

# Le package `lymath` : des formules plus sémantiques

Code source disponible sur <https://github.com/bc-latex/ly-math>.

Version 0.7.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-06-08

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Comment lire cette documentation ?</b>	<b>6</b>
<b>3</b>	<b>A propos des macros</b>	<b>6</b>
3.1	Règles de nommage . . . . .	6
3.1.1	Les macros de même « type » . . . . .	6
3.1.2	Les macros en mode <code>displaystyle</code> . . . . .	6
3.2	Versions étoilées . . . . .	7
3.3	Les arguments, deux conventions à connaître . . . . .	7
3.3.1	Nombre fixé d'arguments . . . . .	7
3.3.2	Nombre variable d'arguments . . . . .	7
<b>4</b>	<b>Deux séparateurs d'arguments par défaut</b>	<b>7</b>
<b>5</b>	<b>Quelques gestions d'espaces</b>	<b>7</b>
5.1	Espace et fraction . . . . .	7
5.2	Espace et racines n-ièmes d'un réel . . . . .	7
5.3	Sommes et produits en mode ligne . . . . .	8
5.4	Espace et point-virgule avec l'option <code>french</code> de <code>babel</code> . . . . .	8
<b>6</b>	<b>Logique et fondements</b>	<b>8</b>
6.1	Différents types d'égalités « standard » . . . . .	8
6.1.1	Définir quelque chose . . . . .	8
6.1.2	Indiquer une identité . . . . .	8
6.1.3	Une égalité à vérifier ou non, une hypothèse, une condition . . . . .	8
6.1.4	Une égalité indiquant le choix d'une valeur . . . . .	9
6.1.5	Une égalité indiquant l'équation d'une courbe que l'on utilise . . . . .	9
6.1.6	Différents types d'inéquations . . . . .	9
6.1.7	Une table récapitulative . . . . .	9

6.1.8	Textes utilisés . . . . .	9
6.2	Équivalences et implications . . . . .	10
6.2.1	Des symboles supplémentaires . . . . .	10
6.2.2	Une table récapitulative . . . . .	10
6.2.3	Équivalences et implications verticales . . . . .	10
6.3	Tables des décorations possibles des opérateurs . . . . .	11
6.4	Détailler un raisonnement . . . . .	11
6.5	Des versions alternatives du quantificateur $\exists$ . . . . .	13
<b>7</b>	<b>Ensembles et applications</b>	<b>14</b>
7.1	Différents types d'ensembles . . . . .	14
7.1.1	Ensembles versus accolades . . . . .	14
7.1.2	Ensembles pour la géométrie . . . . .	14
7.1.3	Ensembles probabilistes . . . . .	14
7.1.4	Ensembles pour l'algèbre générale . . . . .	15
7.2	Ensembles classiques en mathématiques et en informatique théorique . . . . .	15
7.2.1	La liste complète . . . . .	15
7.2.2	Ensembles classiques suffixés . . . . .	15
7.3	Des suffixes à la carte . . . . .	16
7.4	Intervalles . . . . .	16
7.4.1	Intervalles réels - Notation française ( ? ) . . . . .	16
7.4.2	Intervalles réels - Notation américaine . . . . .	17
7.4.3	Intervalles discrets d'entiers . . . . .	17
7.5	Unions et intersections . . . . .	17
7.6	Cardinal, image et compagnie . . . . .	18
7.7	Application totale, partielle, injective, surjective et/ou bijective . . . . .	18
<b>8</b>	<b>Géométrie</b>	<b>19</b>
8.1	Points et lignes . . . . .	19
8.1.1	Points . . . . .	19
8.1.2	Lignes . . . . .	19
8.1.3	Droites parallèles ou non . . . . .	20
8.2	Vecteurs . . . . .	20
8.2.1	Les écrire . . . . .	20
8.2.2	Norme . . . . .	21
8.2.3	Produit scalaire – Écriture minimaliste . . . . .	21
8.2.4	Produit scalaire – Écriture « à la physicienne » . . . . .	21
8.2.5	Produit vectoriel . . . . .	22
8.2.6	Plan – Déterminant de deux vecteurs . . . . .	22
8.3	Coordonnées . . . . .	23
8.4	Nommer un repère . . . . .	23
8.5	Arcs circulaires . . . . .	24
8.6	Angles . . . . .	25
8.6.1	Angles géométriques « intérieurs » . . . . .	25
8.6.2	Angles orientés de vecteurs . . . . .	25
<b>9</b>	<b>Analyse</b>	<b>26</b>
9.1	Constantes . . . . .	26
9.1.1	Constantes classiques . . . . .	26
9.1.2	Constantes latines personnelles . . . . .	26
9.2	La fonction valeur absolue . . . . .	26

