

Ateliers L^AT_EX
Deuxième atelier:
Mathématiques (enfin!)
et
structure de documents

Yves Delhayé

19 janvier 2010

Ateliers L^AT_EX
Deuxième atelier:
Mathématiques
(enfin!)
et
structure de
documents

Yves Delhayé

Mathématiques
élémentaires
Différents modes
mathématiques
Opérations
élémentaires
Structures
indispensables

Symboles en vrac



Mathématiques élémentaires : le plan

Ateliers L^AT_EX
Deuxième atelier:
Mathématiques
(enfin!)
et
structure de
documents

Yves Delhaye

Mathématiques élémentaires

Différents modes mathématiques

Opérations élémentaires

Structures indispensables

Mathématiques
élémentaires

Différents modes
mathématiques

Opérations
élémentaires

Structures
indispensables

Symboles en vrac



Mathématiques élémentaires

Différents modes mathématiques

Opérations élémentaires

Structures indispensables

Mathématiques
élémentaires

Différents modes
mathématiques

Opérations
élémentaires

Structures
indispensables

Symboles en vrac

Symboles en vrac



On s'en doute : les modes mathématiques sont destinés à composer des formules mathématiques mais en fait, on peut utiliser le mode mathématique pour des constructions qui n'ont rien à voir avec les math. Comme il s'agit d'un chapitre d'introduction, nous ne verrons cependant que l'aspect purement mathématique de la chose !



Il existe deux modes mathématiques : le mode en texte et le mode hors texte.



Le mode en texte permet de construire une formule destinée à être incorporée dans le texte d'où une étendue verticale réduite au minimum comme dans $\frac{1}{2}x = \frac{x}{2}$.



Le mode hors texte qui permet de composer des formules centrées (par défaut) et qui utilisent toute la ligne rien que pour elles ce qui permet des espacements verticaux plus généreux. Voici la même formule en hors texte :

$$\frac{1}{2}x = \frac{x}{2}$$



L^AT_EX permet des constructions telles que $\frac{1}{2}x = \frac{x}{2}$,
c'est-à-dire composer une formule en texte comme elle
apparaît en hors texte mais, contrairement à ce que pensent
beaucoup de personnes, c'est une erreur de vouloir le faire.
L'interlignage est complètement détruit ce qui est une faute
typographique de débutant !



Délimiteurs des modes mathématiques

Comment tape-t-on la formule qui a servi d'exemple ? Tout d'abord, il faut indiquer à \LaTeX quand commence et se termine la formule. Le tableau 1 suivant résume les possibilités :

délimiteurs	mode	exemple
<code>\$... \$</code>	en texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\(... \)</code>	en texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\begin{math} ... \end{math}</code>	en texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\$\$... \$\$</code>	hors texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\[... \]</code>	hors texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\begin{displaymath} ... \end{displaymath}</code>	hors texte	$\frac{1}{2}x = \frac{x}{2}$
<code>\begin{equation} ... \end{equation}</code>	+ n^0	$\frac{1}{2}x = \frac{x}{2}$



Pour un \$ de plus !

Les délimiteurs utilisant le signe dollar sont un codage \TeX . Il est préférable d'utiliser un des deux codages purement \LaTeX pour éviter certains effets bizarres qui peuvent survenir dans des situations complexes. J'avoue que faisant la navette entre \TeX et \LaTeX , j'utilise souvent les dollars mais ce n'est pas bien !



`\begin{equation}... \end{equation}`

`\begin{equation} ... \end{equation}` permet d'avoir des équations numérotées en mode hors texte. Cet environnement est très pratique si nous désirons référencer des équations dans un texte.

