'espacement à l'intérieur d'une paire ouvrante-fermante est différent de l'espacement à l'extérieur. Cependant, cela n'est pas toujours le cas :

- LaTeX ne sait pas déterminer si les délimiteurs `|` et `\|` sont ouvrant ou fermants (puisque'ils sont identiques à gauche et à droite) ;
- l'adjonction des commandes `\left` et `\right` modifie cet espacement.

Pour éviter cela, on peut utiliser les commandes `\mathopen{}` en ouverture et `\mathclose{}` en fermeture. Par exemple :

```
\documentclass[10pt]{article}

\usepackage{amsmath}

\begin{document}

\begin{align}
y &= \sin |x| \\
y &= \sin \mathopen{|} x \mathclose{|}
```

$$y = \sin |x| \quad (1)$$

$$y = \sin |x| \quad (2)$$

$$x = \sin \left(\frac{1}{2} \right) \quad (3)$$

$$x = \sin \left(\frac{1}{2} \right) \quad (4)$$

Utilisation de `\mathopen{}` et `\mathclose{}`.

```

\\
x &= \sin \left ( \frac{1}{2} \right ) \\
x &= \sin \mathopen{}\left ( \frac{1}{2}
\right )\mathclose{}
\end{align}

\end{document}

```

À l'inverse, on peut lui demander de considérer un délimiteur comme un caractère « normal » avec la commande `\mathord{}`.

Si l'on a des problèmes d'espacement à l'intérieur d'une formule, on peut encapsuler la formule dans un `\mathinner{...}` : elle est alors considérée comme étant « à l'intérieur » (*inner*) de délimiteurs (même si ceux-ci sont absents). Par exemple, pour le produit de fractions :

```

\documentclass[10pt]{article}

\usepackage{amsmath}

\begin{document}

\begin{align}
x &= \frac{1}{2} \frac{3}{4} \\
x &= \mathinner{\frac{1}{2} \frac{3}{4}}
\end{align}

\end{document}

```

$$x = \frac{1}{2} \frac{3}{4} \quad (1)$$

$$x = \frac{1}{2} \frac{3}{4} \quad (2)$$

Utilisation de `\mathinner`.

Les fonctions mathématiques

Code LaTeX	Aperçu
<code>\sin</code>	sin
<code>\cos</code>	cos
<code>\tan</code>	tan
<code>\arcsin</code>	arcsin
<code>\arctan</code>	arctan
<code>\arccos</code>	arccos
<code>\exp</code>	exp

On peut créer soi-même des fonctions mathématiques avec la commande `\DeclareMathOperator{nom de la commande}{texte}` du paquet `amsmath` : le code

```

\DeclareMathOperator{\argsinh}{argsinh}

```

définit une commande `\argsinh` qui affiche le texte "argsinh" de la même façon que les commandes `\sin`, ...

Sommes

Le signe de sommation :

```
\[\exp(x)=\sum_{k=0}^{\infty}\frac{x^k}{k!}\]
```

donne

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

(`\infty` donne le signe infini).

Remarque : ceci est valable pour le mode "mathématiques affichées" (hors-texte et centrées). Dans le mode "mathématiques en ligne" (dans le texte, ouvert par `\(` ou `$` et fermé par `\)` ou `$`), les bornes n'apparaissent pas sous et sur le symbole, mais respectivement en indice et en exposant. Pour obtenir le bon résultat, il faut utiliser la commande `\sum\limits_{borne inf}^{borne sup}`. Le mot-clé `\limits` indique que l'indice doit être en réalité placé *dessous* et l'exposant *dessus*. Ce mot-clé fonctionne avec tous les grands opérateurs (`\sum`, `\prod` etc.) décrits ci-dessous.

Produits

Le signe de produit :

```
$n!=\prod_{k=1}^n k$
```

donne

$$n! = \prod_{k=1}^n k.$$

Intégrales

L'intégrale :

```
$\int_a^b f(x) \, , \, \mathrm{d}x$
```

donne

$$\int_a^b f(x) \, dx$$

(le code « `\,` » insère un espace fin et le code « `\mathrm` » permet d'obtenir un « d » romain).