9.3	Fonctions nommées spéciales . . . . .	27
9.3.1	Sans paramètre . . . . .	27
9.3.2	Avec un paramètre . . . . .	27
9.3.3	Toutes les fonctions nommées en plus . . . . .	27
9.4	Des notations complémentaires pour des suites spéciales . . . . .	27
9.5	Calcul différentiel . . . . .	28
9.5.1	Les opérateurs $\partial$ et $d$ . . . . .	28
9.5.2	Dérivation totale . . . . .	28
9.5.3	Dérivation partielle . . . . .	29
9.6	Calcul intégral . . . . .	29
9.6.1	L'opérateur crochet – 1 <sup>ère</sup> version . . . . .	29
9.6.2	L'opérateur crochet – 2 <sup>nde</sup> version . . . . .	30
9.6.3	Intégrales multiples . . . . .	30
9.7	Tableaux de variation et de signe . . . . .	30
9.8	Comparaison asymptotique de suites et de fonctions . . . . .	32
9.8.1	Les notations $\mathcal{O}$ et $\mathcal{o}$ . . . . .	32
9.8.2	La notation $\Omega$ . . . . .	32
9.8.3	La notation $\Theta$ . . . . .	32
<b>10</b>	<b>Probabilité</b>	<b>33</b>
10.1	Probabilité « simple » . . . . .	33
10.2	Probabilité conditionnelle . . . . .	33
10.3	Arbres pondérés . . . . .	33
<b>11</b>	<b>Arithmétique</b>	<b>36</b>
11.1	Opérateurs de base . . . . .	36
11.2	Fractions continuées . . . . .	36
11.2.1	Fractions continuées standard . . . . .	36
11.2.2	Fractions continuées généralisées . . . . .	37
11.2.3	Comme une fraction continuée isolée . . . . .	37
11.2.4	L'opérateur $\mathcal{K}$ . . . . .	37
<b>12</b>	<b>Algèbre</b>	<b>38</b>
12.1	Polynômes, séries formelles et compagnie . . . . .	38
12.1.1	Polynômes et fractions polynômiales . . . . .	38
12.1.2	Séries formelles et leurs corps de fractions . . . . .	39
12.1.3	Polynômes de Laurent et séries formelles de Laurent . . . . .	39
12.2	Matrices . . . . .	39
<b>13</b>	<b>Historique</b>	<b>42</b>
<b>14</b>	<b>Toutes les fiches techniques</b>	<b>45</b>
14.1	Deux séparateurs d'arguments par défaut . . . . .	45
14.2	Quelques gestions d'espaces . . . . .	45
14.2.1	Fractions . . . . .	45
14.2.2	Racines n-ièmes . . . . .	45
14.2.3	Sommes et produits . . . . .	45
14.3	Logique et fondements . . . . .	45
14.3.1	Les textes pour les opérateurs de « comparaison algébrique » et de logique . . .	45
14.3.2	Les opérateurs de « comparaison algébrique » . . . . .	46
14.3.3	Les opérateurs de logique . . . . .	47

14.3.4	Les opérateurs de logique « verticaux » . . . . .	48
14.3.5	Les textes pour détailler un raisonnement . . . . .	48
14.3.6	Détailler un raisonnement . . . . .	48
14.3.7	Des versions alternatives du quantificateur $\exists$ . . . . .	48
14.4	Ensembles et applications . . . . .	48
14.4.1	Ensembles versus accolades . . . . .	48
14.4.2	Ensembles pour la géométrie . . . . .	49
14.4.3	Ensembles probabilistes . . . . .	49
14.4.4	Ensembles pour l'algèbre générale . . . . .	49
14.4.5	Ensembles classiques . . . . .	49
14.4.6	Des suffixes à la carte . . . . .	50
14.4.7	Intervalles réels - Notation française ( ? ) . . . . .	50
14.4.8	Intervalles réels - Notation américaine . . . . .	51
14.4.9	Intervalles discrets d'entiers . . . . .	52
14.4.10	Unions et intersections . . . . .	52
14.4.11	Cardinal, image et compagnie . . . . .	52
14.4.12	Application totale, partielle, injective, surjective et/ou bijective . . . . .	52
14.5	Géométrie . . . . .	53
14.5.1	Points . . . . .	53
14.5.2	Lignes . . . . .	53
14.5.3	Droites parallèles ou non . . . . .	53
14.5.4	Vecteurs . . . . .	53
14.5.5	Norme . . . . .	54
14.5.6	Produit scalaire – Écriture minimaliste . . . . .	54
14.5.7	Produit scalaire – Écriture « à la physicienne » . . . . .	54
14.5.8	Produit vectoriel . . . . .	54
14.5.9	Plan – Déterminant de deux vecteurs . . . . .	55
14.5.10	Coordonnées . . . . .	56
14.5.11	Nommer un repère . . . . .	56
14.5.12	Arcs circulaires . . . . .	56
14.5.13	Angles géométriques « intérieurs » . . . . .	57
14.5.14	Angles orientés de vecteurs . . . . .	57
14.6	Analyse . . . . .	57
14.6.1	Constantes classiques . . . . .	57
14.6.2	Constantes latines personnelles . . . . .	57
14.6.3	Sans paramètre . . . . .	58
14.6.4	Avec un paramètre . . . . .	58
14.6.5	Des suites spéciales . . . . .	58
14.6.6	Calcul différentiel . . . . .	59
14.6.7	Dérivation totale . . . . .	59
14.6.8	Dérivation partielle . . . . .	59
14.6.9	L'opérateur crochet – 1 <sup>ère</sup> version . . . . .	60
14.6.10	L'opérateur crochet – 2 <sup>nde</sup> version . . . . .	60
14.6.11	L'opérateur d'intégration standard . . . . .	60
14.6.12	Les notations $\mathcal{O}$ et $\mathcal{o}$ . . . . .	60
14.6.13	La notation $\Omega$ . . . . .	60
14.6.14	La notation $\Theta$ . . . . .	60
14.7	Probabilité . . . . .	60
14.7.1	Probabilité « simple » . . . . .	60
14.7.2	Probabilité conditionnelle . . . . .	61

14.7.3 Arbres pondérés . . . . .	61
14.8 Arithmétique . . . . .	61
14.8.1 Opérateurs de base . . . . .	61
14.8.2 Fractions continuées standard . . . . .	61
14.8.3 Fractions continuées généralisées . . . . .	61
14.8.4 Comme une fraction continuée isolée . . . . .	61
14.8.5 L'opérateur $\mathcal{K}$ . . . . .	62
14.9 Algèbre . . . . .	62
14.9.1 Polynômes, séries formelles et compagnie . . . . .	62