Si $n \rightarrow +\infty$, alors

$$\sum_{i=1}^n \left(\frac{1}{2}\right)^i = 2 \quad (1)$$

$$\sum_{i=1}^n \frac{1}{i} = +\infty \quad (2)$$

$$\sum_{i=1}^n (-1)^i \frac{1}{i} = \dots \quad (3)$$

1 Si

`\(n \rightarrow + \infty\)`, alors

2 `\begin{equation}`

3 `\sum_{i=1}^n \left(\frac{1}{2}\right)^i`

4 `\end{equation}`

5

6 `\begin{equation}`

7 `\sum_{i=1}^n \frac{1}{i} = + \infty`

8 `\end{equation}`

9

10 `\begin{equation}`

11 `\sum_{i=1}^n (-1)^i \frac{1}{i} =`

12 `\end{equation}`

Maintenant que nous savons nous mettre en mode mathématique, il reste à voir ce qu'on peut mettre dedans !
Durant cette première approche, nous ne verrons que les bases, les compositions plus complexes seront traitées plus tard.



Maintenant que nous savons nous mettre en mode mathématique, il reste à voir ce qu'on peut mettre dedans ! Durant cette première approche, nous ne verrons que les bases, les compositions plus complexes seront traitées plus tard.

Dans cette section, nous allons voir un grand nombre de macros permettant des constructions diverses. À part une seule, ces macros sont interdites en dehors des modes mathématiques : leur utilisation dans du texte provoquera une erreur lors de la compilation.



Les quatre opérations

$\text{T}_{\text{E}}\text{X}$ gère évidemment les lettres et les quatre opérations de base. Examinons la formule $x + 3 = y - 5$ obtenue avec le code `\(x+3=y-5\)`.

Ateliers $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
Deuxième atelier:
Mathématiques
(enfin!)
et
structure de
documents

Yves Delhaye

Mathématiques
élémentaires

Différents modes
mathématiques

**Opérations
élémentaires**

Structures
indispensables

Symboles en vrac



Les quatre opérations

$\text{T}_{\text{E}}\text{X}$ gère évidemment les lettres et les quatre opérations de base. Examinons la formule $x + 3 = y - 5$ obtenue avec le code `\(x+3=y-5\)`. On peut remarquer que des espaces de tailles variées ont été automatiquement ajoutées autour des signes plus, moins et égal. En fait, des espaces mis dans le source n'ont strictement aucune action en mode mathématique (hormis, bien sur, celui de stopper un nom de macro).



Les quatre opérations

\TeX gère évidemment les lettres et les quatre opérations de base. Examinons la formule $x + 3 = y - 5$ obtenue avec le code `\(x+3=y-5\)`. On peut remarquer que des espaces de tailles variées ont été automatiquement ajoutés autour des signes plus, moins et égal. En fait, des espaces mis dans le source n'ont strictement aucune action en mode mathématique (hormis, bien sur, celui de stopper un nom de macro). Par exemple `\(x + 3 = y - 5\)` aurait donné strictement le même résultat.



Les quatre opérations

\TeX gère évidemment les lettres et les quatre opérations de base. Examinons la formule $x + 3 = y - 5$ obtenue avec le code `\(x+3=y-5\)`. On peut remarquer que des espaces de tailles variées ont été automatiquement ajoutés autour des signes plus, moins et égal. En fait, des espaces mis dans le source n'ont strictement aucune action en mode mathématique (hormis, bien sur, celui de stopper un nom de macro). Par exemple `\(x + 3 = y - 5\)` aurait donné strictement le même résultat. On remarquera également que *les lettres sont en italique*; c'est la règle pour les formules de math sauf qu'en France, la règle complète indique que les majuscules ne doivent pas être en italique ce que ne fait pas \TeX par défaut (dans ce manuel, on transgressera cette règle typographique et même les majuscules seront en italique).



Les signes plus, moins et égal sont présents sur le clavier mais pas ceux de la multiplication et de la division (celle de l'école primaire). On peut bien évidemment taper un " / " pour la division mais la macro `\div` donne le symbole \div . Le signe multiplié est obtenu avec la macro `\times` qui donne le symbole \times .



T_EX est intelligent et la formule `x\times-3` est comprise comme il se doit, c'est-à-dire comme le produit de x et de -3 ce qui fait que les espaces sont correctement gérées : $x \times -3$. En définitive, si vous tapez votre formule correctement, T_EX la composera correctement d'un point de vue typographique.