Matrices

Latex n'a pas une commande particulière pour créer une matrice. Il dispose à la place d'un environnement un peu plus général, appelé tableau ou `array`. L'environnement `array` est fondamentalement équivalent à l'environnement `tabular` (voyez [le chapitre sur les tableaux](#) pour vous remémorer la syntaxe d'une table). Les rangées sont paramétrables, et peuvent être employées dans beaucoup de cas de figure, mais nous nous concentrerons sur les matrices. Vous pouvez employer les tableaux pour placer et aligner vos données comme vous voulez, et ensuite l'entourer avec des parenthèses gauches et droites appropriées pour obtenir votre matrice. Pour une matrice (2, 2) simple :

```
\[ \left(
  \begin{array}{c c }
    1 & 2 \\
    3 & 4
  \end{array} \right)
\]
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Voyons encore un autre exemple :

```

\begin{array}{l|cr}
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\
7 & 8 & 9
\end{array}

```

va donner

1	2	3
4	5	6
7	8	9

Après la déclaration de l'environnement, il faut insérer le type et le nombre de colonnes. Ici, `{l|cr}` :

- `l` : les éléments de la colonne sont alignés à gauche (comme `align=left` en HTML) ;
- `c` : les éléments de la colonne sont centré sur la colonne (comme `align=center` en HTML) ;
- `r` : les éléments de la colonne sont alignés à droite (comme `align=right` en HTML) ;
- `|` : insert un trait vertical entre deux colonnes.

À l'intérieur de l'environnement, on peut trouver les codes suivants :

- `\hline` : insert une ligne horizontale sur toute la largeur du tableau ;
- `&` : passe à la colonne suivante (comme `<td>` en HTML) ;
- `\\` : passe à la ligne suivante (comme `</tr>` en HTML).

Pour une documentation plus approfondie sur les tableaux, [allez voir le chapitre sur les tableaux](#).

Si vous incluez beaucoup de mathématiques dans vos documents, alors probablement vous vous apercevrez rapidement que vous souhaitez avoir un peu plus de contrôle sur la présentation. Certaines possibilités rendent l'écriture des formules plus complexes — mais qui a dit que la composition des mathématiques était facile ?

Les caractères grecs

Je pense que vous avez compris le

Code LaTeX	Aperçu	Code LaTeX	Aperçu	Code LaTeX	Aperçu
<code>\alpha</code>	α	<code>\pi</code>	π	<code>\Pi</code>	Π
<code>\beta</code>	β	<code>\sigma</code>	σ	<code>\Sigma</code>	Σ
<code>\gamma</code>	γ	<code>\tau</code>	τ	<code>\Tau</code>	T
<code>\delta</code>	δ	<code>\omega</code>	ω	<code>\Omega</code>	Ω
<code>\eta</code>	η	<code>\phi</code>	ϕ	<code>\Phi</code>	Φ
<code>\theta</code>	θ	<code>\varphi</code>	φ		
<code>\vartheta</code>	ϑ				

principe. Je ne vais donc pas tous les énumérer (pour l'instant).

Ajouter du texte dans les formules

Je doute que ce soit tous les jours que vous aurez besoin d'inclure du texte dans une formule.

Cependant, il arrive parfois que ce soit nécessaire. Un simple collage d'un texte directement dans l'environnement mathématique ne vous donnera pas les résultats que vous attendez. Par exemple :

```
\[
50 pommes \times 100 pommes = beaucoup de pommes^2
\]
```

$$50\text{pommes} \times 100\text{pommes} = \text{beaucoupdepommes}^2$$

ou