---

# 1 Introduction

L<sup>A</sup>T<sub>E</sub>X est un excellent langage, pour ne pas dire le meilleur, pour rédiger des documents contenant des formules mathématiques. Malheureusement toute la puissance de L<sup>A</sup>T<sub>E</sub>X permet d'écrire des codes très peu sémantiques. Le modeste but du package `lymath` est de fournir quelques macros sémantiques pour la rédaction de formules mathématiques élémentaires. Considérons le code L<sup>A</sup>T<sub>E</sub>X suivant.

```
Sachant que $\frac{df}{dx}(x) = 4 \cos(x^2)$ sur $[a ; b]$, nous avons :  
  
$\int_a^b \cos(x^2) dx = \left[ \frac{1}{4} f(x) \right]_a^b$.
```

Avec `lymath`, vous pouvez écrire le code suivant.

```
Sachant que $\frac{d}{dx}f(x) = 4 \cos(x^2)$ sur $\text{intervalC}\{a\}\{b\}$, nous avons :  
  
$\int_a^b \cos(x^2) \, dd{x} = \hook{\frac{1}{4} f(x)}{\{a\}\{b\}}$.
```

Même si certaines commandes sont plus longues à écrire que ce que permet L<sup>A</sup>T<sub>E</sub>X, il y a trois avantages à utiliser des commandes sémantiques.

1. La mise en forme dans votre document sera consistante.
2. Il est facile de changer une mise en forme sur l'ensemble d'un document.
3. `lymath` résout certains problèmes "complexes" pour vous.

## 2 Comment lire cette documentation ?

Le choix a été fait de fournir des exemples comme documentation du package suivis de fiches techniques des macros-commandes. Les exemples se présentent comme ci-dessous.

```
$\displaystyle \int_a^b \cos(x^2) dx  
= \left[ \frac{1}{4} f(x) \right]_a^b$
```

$$\int_a^b \cos(x^2) dx = \left[ \frac{1}{4} f(x) \right]_a^b$$

## 3 A propos des macros

### 3.1 Règles de nommage

#### 3.1.1 Les macros de même « type »

Les macros partageant une même fonctionnalité mathématique suivent les deux règles suivantes.

1. Un gros préfixe est utilisé : par exemple, `\derpow` et `\derfrac` sont des macros pour rédiger des dérivées de fonctions<sup>1</sup>.
2. Un mini préfixe permet d'indiquer une spécialisation d'une macro : par exemple, `\nparallel` est la version négative de `\parallel` (*le mini préfixe **n** est pour **n**-ot.*).

#### 3.1.2 Les macros en mode `displaystyle`

Les macros évitant d'avoir à taper `\displaystyle` auront un nom commençant par la lettre d.

---

1. Ce choix est assumé même si pour les macros du type `\set...` on obtient un nom faisant penser à « régler ... » au lieu de « ensemble de type ... ».

## 3.2 Versions étoilées

Les versions étoilées proposent des mises en forme correspondant aux cas les moins usuels, les moins pratiques : par exemple, la macro `\derpow` utilise un exposant entre des parenthèses pour rédiger une dérivée n<sup>e</sup> tandis que la version étoilée, si c'est possible, affichera des primes en exposant.

## 3.3 Les arguments, deux conventions à connaître

### 3.3.1 Nombre fixé d'arguments

Dans ce cas, c'est la syntaxe L<sup>A</sup>T<sub>E</sub>X usuelle qui sera à utiliser comme dans `\derfrac{f}{x}`.

### 3.3.2 Nombre variable d'arguments

Certaines macros offrent la possibilité de fournir un nombre variable d'arguments comme dans `\coord{x | y | z | t}` et `\coord{x | y}`. Ceci se fait en utilisant un seul argument, au sens de L<sup>A</sup>T<sub>E</sub>X, dont le contenu est formé de morceaux séparés par des traits verticaux `|`. Ainsi dans `\coord{x | y | z | t}`, l'unique argument `x | y | z | t`, au sens de L<sup>A</sup>T<sub>E</sub>X, sera analysé par `lymath` comme étant formé des quatre arguments `x`, `y`, `z` et `t`.

## 4 Deux séparateurs d'arguments par défaut

La macro `\lymathsep` définit le séparateur d'arguments de premier niveau, et `\lymathsubsep` celui des arguments de deuxième niveau. Cette documentation utilisant l'option `french` de `babel`, la valeur de `\lymathsep` est `;` et celle de `\lymathsubsep` est `,`. Sans ce choix, les valeurs de `\lymathsep` et `\lymathsubsep` seront `,` et `;` respectivement.

## 5 Quelques gestions d'espaces

### 5.1 Espace et fraction

Quand on utilise `\frac` ou `\dfrac`, de petits espaces sont automatiquement ajoutés pour éviter d'avoir des traits de fraction trop petits. Le comportement par défaut se retrouve en utilisant les macros `\stdfrac` et `\stdfrac`. Voici un exemple.

```
$\frac{2}{3} = \stdfrac{2}{3}$ ,  
$\dfrac{2}{3} = \stdfrac{2}{3}$
```

$$\frac{2}{3} = \frac{2}{3}, \frac{2}{3} = \frac{2}{3}$$

### 5.2 Espace et racines n-ièmes d'un réel

`\sqrt` a été redéfini pour ajouter un peu d'espaces. Le comportement par défaut se retrouve en utilisant la macro `\stdsqrt`. Voici un exemple.

```
$\sqrt{2} = \stdsqrt{2}$ ,  
$\sqrt[n]{45} = \stdsqrt[n]{45}$
```

$$\sqrt{2} = \sqrt{2}, \sqrt[n]{45} = \sqrt[n]{45}$$

## 5.3 Sommes et produits en mode ligne

Pour limiter l'espace, L<sup>A</sup>T<sub>E</sub>X affiche  $\sum_{k=0}^n$  et non  $\sum_{k=0}^n$  sauf si l'on utilise la commande `\displaystyle`. Les macros `\dsum` et `\dprod` permettent de se passer de `\displaystyle`. Voici un exemple.