Avec les variables et les quatre opérations, on ne fait quand même pas grand chose. Certaines structures reviennent très souvent dans les formules de mathématiques.



Les exposants et les indices utilisent deux caractères réservés de T_EX, à savoir respectivement \wedge et $_$. Leur syntaxe est on ne peut plus simple : ce qui suit immédiatement \wedge est placé en exposant (même chose avec $_$ pour les indices).



Il est également possible de mélanger exposant et indice sans ce soucier de l'ordre de spécification. Dans les exemples qui suivent, regardez avec attention la hauteur de ces éléments.

x^2, l^2, x_2, y_2	x^2, x_i, x_i^2	et	x_i^2
	x^2		
	x_i		
	x_i^2		
$x^2 \times y^2, x^2 y^2$			et 10^{-3}

1 $\backslash(x^2\backslash), \backslash(l^2\backslash),$
 $\backslash(x_2\backslash), \backslash(y_2\backslash)$

2 $\backslash(x^2\backslash), \backslash(x_i\backslash),$
 $\backslash(x^2_i\backslash)$

3 et $\backslash(x_i^2\backslash)$

4

5 $\backslash[x^2\backslash] \backslash[x_i\backslash]$
 $\backslash[x^2_i\backslash]$

6

7 $\backslash(x^2\backslash \mathbf{times} y^2\backslash), \backslash(x^2 y^2\backslash)$

8 et $\backslash(10^{-3}\backslash)$



Le dernier exemple “ 10^{-3} ” est certainement une faute ; on pensait sans doute à 10^{-3} en tapant cette formule. T_EX a fait ce qu’on lui a demandé, en l’occurrence écrire 10, suivi de – en exposant puis un 3 (qui n’est plus en exposant car ne suivant pas immédiatement le symbole \wedge). Pour obtenir la forme désirée, il faut faire suivre l’accent circonflexe d’un groupe qui contiendra tout ce qui sera en exposant : la formule correcte est obtenue avec la syntaxe $10^{\{-3\}}$.



Un exposant particulièrement utilisé en mathématique est le prime (dérivée par exemple). \TeX permet un raccourci sous la forme :

$$f'(x) = 2x$$

$$\text{Si } f'(x) = 2x \text{ alors } f''(x) = 2$$
$$\text{et } f'''(x) = 0.$$

1 $\backslash(f'(x)=2x\backslash)$

2

3 Si $\backslash(f'(x)=2x\backslash)$ alors $\backslash(f''(x)=2\backslash)$

4

et $\backslash(f'''(x)=0\backslash)$.

C'est-à-dire que la mise en exposant est automatique. En regardant à la loupe, on pourra voir que le prime dans la formule mathématique " f' " n'a pas la même forme que l'apostrophe dans le texte " Il f'afroit fur le fable ".



On peut bien sûr employer le symbole / et c'est d'ailleurs mieux dans les formules en texte car les fractions à barre horizontale dans une formule en texte sont quelque peu ridicules (comme dans l'exemple $\frac{1}{2}$). En revanche, dans les formules hors texte, il est souvent préférable de construire une fraction à barre horizontale. Celles-ci se construisent grâce à la macro `\frac` suivi de deux groupes : le premier indiquant le numérateur et le second le dénominateur. Voici un exemple pour clarifier les idées :

$$\frac{x}{y} \div \frac{y}{x} = \frac{x}{y} \times \frac{x}{y} = \frac{x^2}{y^2}$$

```
1  \[ \frac{x}{y} \div \frac{y}{x}
2    = \frac{x}{y} \times \frac{x}{y}
3    = \frac{x^2}{y^2}
4  \]
```

En fait, on aurait pu écrire `\frac x y` à la place de `\frac{x}{y}` mais c'est une mauvaise habitude car les groupes étaient absolument obligatoires pour la dernière fraction. Il y a intérêt à automatiser les gestes lorsqu'on compose des formules mathématiques, les difficultés arrivant bien trop vite !