```
\[
5 \acute{e}\l\acute{e}phants + 4 \acute{e}\l\acute{e}phants = 9
\acute{e}\l\acute{e}phants
\]
```

$$5\acute{e}\acute{e}\acute{p}\acute{h}\acute{a}\acute{n}\acute{t}\acute{s} + 4\acute{e}\acute{e}\acute{p}\acute{h}\acute{a}\acute{n}\acute{t}\acute{s} = 9\acute{e}\acute{e}\acute{p}\acute{h}\acute{a}\acute{n}\acute{t}\acute{s}$$

Il y a deux problèmes visibles. Premièrement, il n'y a aucun espace entre les nombres et le texte, ni d'espaces entre les mots entre eux. Deuxièmement, les mots n'apparaissent pas correctement; les lettres sont davantage espacées par rapport à la normale. Cela vient du fait que ce sont simplement des objets façonnés du mode mathématique qui n'attend pas ces mots. Tous les espaces que vous saisissez en mode mathématique sont ignorés et Latex espace les éléments selon ses propres règles. Il suppose que tous les caractères représentent des noms de variable. Pour montrer que chaque symbole est individuel, ils ne sont pas placés les uns à côté des autres aussi étroitement que dans du texte normal.

Il y a un certain nombre de manières selon lesquelles le texte peut être ajouté correctement. La manière typique est d'envelopper le texte avec la commande `\mbox{...}`. Cette commande n'a pas été présentée avant, cependant son travail est fondamentalement de créer une boîte juste assez large pour contenir le texte considéré. Le texte placé dans cette boîte ne peut pas être cassé sur plusieurs lignes. Voyons ce qui se produit quand le code de la formule ci-dessus est adapté :

$$50\text{pommes} \times 100\text{pommes} = \text{beaucoup de pommes}^2$$

Le texte a un meilleur aspect. Cependant, il n'y a aucun espace entre les nombres et les mots. Malheureusement, vous êtes contraints d'ajouter vous-mêmes ces derniers. Il y a de nombreuses façons d'ajouter des espaces entre les objets mathématiques, mais pour la simplicité, je trouve plus facile, au moins dans ce cas, d'insérer directement un caractère d'espace dans la boîte `\mbox` elle-même (juste avant le texte).

```
\[
  50 \mbox{ pommes} \times 100 \mbox{ pommes} =
  \mbox{beaucoup de pommes}^2
\]
```

$$50 \text{ pommes} \times 100 \text{ pommes} = \text{beaucoup de pommes}^2$$

Il est préférable d'utiliser la commande `\text` du package `amsmath` ; `\text` adapte la largeur du texte à la largeur du texte environnant :

- `$_\underbrace{x + \cdots + x}_{10 \text{ fois}}$` va donner

$$\underbrace{x + \cdots + x}_{10 \text{ fois}}$$

- alors que `$_\underbrace{x + \cdots + x}_{10 \text{ fois}}$` va donner

$$\underbrace{x + \cdots + x}_{10 \text{ fois}}$$

La mise en caractère romain dans les formules

Il est une convention d'écrire les variables en italique (cela se fait par défaut, ex. : x^2). On peut mettre du texte en romain. Par exemple,

```
$p_{\mathrm{ext}}$
```

va donner : p_{ext} .

Les fonctions s'écrivent également en romain ; les fonctions les plus courantes possèdent leur abréviation. Il ne faut pas utiliser `\textrm` en mode mathématiques.

Mise en forme du texte

Employer `\mbox` permet de résoudre correctement le problème de base. Cependant, il y a une alternative qui offre un peu plus de flexibilité. Vous pouvez relire [le chapitre sur la mise en forme du texte](#), l'introduction aux commandes de modification de forme de police, comme `\textrm`, `\textit`, `\textbf`, etc. Ces commandes écrivent leur paramètre en utilisant une certaine forme de police, et par exemple `\textbf{texte en caractères gras}` donne **texte en caractères gras**. Ces commandes sont également valables dans un environnement mathématique et permettent d'inclure du texte. L'avantage supplémentaire ici est que vous pouvez avoir un meilleur contrôle de la forme de la police de caractères, plus qu'avec du texte normal inséré avec `\mbox`.