<pre><code>\dsum_{k=0}^n 2^k =</code> <code>\sum_{k=0}^n 2^k\$</code></pre>	$\sum_{k=0}^n 2^k = \sum_{k=0}^n 2^k$
<pre><code>\dprod_{k=1}^n k =</code> <code>\prod_{k=1}^n k\$</code></pre>	$\prod_{k=1}^n k = \prod_{k=1}^n k$

## 5.4 Espace et point-virgule avec l'option french de babel

Seulement si vous utilisez `babel` avec l'option `french`, comme c'est le cas dans cette documentation, alors vous verrez le même espacement autour du point-virgule dans  $A(x;y)$ . Que c'est beau!

# 6 Logique et fondements

## 6.1 Différents types d'égalités « standard »

D'un point de vue pédagogique, il peut être intéressant de disposer de différentes façon d'écrire une égalité, une non égalité ou une inégalité. Bien entendu on tord les règles de typographie avec ce type de pratique mais c'est pour le bien de la communauté.

### 6.1.1 Définir quelque chose

L'exemple suivant montre trois façons de rédiger une égalité signifiant une définition<sup>2</sup> (la section 6.1.8 explique comment est défini le texte « déf »).

<pre><code>\$f(x) \eqdef x^3 + 1\$</code> ou <code>\$f(x) \eqdef* x^3 + 1\$</code></pre>	$f(x) \stackrel{\text{déf}}{=} x^3 + 1 \text{ ou } f(x) := x^3 + 1$
--	---

### 6.1.2 Indiquer une identité

L'exemple suivant montre deux façons de rédiger des identités, la notation symbolique n'étant pas standard (la section 6.1.8 explique comment est défini le texte « id »).

<pre><code>\$(a + b)^2 \eqid a^2 + b^2 + 2 a b\$</code> ou <code>\$(a + b)^2 \eqid* a^2 + b^2 + 2 a b\$ .</code></pre>	$(a+b)^2 \stackrel{\text{id}}{=} a^2+b^2+2ab \text{ ou } (a+b)^2 \Rightarrow a^2+b^2+2ab$
--	---

### 6.1.3 Une égalité à vérifier ou non, une hypothèse, une condition

Se reporter à la section 6.1.8 pour savoir comment sont définis les textes « cond » et « hyp ».

---

2. Le symbole peu courant  $:=$  est utilisé par le langage B qui permet de spécifier et prouver certains programmes.



<code>\$(a + b)^3 \eqtest a^3 + b^3 + 3 a b\$</code>	$(a + b)^3 \stackrel{?}{=} a^3 + b^3 + 3ab$
<code>\$(a + b)^3 \neqid a^3 + b^3 + 3 a b\$</code>	$(a + b)^3 \stackrel{id}{\neq} a^3 + b^3 + 3ab$
<code>\$x \neqhyp 0\$ ou \$x \neqcond 0\$</code>	$x \stackrel{hyp}{\neq} 0 \text{ ou } x \stackrel{cond}{\neq} 0$
<code>\$x \eqcond 0\$ ou \$x \eqhyp 0\$ ?</code>	$x \stackrel{cond}{=} 0 \text{ ou } x \stackrel{hyp}{=} 0 ?$

#### 6.1.4 Une égalité indiquant le choix d'une valeur

La section 6.1.8 permet de savoir comment le texte « *choix* » est défini.

<code>\$x \geqcond 4\$ implique \$x^2 \geqcons 16\$.</code>	$x \stackrel{cond}{\geq} 4 \text{ implique } x^2 \stackrel{cons}{\geq} 16.$
<code>Alors \$x \eqchoice 123\$ donne \$123^2 \geqappli 16\$.</code>	$\text{Alors } x \stackrel{choix}{=} 123 \text{ donne } 123^2 \stackrel{appli}{\geq} 16.$

#### 6.1.5 Une égalité indiquant l'équation d'une courbe que l'on utilise

La section 6.1.8 permet de savoir comment les textes « *graph* » et « *appli* » sont définis (*la macro \setgeo est définie dans la section 7.1.2*).

<code>\$M \in \setgeo{C}: y \eqplot x^2 + 3\$ donne \$y_M \eqappli x_M^2 + 3\$.</code>	$M \in \mathcal{C} : y \stackrel{graph}{=} x^2 + 3 \text{ donne } y_M \stackrel{appli}{=} x_M^2 + 3.$
--	---

#### 6.1.6 Différents types d'inéquations

Le principe reste le même pour les symboles d'équations excepté qu'il n'y a ici aucune écriture purement symbolique. Voici un code « fourre-tout » montrant quelques exemples.

<code>\$x \leqtest x^2\$ ou \$x \ltest x^2\$ ou \$x \geqhyp 1\$ ou \$x \gcond 2\$.</code>	$x \stackrel{?}{\leq} x^2 \text{ ou } x \stackrel{?}{<} x^2 \text{ ou } x \stackrel{hyp}{\geq} 1 \text{ ou } x \stackrel{cond}{>} 2.$
---	---

#### 6.1.7 Une table récapitulative

La table 1 page 11 fournit toutes les associations autorisées entre opérateurs de comparaison et décorations.