Le numérateur et le dénominateur sont composés avec un style plus tassé (équivalent au style en texte). Ainsi, un empilement de fraction peut sembler un peu bizarre :

$$\frac{\frac{x^2}{y^2}}{\frac{y^2}{x^2}} = \frac{x^4}{y^4}$$

1 `\[\frac{\frac{x^2}{y^2}}{\frac{y^2}{x^2}}\]`
2 `= \frac{x^4}{y^4}\]`



"displaystyle"

Ce comportement est souvent préférable mais, si on veut absolument avoir la taille normale hors texte, on peut appeler la macro `\displaystyle` (*display* étant ici traduit par *hors texte*) et tout ce qui se trouve à la suite de cette macro à l'intérieur d'un groupe sera mis à la taille " display ".

$$\frac{\frac{x^2}{y^2}}{\frac{y^2}{x^2}} = \frac{x^4}{y^4}$$

1 `\[\frac{\displaystyle\frac{x^2}{y^2}}{\frac{y^2}{x^2}}`
2 `\{\displaystyle\frac{y^2}{x^2}\}`
3 `= \frac{x^4}{y^4}\]`

C'est de cette façon qu'on a obtenu l'exemple pas très joli du début de section : $\frac{1}{2}x = \frac{x}{2}$ qui faisait peu de cas de l'interlignage.



Peu de chose à dire. La macro `sqrt` place ce qui suit sous le signe racine. Comme pour les fractions, il vaut mieux ne pas trop réfléchir et utiliser systématiquement un groupe après cette macro.

$$\sqrt{\sqrt{25} + 4} = 3$$

en texte cela donne :

$$\sqrt{\sqrt{25} + 4} = 3$$

1 `\[\sqrt{\sqrt{25}+4} = 3\]`

2 en texte cela donne :

3 `\(\sqrt{\sqrt{25}+4} = 3\)`



La macro `\sqrt` permet des racines n ième en précisant le n entre crochets juste après la macro.

$$f_n(x) = \sqrt[n]{x+1}$$

1 `\[f_n(x)=\sqrt[n]{x+1}\]`



Points de suspension

\LaTeX offre quatre type de points de suspension. La macro `\ldots` spécifie les points de suspension " normaux ", c'est la seule des quatre macros à pouvoir être utilisée en dehors des modes mathématiques. On la place principalement entre deux virgules ou entre des lettres.

La macros `\cdots` place les points un peu plus haut de façon à ce qu'ils soient alignés avec les signes opératoires usuels (moins, plus, égal, ...).

La macros `\vdots` place trois points verticalement (\vdots) et la macro `\ddots` les place en diagonal (\ddots). On les utilise plutôt dans des matrices.



$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$



La caverne d'Ali Baba aux symboles

Ateliers L^AT_EX
Deuxième atelier:
Mathématiques
(enfin!)
et
structure de
documents

Yves Delhaye

Cette section est une véritable caverne d'Ali Baba. Les scientifiques ayant l'habitude d'utiliser un nombre important de symboles. Nous allons essayer de classifier un peu tout ceci !

Mathématiques
élémentaires

Différents modes
mathématiques

Opérations
élémentaires

Structures
indispensables

Symboles en vrac



Elles sont obtenues en faisant précéder leur nom d'une contre-oblique. Par exemple `\(\beta\)` donnera β . Les majuscules sont obtenues en mettant la première lettre de leur nom en majuscule : `\(\Omega\)` donnera ainsi Ω . Seules les lettres n'existant pas dans l'alphabet romain sont spécifiées. Pour les minuscules, cela n'exclut que le omicron (`\omicron` est donc une macro inexistante) obtenu avec un " o ", en revanche, c'est beaucoup plus fréquent pour les majuscules puisqu'il n'y a que 11 majuscules grecques n'ayant pas d'équivalent graphique dans l'alphabet romain (`\Gamma` Γ , `\Delta` Δ , `\Theta` Θ , `\Lambda` Λ , `\Xi` Ξ , `\Pi` Π , `\Sigma` Σ , `\Upsilon` Υ , `\Phi` Φ , `\Psi` Ψ et `\Omega` Ω).