```
\begin{equation}
  50 \textrm{ pommes}
\times 100 \textbf{ pommes} =
  \textit{beaucoup de
pommes}^2
\end{equation}
```

$$50 \text{ pommes} \times 100 \textbf{ pommes} = \textit{beaucoup de pommes}^2 \quad (1)$$

Cependant, comme cela est le cas pour LaTeX, il y a plus d'une façon de tondre un chat ! Il existe des commandes de modification de forme de police très semblables à celles utilisées précédemment, sauf qu'elles sont précisément conçues pour écrire du texte en mode mathématique. Alors pourquoi s'embêter à vous montrer comment utiliser `\textrm` et les autres si on peut trouver les équivalents en mode maths ? Bien, c'est parce qu'elles sont subtilement différentes. Les commandes de changement de forme de police en mode mathématique sont :

Commande	Forme de police	Exemple
<code>\mathrm{...}</code>	Roman	$e = \mathbf{mc}^2$
<code>\mathit{...}</code>	Italic	$e = \mathit{mc}^2$
<code>\mathbf{...}</code>	En gras	$e = \mathbf{mc}^2$
<code>\mathsf{...}</code>	Sans serif	$e = \mathsf{mc}^2$
<code>\mathtt{...}</code>	Typewriter	$e = \mathtt{mc}^2$
<code>\mathcal{...}</code>	Calligraphie	$e = \mathcal{mc}^2$

Les commandes de changement de forme de police mathématique peuvent s'appliquer à une formule entière, et pas simplement un seul élément textuel : elles mettent en forme uniquement les lettres, les nombres, et les lettres majuscules grecques, mais le reste des objets mathématiques est ignoré. Ainsi, généralement, il vaut mieux employer les commandes spécifiques de maths s'il y a lieu. Notez que l'exemple de calligraphie donne un rendu plutôt étrange. C'est parce que pour les lettres, il exige des caractères majuscules. Les lettres restantes sont transformées en symboles spéciaux.

Changement de la taille des formules

Probablement un événement rare, mais il pourrait arriver que vous préféreriez disposer d'une commande pour changer la taille des caractères. Par exemple, en utilisant le mode texte en mode mathématique, par défaut une simple fraction ressemblera à ceci $\frac{1}{2}$ et vous aimeriez certainement la voir plus grande, comme dans le mode mathématique isolé par exemple, comme ceci : $\frac{1}{2}$. Une solution simple est d'utiliser les tailles prédéfinies pour les objets mathématiques :

<code>\displaystyle</code>	Taille pour les formules en mode mathématique isolé
<code>\textstyle</code>	Taille pour les formules en mode texte
<code>\scriptstyle</code>	Taille pour les indices supérieurs/inférieurs
<code>\scriptscriptstyle</code>	Taille pour les indices supérieurs/inférieurs des indices

Un exemple classique pour illustrer cela est celui des fractions continues. Examinons le code suivant :

```
\begin{equation}
x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}
\end{equation}
```

Comme vous pouvez le voir, les fractions continues deviennent plus petites au fur et à mesure (bien qu'elles ne le deviennent pas indéfiniment dans l'exemple), et atteignent la taille limite de `\scriptscriptstyle`. Si vous voulez garder une taille acceptable, vous pouvez définir chaque fraction en utilisant le style mathématique isolé à la place.

```
\begin{equation}
x = a_0 + \frac{1}{\displaystyle a_1
+ \frac{1}{\displaystyle a_2
+ \frac{1}{\displaystyle a_3 + a_4}}}
\end{equation}
```

Une autre solution est d'employer la commande `\DeclareMathSizes` permettant de sélectionner vos tailles préférées. Vous pouvez seulement définir des tailles pour les style `\displaystyle`, `\textstyle`, etc. Un inconvénient de cette commande est que les tailles des caractères en mode mathématiques ne peuvent être fixées que globalement, car elle ne peut être utilisée que dans le préambule du document.

Cependant, il est assez facile d'employer : `\DeclareMathSizes{ds}{ts}{ss}{sss}`, où *ds* représente la taille d'affichage (*display size*), *ts* est la taille du texte (*text size*), etc. Les valeurs que vous donnez pour définir la taille sont supposées représenter un nombre de points (pt). Dans le [tutorial10/mathsiz2.pdf exemple document (mathsiz2.pdf)], les tailles des maths ont été rendues beaucoup plus grandes que nécessaire afin de mettre en évidence les changements intervenus. Note : les changements ont lieu uniquement si la valeur du premier paramètre correspond exactement à la taille courante des textes de document. Il est donc fréquent de voir un ensemble de déclarations communes dans le préambule, lorsque la police principale est changée. Par exemple,