#### 6.1.8 Textes utilisés

Voici les macros définissant les textes utilisés qui tiennent compte de l'utilisation ou non de l'option `french` de `babel`. Nous ne donnons que les versions françaises.

<code>\textopappli</code> donne « <i>appli</i> »	<code>\textophyp</code> donne « <i>hyp</i> »
<code>\textopchoice</code> donne « <i>choix</i> »	<code>\textopid</code> donne « <i>id</i> »
<code>\textopcond</code> donne « <i>cond</i> »	<code>\textopplot</code> donne « <i>graph</i> »
<code>\textopcons</code> donne « <i>cons</i> »	<code>\textoptest</code> donne « <i>?</i> »
<code>\textopdef</code> donne « <i>déf</i> »	

## 6.2 Équivalences et implications

### 6.2.1 Des symboles supplémentaires

#### Un premier exemple – Implication réciproque

En plus des opérateurs `\iff` et `\implies` proposés par L<sup>A</sup>T<sub>E</sub>X, il a été ajouté l'opérateur `\liesimp`, où l'on a inversé les groupes syllabiques de `\implies`, un opérateur pour obtenir  $\Leftarrow$ <sup>3</sup>, ainsi que des versions négatives. Voici un exemple d'utilisation.

<code>\$(A \implies B)</code> <code>\iff (B \liesimp A)\$</code>	$(A \implies B) \iff (B \Leftarrow A)$
<code>\$(A \implies B)</code> <code>\notiff (A \notimplies B)\$</code>	$(A \implies B) \niff (A \nRightarrow B)$

#### Un deuxième exemple – Des opérateurs décorés

Tout comme pour les égalités, il existe des versions décorées de type test, hypothèse, condition ... Elles sont toutes présentes dans l'exemple suivant.

<code>\$A \iffappli B \notiffchoix C\$</code>	$A \overset{\text{appli}}{\iff} B \overset{\text{choix}}{\niff} C$
<code>\$A \impliescond B \notimpliescons C\$</code>	$A \overset{\text{cond}}{\implies} B \overset{\text{cons}}{\nimplies} C$
<code>\$A \liesimphyp B \notliesimptest C\$</code>	$A \overset{\text{hyp}}{\Leftarrow} B \overset{?}{\nLeftarrow} C$

### 6.2.2 Une table récapitulative

La table 1 page suivante montre toutes les associations autorisées entre opérateurs logiques et décorations.

### 6.2.3 Équivalences et implications verticales

#### À quoi cela sert-il ?

Dans la section 6.4 est expliqué comment détailler les étapes d'un raisonnement. Avec cet outil, il devient utile d'avoir des versions verticales non décorées des symboles d'équivalence et d'implication. Voici comment les obtenir (*tous les cas possibles ont été indiqués*).

3. Penser aussi aux preuves d'équivalence par double implication.

<pre> \begin{tabular}{ccccc} \$A\$          &amp; &amp; \$B\$ &amp; \$C\$          &amp; \$D\$ &amp; \$E\$          &amp; \$F\$ \\ \$\viff\$       &amp; &amp; \$\vimplies\$ &amp; \$\vliesimp\$  &amp; \$\notviff\$ &amp; \$\notvimplies\$ &amp; \$\notvliesimp\$ \\ \$A\$          &amp; &amp; \$B\$ &amp; \$C\$          &amp; \$D\$ &amp; \$E\$          &amp; \$F\$ \end{tabular} </pre>	$ \begin{array}{cccccc} A & B & C & D & E & F \\ \Downarrow & \Downarrow & \Uparrow & \Downarrow & \Downarrow & \Downarrow \\ A & B & C & D & E & F \end{array} $
---	---

## 6.3 Tables des décorations possibles des opérateurs

La table 1 de la présente page donne toutes les associations autorisées entre opérateurs et décorations. Il faut retenir que les versions étoilées produisent des écritures symboliques.

TABLE 1 – Décorations

	appli	choice	cond	cons	def	def*	hyp	id	id*	plot	test
$\backslash eq$	×	×	×	×	×	×	×	×	×	×	×
$\backslash neq$	×	×	×	×			×	×			×
$\backslash l$	×	×	×	×			×			×	×
$\backslash g$	×	×	×	×			×			×	×
$\backslash leq$	×	×	×	×			×			×	×
$\backslash geq$	×	×	×	×			×			×	×
$\backslash iff$	×	×	×	×			×				×
$\backslash notiff$	×	×	×	×			×				×
$\backslash implies$	×	×	×	×			×				×
$\backslash notimplies$	×	×	×	×			×				×
$\backslash liesimp$	×	×	×	×			×				×
$\backslash notliesimp$	×	×	×	×			×				×

## 6.4 Détailler un raisonnement

### Exemple 1 – Avec les réglages par défaut

L'environnement `explain` permet de détailler les étapes principales d'un calcul ou d'un raisonnement en s'appuyant sur la macro `\explnext` dont le nom vient de « *expl-ain next step* » soit « *expliquer la prochaine étape* »<sup>4</sup>. Voici un exemple avec les réglages par défaut (notez les espaces verticaux ajoutés automatiquement avant et après).

4. Le gros du travail est fait par l'environnement `flalign` du package `amsmath` qui est automatiquement chargé par `lymath`.

```

\begin{explain}
  (a + b)^2
  \explnext{On utilise  $x^2 = x \cdot x$ .}
  (a + b) (a + b)
  \explnext{Double développement depuis la parenthèse gauche.}
  a^2 + a b + b a + b^2
  \explnext{Commutativité du produit.}
  a^2 + 2 a b + b^2
\end{explain}

```

---

```


$$(a + b)^2$$


$$= \quad \{ \text{On utilise } x^2 = x \cdot x. \}$$


$$(a + b)(a + b)$$


$$= \quad \{ \text{Double développement depuis la parenthèse gauche.} \}$$


$$a^2 + ab + ba + b^2$$


$$= \quad \{ \text{Commutativité du produit.} \}$$


$$a^2 + 2ab + b^2$$


```

**Remarque.** La macro `\explnext` utilise les macros constantes suivantes.

- `\textexplainleft` et `\textexplainright` qui donnent `{` et `}` respectivement par défaut.
- `\textexplainspacein` est l'espace entre le symbole et la courte explication. Par défaut, cette macro vaut `2em`.