D'autre part, 6 minuscules présentent deux graphies différentes, à chaque fois l'une d'elle sera obtenue en faisant précéder le nom de la lettre par var. On a :

<code>\epsilon</code>	ϵ	<code>\varepsilon</code>	ε
<code>\theta</code>	θ	<code>\vartheta</code>	ϑ
<code>\pi</code>	π	<code>\varpi</code>	ϖ
<code>\rho</code>	ρ	<code>\varrho</code>	ϱ
<code>\sigma</code>	σ	<code>\varsigma</code>	ς
<code>\phi</code>	ϕ	<code>\varphi</code>	φ

Lettres calligraphiques

La macro `\mathcal` permet de faire appel à la fonte calligraphique. Celle définie par défaut donne le résultat \mathcal{A} , \mathcal{B} , ..., \mathcal{Z} .

Soit f une fonction appartenant à \mathcal{F} .

- 1 Soit (f) une fonction appartenant \mathcal{F} .
- 2 (\mathcal{F}) .

Seules les majuscules sont accessibles.

Lorsque nous discuterons des mathématiques “ avancées ”, nous verrons que les extensions de l’ “ American Mathematical Society ” fournissent d’autres possibilités.



Opérateurs binaires

En plus des signes + et − on trouve un grand nombre d'opérateurs binaires. Ces signes seront séparés de ce qui les entoure de la même façon que pour une addition ou une soustraction. La liste complète est donnée au tableau 2.

<code>\pm</code>	\pm	<code>\cap</code>	\cap	<code>\vee</code>	\vee
<code>\mp</code>	\mp	<code>\cup</code>	\cup	<code>\wedge</code>	\wedge
<code>\setminus</code>	\setminus	<code>\uplus</code>	\uplus	<code>\oplus</code>	\oplus
<code>\cdot</code>	\cdot	<code>\sqcap</code>	\sqcap	<code>\ominus</code>	\ominus
<code>\times</code>	\times	<code>\sqcup</code>	\sqcup	<code>\otimes</code>	\otimes
<code>\ast</code>	\ast	<code>\triangleleft</code>	\triangleleft	<code>\oslash</code>	\oslash
<code>\star</code>	\star	<code>\triangleright</code>	\triangleright	<code>\odot</code>	\odot
<code>\diamond</code>	\diamond	<code>\wr</code>	\wr	<code>\dagger</code>	\dagger
<code>\circ</code>	\circ	<code>\bigcirc</code>	\bigcirc	<code>\ddagger</code>	\ddagger
<code>\bullet</code>	\bullet	<code>\bigtriangleup</code>	\bigtriangleup	<code>\amalg</code>	\amalg
<code>\div</code>	\div	<code>\bigtriangledown</code>	\bigtriangledown		



En plus des signes =, > et <, on trouve également un grand nombre de relations. Ces signes seront séparés de ce qui les entoure par les mêmes espacements que ceux entourant le signe =. La liste complète est donnée dans le tableau 3

<code>\leq</code>	\leq	<code>\geq</code>	\geq	<code>\equiv</code>	\equiv
<code>\prec</code>	\prec	<code>\succ</code>	\succ	<code>\sim</code>	\sim
<code>\preceq</code>	\preceq	<code>\succeq</code>	\succeq	<code>\simeq</code>	\simeq
<code>\ll</code>	\ll	<code>\gg</code>	\gg	<code>\asymp</code>	\asymp
<code>\subset</code>	\subset	<code>\supset</code>	\supset	<code>\approx</code>	\approx
<code>\subseteq</code>	\subseteq	<code>\supseteq</code>	\supseteq	<code>\cong</code>	\cong
<code>\sqsubseteq</code>	\sqsubseteq	<code>\sqsupseteq</code>	\sqsupseteq	<code>\bowtie</code>	\bowtie
<code>\in</code>	\in	<code>\ni</code>	\ni	<code>\propto</code>	\propto
<code>\vdash</code>	\vdash	<code>\dashv</code>	\dashv	<code>\models</code>	\models
<code>\smile</code>	\smile	<code>\mid</code>	\mid	<code>\doteq</code>	\doteq
<code>\frown</code>	\frown	<code>\parallel</code>	\parallel	<code>\perp</code>	\perp