```
\DeclareMathSizes{10}{18}{12}{8}    % Pour un texte de taille 10
\DeclareMathSizes{11}{19}{13}{9}    % Pour un texte de taille 11
\DeclareMathSizes{12}{20}{14}{10}   % Pour un texte de taille 12
```

Équations multi-lignes (environnement `eqnarray`)

Imaginez que vous ayez une équation que vous voulez manipuler, par exemple, pour la simplifier. Souvent, cela s'effectue après un certain nombre d'étapes en aidant le lecteur à comprendre comment obtenir à partir de l'équation originale le résultat final. Cela devrait être une tâche relativement simple, mais nous verrons que les commandes utilisées dans le cours précédent pour écrire des mathématiques ne sont pas adaptées. En utilisant ce que nous savons jusqu'ici nous avons : Le résultat obtenu est plutôt laid. Le moyen d'afficher les divers éléments en les alignant est d'utiliser une table, et d'y placer les équations en ligne. Allons y :

```
\begin{tabular}{r l }
```

```
\(10xy^2+15x^2y-5xy\) & \ (= 5\left(2xy^2+3x^2y-xy\right)\) \\
& \ (= 5x\left(2y^2+3xy-y\right)\) \\
& \ (= 5xy\left(2y+3x-1\right)\)
\end{tabular}
```

Cela ne donne pas un résultat trop mauvais. Peut-être que l'espace supplémentaire entre le membre gauche de l'équation et le signe égal est plus grand que l'espace du côté droit et ne donne pas une apparence parfaite. Cela peut être rectifié en ajoutant une colonne supplémentaire et en mettant le signe égal dans la colonne centrale :

```
\begin{tabular}{ r c l }
\ (10xy^2+15x^2y-5xy\) & \ (=) & \ (5\left(2xy^2+3x^2y-xy\right)\) \\
\\
& \ (=) & \ (5x\left(2y^2+3xy-y\right)\) \\
& \ (=) & \ (5xy\left(2y+3x-1\right)\)
\end{tabular}
```

C'est beaucoup mieux. Un autre problème est que l'espace vertical entre les lignes fait apparaître les équations un peu trop serrées sur le côté droit. Il serait souhaitable d'ajouter un petit espace pour que les lignes dans la table soient un peu plus grandes. (vous devez ajouter `\usepackage{array}` dans votre préambule pour réaliser ce travail.)

% Augmente la taille verticalement. Nécessite le paquet tableau. Voir le préambule.

```
\setlength{\extrarowheight}{0.3cm}
```

Fichier:[Multiligne4.png](#)

Cependant, par ce moyen nous avons dû effectuer beaucoup trop de travail, et il reste un inconvénient important au bout du compte : vous ne pouvez pas numéroté les équations. Puisqu'une fois que vous êtes dans l'environnement `tabular`, vous pouvez seulement employer le type intégré d'affichage de maths. Pour la numérotation des équations, vous devez employer le paquet du mode d'affichage `equation`, mais vous ne pouvez pas dans cet exemple. C'est là que `eqnarray` devient extrêmement utile.

`eqnarray`, comme le nom le suggère provient du paquet `array` qui est fondamentalement un environnement simplifié de `tabular`. Le paquet `array` fut introduit dans le [latexutorial9.html tutorial 9 (Mathematics I)] pour former des matrices. Voici un exemple d'utilisation d'`eqnarray`:

```
\begin{eqnarray}
10xy^2+15x^2y-5xy & = & 5\left(2xy^2+3x^2y-xy\right) \\
& = & 5x\left(2y^2+3xy-y\right) \\
& = & 5xy\left(2y+3x-1\right)
\end{eqnarray}
```

Comme vous pouvez le voir, tout est bien présenté et organisé comme prévu. Bien, sauf que chaque rangée du tableau a reçu son propre numéro d'équation. Tandis que cette possibilité est utile dans certains cas, elle n'est pas requise ici, seulement un numéro suffira. Pour supprimer des numéros d'équation d'une rangée donnée, ajoutez la commande `\nonumber` juste avant la commande de fin de ligne (`\\`).