## Exemple 2 – Utiliser un autre symbole globalement

L'environnement `explain` possède un argument optionnel qui est `=` par défaut. Ceci permet de faire ce qui suit sans effort.

<pre> \begin{explain}[\viff]   x^2 + 10 x + 25 = 0   \explnext{Identité remarquable.}   (x + 5)^2 = 0   \explnext{<math>X^2 = 0</math> si et              seulement si <math>X = 0</math>.}   x = -5 \end{explain} </pre>	$x^2 + 10x + 25 = 0$ $\Leftrightarrow \quad \{ \text{Identité remarquable.} \}$ $(x + 5)^2 = 0$ $\Leftrightarrow \quad \{ X^2 = 0 \text{ si et seulement si } X = 0. \}$ $x = -5$
---	---

## Exemple 3 – Juste utiliser des symboles

Si l'argument obligatoire de la macro `\explnext` est vide alors seul le symbole est affiché. Voici un court exemple de ceci.

<pre> \begin{explain}[\viff]   a^2 = b^2   \explnext{}   a = \pm b \end{explain} </pre>	$a^2 = b^2$ $\Leftrightarrow$ $a = \pm b$
---	---

### Exemple 4 – Utiliser un autre symbole localement

La macro `\explnext` possède un argument optionnel qui utilise par défaut celui de l’environnement. En utilisant cette option, on choisit alors localement le symbole à employer. Voici un exemple farfelu d’utilisation.

<pre> \begin{explain}[\viff]   0 \leq a &lt; b     \explnext[\vimplies]%       {Croissance de <math>x^2</math>        sur <math>\mathbb{R}_+</math>.}   <math>a^2 &lt; b^2</math>     \explnext{}   <math>a^2 - b^2 &lt; 0</math>     \explnext{Identité remarquable.}   <math>(a - b)(a + b) &lt; 0</math>     \explnext[\vimplies]{}   <math>a \neq b</math> \end{explain} </pre>	$0 \leq a < b$ $\Downarrow \quad \{ \text{Croissance de } x^2 \text{ sur } \mathbb{R}_+. \}$ $a^2 < b^2$ $\Updownarrow$ $a^2 - b^2 < 0$ $\Updownarrow \quad \{ \text{Identité remarquable.} \}$ $(a - b)(a + b) < 0$ $\Downarrow$ $a \neq b$
---	--

## 6.5 Des versions alternatives du quantificateur $\exists$

### Quantifier l’existence

Voici deux versions, l’une classique, et l’autre beaucoup moins, permettant de préciser le quantificateur  $\exists$ .

<pre> <math>\exists</math>\existssone <math>x \in E</math> pour \og il existe un seul <math>x</math> \fg.  <math>\exists</math>\existmulti{\leq 1} <math>y \in F</math> pour \og il existe au plus un <math>y</math> \fg.  <math>\exists</math>\existmulti{=1} \eqdef \existssone\$ </pre>	$\exists! x \in E$ pour « il existe un seul $x$ ». $\exists_{\leq 1} y \in F$ pour « il existe au plus un $y$ ». $\exists_{=1} \stackrel{\text{déf}}{=} \exists!$
--	---

### Versions négatives

<pre> <math>\nexists</math>\nexistssone\$ pour \og il n’existe pas un unique \fg.  <math>\nexists</math>\nexistmulti{&gt;4}\$ pour \og il n’existe pas plus de quatre \fg.  <math>\nexists</math>\nexistss\$ est proposé par \verb+amssymb+. </pre>	$\nexists!$ pour « il n’existe pas un unique ». $\nexists_{>4}$ pour « il n’existe pas plus de quatre ». $\nexists$ est proposé par <code>amssymb</code> .
---	--

## 7 Ensembles et applications

### 7.1 Différents types d'ensembles

#### 7.1.1 Ensembles versus accolades

Exemple d'utilisation 1

<code>\$\setgene{1 ; 3 ; 5}\$ .</code>	$\{1;3;5\}$ .
--	---------------

Exemple d'utilisation 2

<code>\$\displaystyle</code> <code>\setgene{\frac{1}{3} ; \frac{5}{7} ;</code> <code>\frac{9}{11}}\$</code> ou <code>\$\displaystyle</code> <code>\setgene*{\frac{1}{3} ; \frac{5}{7} ;</code> <code>\frac{9}{11}}\$ .</code>	$\left\{\frac{1}{3}; \frac{5}{7}; \frac{9}{11}\right\}$ ou $\left\{\frac{1}{3}; \frac{5}{7}; \frac{9}{11}\right\}$ .
---	--

#### 7.1.2 Ensembles pour la géométrie

Exemple d'utilisation 1

<code>\$\setgeo{C}\$ ,</code> <code>\$\setgeo{D}\$ ou</code> <code>\$\setgeo{d}\$</code>	$\mathcal{C}$ , $\mathcal{D}$ ou $d$
--	--------------------------------------

**Remarque.** Pour le moment, il n'est pas possible de taper `$\setgeo{ABC}$` avec plusieurs lettres.

Exemple d'utilisation 2 – Avec des indices

<code>\$\setgeo*{C}{1}\$ ou</code> <code>\$\setgeo*{C}{2}\$</code>	$\mathcal{C}_1$ ou $\mathcal{C}_2$
---	------------------------------------