Tab.: Relations



Ces relations peuvent être négativisées en les faisant précéder de la macro `\not`. Par exemple, la séquence `\(\not\approx\)` produira le symbole $\not\approx$.



Grands opérateurs

Ils sont indiqués par soucis d'exhaustivité

. Indiquons toutefois que leur taille varie selon qu'on se trouve en mode en texte ou en mode hors texte. Le tableau 4 en dresse la liste complète :

<code>\sum</code>	Σ	\sum	<code>\bigcap</code>	\cap	\bigcap	<code>\bigodot</code>	\odot	\bigodot
<code>\prod</code>	Π	\prod	<code>\bigcup</code>	\cup	\bigcup	<code>\bigotimes</code>	\otimes	\bigotimes
<code>\coprod</code>	\amalg	\coprod	<code>\bigsqcup</code>	\sqcup	\bigsqcup	<code>\bigoplus</code>	\oplus	\bigoplus
<code>\int</code>	\int	\int	<code>\bigvee</code>	\vee	\bigvee	<code>\biguplus</code>	\uplus	\biguplus
<code>\oint</code>	\oint	\oint	<code>\bigwedge</code>	\wedge	\bigwedge			

Tab.: Grands opérateurs



Les délimiteurs sont des symboles qui encadrent une sous-formule.

(())	\uparrow	↑
[[]]	\downarrow	↓
\{	{ }	}	\updownarrow	↕
\lfloor		\rfloor	\Uparrow	⇑
\lceil		\rceil	\Downarrow	⇓
\langle	<	\rangle	\Updownarrow	↕
/	/	\backslash	\	
		\		

Tab.: Délimiteurs

Flèches

L^AT_EX propose une vaste panoplie de flèches. Le tableau 6 en dresse la liste complète.

<code>\leftarrow</code>	\leftarrow	<code>\longleftarrow</code>	\longleftarrow	<code>\uparrow</code>
<code>\Leftarrow</code>	\Leftarrow	<code>\Longleftarrow</code>	\Longleftarrow	<code>\Uparrow</code>
<code>\rightarrow</code>	\rightarrow	<code>\longrightarrow</code>	\longrightarrow	<code>\downarrow</code>
<code>\Rightarrow</code>	\Rightarrow	<code>\Longrightarrow</code>	\Longrightarrow	<code>\Downarrow</code>
<code>\leftrightharpoonup</code>	\leftrightharpoonup	<code>\longleftarrow</code>	\longleftrightarrow	<code>\updownarrow</code>
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Longleftarrow</code>	\longleftrightarrow	<code>\Updownarrow</code>
<code>\mapsto</code>	\mapsto	<code>\logmapsto</code>	\mapsto	<code>\nearrow</code>
<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\searrow</code>
<code>\leftharpoonup</code>	\leftharpoonup	<code>\rightharpoonup</code>	\rightarrow	<code>\swarrow</code>
<code>\leftharpoondown</code>	\leftharpoondown	<code>\rightharpoondown</code>	\rightarrow	<code>\nwarrow</code>
<code>\rightleftharpoons</code>	\rightleftharpoons			

Tab.: Flèches



Symboles inclassables

Tous ces symboles sont des symboles ordinaires, c'est-à-dire qu'ils se comporteront comme des lettres au niveau des espacements. Ils sont présentés au niveau du tableau 7.