```
\begin{eqnarray}
  10xy^2+15x^2y-5xy & = & 5\left(2xy^2+3x^2y-xy\right) \nonumber \\
  & = & 5x\left(2y^2+3xy-y\right) \nonumber \\
  & = & 5xy\left(2y+3x-1\right) \\
\end{eqnarray}
```

Si vous ne voulez pas du tout de numéro d'équation, alors plutôt que d'ajouter la commande `\\nonumber` à chaque rangée, employez la version étoilée de l'environnement c'est-à-dire `\begin{eqnarray*}...\\end{eqnarray*}`.

Note : il y a une limite de 3 colonnes dans l'environnement de `eqnarray`. Si vous avez besoin de davantage de flexibilité, vous ferez mieux d'utiliser les paquets de Maths AMS et l'environnement `align`. Pour des raisons typographiques, l'environnement `align` doit être préféré à l'environnement de `eqnarray`.

Casser les longues équations

D'abord voyons un exemple de longue équation (note : lorsque vous visualiserez l'exemple de document de cette rubrique, vous verrez que cette équation est en fait plus large que la largeur du texte. Ce n'est pas si évident que cela sur cette page Web)

Latex ne casse pas les longues équations pour les faire entrer dans les marges comme il le fait avec le texte normal. Ainsi, il vous appartient de composer l'équation convenablement (si elle déborde la marge). Ceci exige typiquement une certaine utilisation créatrice d'`eqnarray` afin d'obtenir des objets découpés sur une nouvelle ligne et bien alignés. Par exemple,

```
\begin{eqnarray*}
  \left(1+x\right)^n & = & 1 + nx + \frac{n\left(n-1\right)}{2!}x^2 \\
  \\
  & + & \frac{n\left(n-1\right)\left(n-2\right)}{3!}x^3 \\
  & + & \frac{n\left(n-1\right)\left(n-2\right)\left(n-3\right)}{4!}x^4 \\
  & + & \ldots \\
\end{eqnarray*}
```

Vous pouvez noter qu'à partir de la deuxième ligne, l'espace entre le plus initial et la fraction suivante est légèrement plus petit que la normale. C'est dû au fait que Latex traite les signes `+` et `-` de deux façons différentes. Le plus souvent il les considère comme un opérateur binaire. Quand deux objets mathématiques apparaissent de part et d'autre du signe, Latex suppose qu'il est un opérateur binaire, et attribue un certain espace de chaque côté du signe. La manière alternative est une désignation de signe; lorsque vous énoncez

qu'une certaine quantité mathématique est positive ou négative. Comme en maths, on suppose que de tels objets sont positifs à moins qu'un signe le précède. Dans cet exemple, vous voulez que le signe apparaisse près de l'élément approprié pour montrer leur correspondance. C'est l'interprétation pour laquelle LaTeX a opté dans l'exemple ci-dessus. Pour ajouter la place nécessaire, vous pouvez insérer *un caractère invisible* en utilisant `{}`, comme nous le montrons ci-après :

```
\begin{eqnarray*}
\left(1+x\right)^n & = & 1 + nx + \frac{n\left(n-1\right)}{2!}x^2 \\
& & + \frac{n\left(n-1\right)\left(n-2\right)}{3!}x^3 \\
& & + \frac{n\left(n-1\right)\left(n-2\right)\left(n-3\right)}{4!}x^4 \\
& & + \ldots
\end{eqnarray*}
```

Une autre solution serait d'éviter ce problème en laissant le symbole `+` à l'extrémité de la ligne précédente plutôt que de le placer au début de la ligne courante :

Latex supporte une autre convention d'écriture des longues équations. C'est la manière qui est la plus fréquemment utilisée pour composer de longues équations dans les livres et les articles, et c'est évidemment ma manière préférée de les écrire. L'utilisation d'`eqnarray`, conjointement avec la commande `\lefteqn{...}` placée autour du contenu précédant le signe `=` donne le résultat suivant :