#### 7.1.3 Ensembles probabilistes

Exemple d'utilisation 1

<code>\$\setproba{E}\$ ou</code> <code>\$\setproba{G}\$</code>	$\mathcal{E}$ ou $\mathcal{G}$
---	--------------------------------

**Remarque.** Pour le moment, il n'est pas possible de taper `$\setproba{ABC}$` avec plusieurs lettres.

Exemple d'utilisation 2 – Avec des indices

<code>\$\setproba*{E}{1}\$ ou</code> <code>\$\setproba*{E}{2}\$</code>	$\mathcal{E}_1$ ou $\mathcal{E}_2$
---	------------------------------------

### 7.1.4 Ensembles pour l'algèbre générale

#### Exemple d'utilisation 1

$\backslash setalge{A}$ , $\backslash setalge{K}$ , $\backslash setalge{h}$ ou $\backslash setalge{k}$	$A$ , $K$ , $h$ ou $k$
---	------------------------

**Remarque.** Pour le moment, il n'est pas possible de taper  $\backslash setalge{ABC}$  avec plusieurs lettres.

#### Exemple d'utilisation 2 – Avec des indices

$\backslash setalge*{k}{1}$ ou $\backslash setalge*{k}{2}$	$k_1$ ou $k_2$
--	----------------

## 7.2 Ensembles classiques en mathématiques et en informatique théorique

### 7.2.1 La liste complète

Dans l'exemple suivant,  $\mathbb{P}$  désigne l'ensemble des nombres premiers,  $\mathbb{H}$  celui des quaternions,  $\mathbb{O}$  celui des octonions et  $\mathbb{F}$  un ensemble de nombres flottants (*notation à préciser suivant le contexte*).

$\backslash nullset$ , $\backslash NN$ , $\backslash ZZ$ , $\backslash PP$  $\backslash DD$ , $\backslash QQ$ , $\backslash RR$ , $\backslash CC$  $\backslash HH$ , $\backslash OO$  $\backslash FF$	$\emptyset$ , $\mathbb{N}$ , $\mathbb{Z}$ , $\mathbb{P}$ $\mathbb{D}$ , $\mathbb{Q}$ , $\mathbb{R}$ , $\mathbb{C}$ $\mathbb{H}$ , $\mathbb{O}$ $\mathbb{F}$
---	--

### 7.2.2 Ensembles classiques suffixés

L'ensemble  $\mathbb{R}$  nous permet de voir tous les cas possibles.

$\backslash RRn$ , $\backslash RRp$ , $\backslash RRs$ , $\backslash RRsn$ ou $\backslash RRsp$	$\mathbb{R}_-$ , $\mathbb{R}_+$ , $\mathbb{R}^*$ , $\mathbb{R}_-^*$ ou $\mathbb{R}_+^*$
---	---

Nous avons utilisé les suffixes **n** pour **N**egatif, **p** pour **P**ositif, et **s** pour **s**tar, soit "étoile" en anglais. Il y a aussi les suffixes composites **sn** et **sp**.

Notez qu'il est interdit d'utiliser  $\backslash CCn$  pour  $\mathbb{C}_-$  car l'ensemble  $\mathbb{C}$  ne possède pas de structure ordonnée standard. Jetez un oeil à la section suivante pour apprendre à taper  $\mathbb{C}_-$  si vous en avez besoin. L'interdiction est ici purement sémantique !

**Remarque.** La table 2 page suivante montre les associations autorisées entre ensembles classiques et suffixes.

TABLE 2 – Suffixes

	n	p	s	sn	sp
\NN			×		
\PP					
\ZZ	×	×	×	×	×
\DD	×	×	×	×	×
\QQ	×	×	×	×	×
\RR	×	×	×	×	×
\CC			×		
\HH			×		
\OO			×		
\FF	×	×	×	×	×

## 7.3 Des suffixes à la carte

### Exemple d'utilisation

Dans cet exemple, il faut savoir que le 2<sup>e</sup> argument ne peut prendre que les valeurs **n**, **p**, **s**, **sn** ou **sp**.

```
\setspecial{\CC}{n}$ ,
\setspecial{\HH}{sp}$ ou
\setspecial*{\setproba{P}}{n}$
```

$\mathbb{C}_-$ ,  $\mathbb{H}_+^*$  ou  $\mathcal{P}_{\leq 0}$

## 7.4 Intervalles

### 7.4.1 Intervalles réels - Notation française ( ? )

#### Exemple d'utilisation 1

Dans cet exemple, la syntaxe fait référence à **O**-pened et **C**-losed pour "ouvert" et "fermé" en anglais. Nous verrons que **CC** et **OO** sont contractés en **C** et **O**. Notez au passage que la macro utilisée résout un problème d'espacement vis à vis du signe = .

```
$I = ]a ; b] = \intervalOC{a}{b}$
```

$I = ]a ; b] = ]a ; b]$

#### Exemple d'utilisation 2

Les crochets s'étendent verticalement automatiquement. Pour empêcher cela, il suffit d'utiliser la version étoilée de la macro. Dans ce cas, les crochets restent tout de même un peu plus grands que des crochets utilisés directement. Voici un exemple.

```
$\displaystyle
\intervalC{ \frac{1}{2} }{ 1^{2^3} }
=
[ \frac{1}{2} ; 1^{2^3} ]
=
\intervalC*{ \frac{1}{2} }{ 1^{2^3} }$
```

$\left[ \frac{1}{2} ; 1^{2^3} \right] = \left[ \frac{1}{2} ; 1^{2^3} \right] = \left[ \frac{1}{2} ; 1^{2^3} \right]$



## 7.4.2 Intervalles réels - Notation américaine

### Exemple d'utilisation

Dans cet exemple, la syntaxe fait référence à **P**-arenthèse. Cette notation est utilisée États Unis.