<code>\aleph</code>	\aleph	<code>\prime</code>	$'$	<code>\forall</code>	\forall	<code>\infty</code>	∞
<code>\hbar</code>	\hbar	<code>\emptyset</code>	\emptyset	<code>\exists</code>	\exists	<code>\triangle</code>	\triangle
<code>\imath</code>	\imath	<code>\nabla</code>	∇	<code>\neg</code>	\neg	<code>\clubsuit</code>	\clubsuit
<code>\jmath</code>	\jmath	<code>\surd</code>	\surd	<code>\flat</code>	\flat	<code>\diamondsuit</code>	\diamondsuit
<code>\ell</code>	ℓ	<code>\top</code>	\top	<code>\natural</code>	\natural	<code>\heartsuit</code>	\heartsuit
<code>\wp</code>	\wp	<code>\bot</code>	\bot	<code>\sharp</code>	\sharp	<code>\spadesuit</code>	\spadesuit
<code>\Re</code>	\Re	<code>\ </code>	$\ $	<code>\backslash</code>	\backslash	<code>\partial</code>	∂
<code>\Im</code>	\Im	<code>\angle</code>	\angle				

Tab.: Symboles mathématiques divers



Sous ce terme générique, on trouve tous les mots qui exigent d'être écrits en caractères romains (alors que par défaut, les lettres sont écrites en italique). Le tableau 8 indique toutes les possibilités mises à part les macros `\bmod` et `\pmod`.

`\bmod` produit le texte " mod " mais le considère comme un opérateur binaire (donc avec des espaces de chaque côté).

`\pmod` est une macro qui écrit " mod " avec ce qui suit entre parenthèses. Un petit exemple pour mieux comprendre :

$$\inf(a, b) = m \bmod n$$

$$a \equiv b \pmod{m+n}$$

1 `\[\inf(a, b) = m \bmod n\]`

2 `\[a \equiv b \pmod{m+n}\]`

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

Tab.: Noms de fonction



Accents

Les macros servant à accentuer en mode texte ne sont quelquefois plus valides en mode mathématique. Ce dernier offre 10 sortes d'accents plus 2 de taille variable. Le tableau 9 montre ces macros.

<code>\acute</code>	acute	acute	<code>\check</code>	check	grave	grave	<code>\vec</code>	vec	
<code>\bar</code>	bar	ddot	<code>\ddot</code>	ddot	hat	hat	<code>\widehat</code>	widehat	widehat
<code>\breve</code>	breve	dot	<code>\dot</code>	dot	tilde	tilde	<code>\widetilde</code>	widetilde	widetilde

Tab.: Accents mathématiques

Les deux accents de taille variable n'ont en fait que trois tailles possibles (les deux plus grandes tailles étant montrées dans le tableau).



Synonymes

Certaines macros sont particulièrement utilisées dans les textes mathématiques et elles possèdent des synonymes. Ces synonymes ont pour but de leur donner un nom plus court ou plus parlant (pour un anglophone). Le tableau 10 donne les équivalents possibles.

<code>\not=</code> (\neq)	ou <code>\ne</code> ou <code>\neq</code>		
<code>\geq</code> (\geq)	ou <code>\ge</code>	<code>\leq</code> (\leq)	ou <code>\le</code>
<code>\}</code> ($\}$)	ou <code>\rbrace</code>	<code>\{</code> ($\{$)	ou <code>\lbrace</code>
<code>\leftarrows</code> (\longleftrightarrow)	ou <code>\gets</code>	<code>\rightarrow</code> (\rightarrow)	ou <code>\to</code>
<code>\wedge</code> (\wedge)	ou <code>\land</code> (logical and)	<code>\vee</code> (\vee)	ou <code>\lor</code> (lo)
<code>\neg</code> (\neg)	ou <code>\lnot</code> (logical not)	<code>\ni</code> (\ni)	ou <code>\owns</code>
<code>\ </code> ($\ $)	ou <code>\Vert</code>	<code> </code> ($ $)	ou <code>\vert</code>

Tab.: Synonymes de macros mathématiques