```
\begin{eqnarray*}
\lefteqn{\left(1+x\right)^n = } & & \\
& & 1 + nx + \frac{n\left(n-1\right)}{2!}x^2 + \\
& & \frac{n\left(n-1\right)\left(n-2\right)}{3!}x^3 + \\
& & \frac{n\left(n-1\right)\left(n-2\right)\left(n-3\right)}{4!}x^4 + \\
& & \ldots
\end{eqnarray*}
```

Notez que la première ligne du `eqnarray` contient uniquement `\lefteqn`. Et avec cette commande, il n'y a aucun séparateur de colonne (`&`). La raison pour laquelle cette commande affiche les objets comme il se doit est que la commande `\lefteqn` imprime l'argument, en indiquant à LaTeX que la largeur est nulle. Le contenu de la première colonne étant vide, à l'exception de l'espace inter-colonne, cela impose le retrait des lignes suivantes.

Contrôle de l'espacement horizontal

Latex est évidemment un outil très puissant pour la composition de textes mathématiques. La composition de textes scientifiques était l'un des principaux objectifs du noyau du système Tex que LaTeX prolonge. Cependant, il n'interprète pas toujours exactement les formules comme vous aimeriez qu'il le fasse. Il doit parfois choisir entre plusieurs options quand il rencontre une expression ambiguë. Le résultat a tendance à

être légèrement incorrect au niveau de l'espacement horizontal, mais reste malgré tout encore satisfaisant. Pourtant, tous les perfectionnistes voudront sans nul doute affiner leurs formules afin d'assurer un espacement horizontal impeccable. Ce sont généralement des ajustements très subtils.

Dans d'autres circonstances où LaTeX a fait son travail correctement, vous aimeriez juste ajouter un certain espace, pour éventuellement placer un commentaire. Par exemple, dans l'équation suivante, il est préférable de s'assurer qu'il y ait une place suffisante entre les maths et le texte.

```
\[f(n) = \left\{
\begin{array}{l l}
n/2 & \quad \text{si } n\$ \text{ est pair} \\
-(n+1)/2 & \quad \text{si } n\$ \text{ est} \\
\text{impair}} \end{array} \right. \]
```

$$f(n) = \begin{cases} n/2 & \text{si } n \text{ est pair} \\ -(n+1)/2 & \text{si } n \text{ est impair} \end{cases}$$

Latex a défini deux commandes qui peuvent être employées n'importe où dans les documents (pas seulement en mode mathématique) pour insérer un certain espace horizontal. Ce sont les commandes `\quad` et `\qquad`.

Un `\quad` abréviation de quadratin est une espace égale à la taille de la police courante. Ainsi, si vous employez une police de 11 points (11pt), alors l'espace fourni par `\quad` sera également de 11 points (horizontalement, naturellement). Le `\qquad` donne deux fois l'espace précédent. Comme vous pouvez le voir à partir du code de l'exemple ci-dessus, des `\quad` ont été employés pour ajouter une certaine séparation entre les maths et le texte.

Pour revenir à l'affinement mentionné au début du document, un bon exemple montrerait une simple intégrale indéfinie de y par rapport à x :

$$\int y dx$$

Si vous désirez essayer, vous pouvez écrire :

```
\[ \int y \mathrm{d}x \]
```

Cependant, ceci ne donne pas un résultat correct. LaTeX ne respecte pas l'espace blanc à gauche dans le code pour signifier que le y et le x sont des entités indépendantes. Au lieu de cela, il les colle l'une à l'autre. Un `\quad` laisserait un trop grand espace dans cette situation; il n'y a besoin que de quelques petits espaces dans ce type d'exemple, et c'est pour cela que LaTeX fournit :

Commande	Description	Taille
<code>\,</code>	petit espace	3/18 d'un quadratin
<code>\:</code>	moyen espace	4/18 d'un quadratin
<code>\;</code>	grand espace	5/18 d'un quadratin
<code>\!</code>	espace négatif	-3/18 d'un quadratin

NB vous pouvez utiliser plusieurs commandes à la suite pour laisser un plus grand espace si nécessaire.


```
\begin{align}
x &= a + (b + a) \\
&= 2a + b.
\end{align}
```

Le symbole esperluette (&) est utilisé pour préciser les points dans chaque ligne qui seront supposés être alignés verticalement, mais n'apparaissent pas dans le résultat final :

$$\begin{aligned} x &= a + (b + a) \\ &= 2a + b. \end{aligned} \quad \begin{matrix} (1) \\ (2) \end{matrix}$$

Les équations sont numérotées par défaut. Les numéros d'équation peut être supprimés en remplaçant le mot `align` par le mot `align*` au début et à la fin des équations à aligner :

```
\begin{align*}
x &= a + (b + a) \\
&= 2a + b.
\end{align*}
```

Le résultat ressemble à ceci :

$$\begin{aligned} x &= a + (b + a) \\ &= 2a + b. \end{aligned}$$

Une autre solution est de les supprimer sur une ligne particulière en ajoutant l'expression `\notag`, comme nous le montrons ici :

```
\begin{align}
x &= a + (b + a) \notag \\
&= 2a + b.
\end{align}
```

Le résultat ressemble à ceci :

$$\begin{aligned} x &= a + (b + a) \\ &= 2a + b. \end{aligned} \quad (1)$$

Des annotations dans chaque ligne peuvent être ajoutées en les séparant des équations, en utilisant deux esperluettes (&&) :

```
\begin{align}
x &= a + (b + a) && \text{(Axiome C)} \\
&= 2a + b && \text{(Axiomes A et F)}.
\end{align}
```

Voici le résultat :

$$\begin{aligned}
 x &= a + (b + a) && \text{(Axiome C)} \\
 &= 2a + b && \text{(Axiomes A et F)}.
 \end{aligned}$$

Commande multiline

Le paquet `amsmath` comporte également un environnement `multiline` (et sa version non-numérotée `multiline*`) qui permettent d'écrire une équation sur plusieurs lignes. Voici un exemple :

```

\begin{multiline}
  \exp(x) = 1 + x + x^2 / 2 + x^3/ 3! + \cdots \\\
           \cdots + x^n/ n! + \cdots
\end{multiline}

```

Si on souhaite forcer manuellement la justification d'une partie de la formule, on peut utiliser les commandes `\shoveright` et `\shoveleft`. Exemple :

```

\begin{multiline*}
  X = a + b + c + d + e + f + g + h + i + j + k + l + m \\\
      \shoveright{+ n + o + p + q + r + s + t + u + v} \\\
      +w+x+y+z+1+2+3+4+691
\end{multiline*}

```

Résumé

Comme vous pouvez commencer à le voir, la composition de textes mathématiques peut parfois être embêtante. Cependant, parce que LaTeX fournit de nombreux contrôles, vous pouvez obtenir des documents mathématiques de qualité professionnelle avec relativement peu d'effort (une fois que vous vous êtes familiarisé avec LaTeX, bien sûr). Il serait possible d'aller plus loin sur le thème des mathématiques parce que cela semble illimité. Cependant, avec ces deux tutoriels, vous devriez être capable de vous en sortir honorablement.

Fichiers : [tutorial10/textineqn.tex textineqn.tex] | [tutorial10/textineqn.pdf textineqn.pdf] | [tutorial10/mathsize.tex mathsize.tex] | [tutorial10/mathsize.pdf mathsize.pdf] | [tutorial10/mathsize2.tex mathsize2.tex] | [tutorial10/mathsize2.pdf mathsize2.pdf] | [tutorial10/multiline.tex multiline.tex] | [tutorial10/multiline.pdf multiline.pdf] | [tutorial10/longeqns.tex longeqns.tex] | [tutorial10/longeqns.pdf longeqns.pdf] | [tutorial10/hspacing.tex hspacing.tex] | [tutorial10/hspacing.pdf hspacing.pdf]
 ressources utiles : [Guide d'utilisation du paquet `amsmath` \[pdf \(ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf\)\]](#)

Récupérée de « <https://fr.wikibooks.org/w/index.php?title=LaTeX/Mathématiques&oldid=642802> »

La dernière modification de cette page a été faite le 16 juin 2020 à 16:07.

Les textes sont disponibles sous licence Creative Commons attribution partage à l'identique ; d'autres termes peuvent s'appliquer.

Voyez les termes d'utilisation pour plus de détails.