$\begin{aligned} &\$ \backslash intervalPC{a}{b} = \backslash intervalOC{a}{b} \$ \\ &\text{et} \\ &\$ \backslash intervalP{a}{b} = \backslash intervalO{a}{b} \$ . \end{aligned}$	$(a; b] = ]a; b] \text{ et } (a; b) = ]a; b[.$
--	--

## 7.4.3 Intervalles discrets d'entiers

### Exemple d'utilisation

Dans l'exemple, la syntaxe fait référence à  $\mathbb{Z}$  l'ensemble des entiers relatifs.

$\begin{aligned} &\$ \backslash ZintervalC{-1}{4} \\ &= \backslash \{ -1 ; 0 ; 1 ; 2 ; 3 ; 4 \} \$ \\ \\ &\$ \backslash ZintervalC{-1}{4} \\ &= \backslash ZintervalO{-2}{5} \$ . \end{aligned}$	$\begin{aligned} &\llbracket -1; 4 \rrbracket = \{-1; 0; 1; 2; 3; 4\} \\ &\llbracket -1; 4 \rrbracket = \llbracket -2; 5 \rrbracket . \end{aligned}$
--	--

## 7.5 Unions et intersections

### Exemple d'unions

Ci-dessous est utilisée la macro `\bigcup` proposée par le package `amssymb`.

$\begin{aligned} &\$A \backslash cup B\$ \\ \\ &\$ \backslash dcup_{k=1}^n A_k \$ \\ \\ &\$ \backslash bigcup_{k=1}^n B_k \$ \\ \\ &\$ \backslash displaystyle \backslash bigcup_{k=1}^n C_k \$ \end{aligned}$	$\begin{aligned} &A \cup B \\ &\bigcup_{k=1}^n A_k \\ &\bigcup_{k=1}^n B_k \\ &\bigcup_{k=1}^n C_k \end{aligned}$
--	---

### Exemple d'unions disjointes

Ci-dessous sont utilisées les macros `\sqcup` et `\bigsqcup` proposée par le package `amssymb`.

$\begin{aligned} &\$A \backslash sqcup B\$ \\ \\ &\$ \backslash dsqcup_{k=1}^n A_k \$ \\ \\ &\$ \backslash bigsqcup_{k=1}^n B_k \$ \\ \\ &\$ \backslash displaystyle \backslash bigsqcup_{k=1}^n C_k \$ \end{aligned}$	$\begin{aligned} &A \sqcup B \\ &\sqcup_{k=1}^n A_k \\ &\sqcup_{k=1}^n B_k \\ &\sqcup_{k=1}^n C_k \end{aligned}$
--	--

### Exemple d'intersections

Ci-dessous est utilisée la macro `\bigcap` proposée par le package `amssymb`.

<code>\$A \cap B\$</code>	$A \cap B$
<code>\$_{\displaystyle \bigcap_{k=1}^n A_k}\$</code>	$\bigcap_{k=1}^n A_k$
<code>\$_{\displaystyle \bigcap_{k=1}^n B_k}\$</code>	$\bigcap_{k=1}^n B_k$
<code>\$_{\displaystyle \bigcap_{k=1}^n C_k}\$</code>	$\bigcap_{k=1}^n C_k$

## 7.6 Cardinal, image et compagnie

### Cardinal – Un exemple

<code>\$_{\displaystyle \#E} = \text{card } E\$</code>	$\#E = \text{card } E$
--	------------------------

### Image et compagnie – Un exemple

<code>\$_{\displaystyle \ker f}\$ , <code>\$_{\displaystyle \text{dom } f}\$ , \$_{\displaystyle \text{im } f}\$ ou <code>\$_{\displaystyle \text{codom } f}\$</code></code></code>	$\ker f$ , $\text{dom } f$ , $\text{im } f$ ou $\text{codom } f$
---	--

## 7.7 Application totale, partielle, injective, surjective et/ou bijective

Voici des symboles qui, bien que très techniques, facilitent la rédaction de documents à propos des applications totales ou partielles<sup>5</sup> (*on parle aussi d'applications, sans qualificatif, et de fonctions*).

### Applications totales – Un exemple complet

<code>\$f: A \to B\$ est une application totale, c'est à dire définie sur \$A\$ tout entier.</code>
<code>\$i: C \hookrightarrow D\$ est une application totale injective.</code>
<code>\$s: E \twoheadrightarrow F\$ est une application totale surjective.</code>
<code>\$b: G \twoheadrightarrow H\$ est une application totale bijective.</code>
-----
<code><math>f : A \rightarrow B</math> est une application totale, c'est à dire définie sur <math>A</math> tout entier.</code>
<code><math>i : C \hookrightarrow D</math> est une application totale injective.</code>
<code><math>s : E \twoheadrightarrow F</math> est une application totale surjective.</code>
<code><math>b : G \twoheadrightarrow H</math> est une application totale bijective.</code>

5.  $a : E \rightarrow F$  est une application totale si  $\forall x \in E, \exists ! y \in F$  tel que  $y = a(x)$ . Plus généralement,  $f : E \rightarrow F$  est une application partielle si  $\forall x \in E, \exists \leq 1 y \in F$  tel que  $y = f(x)$ , autrement dit soit  $f(x)$  existe dans  $F$ , soit  $f$  n'est pas définie en  $x$ .

## Applications partielles – Un exemple complet

`$f: A \pto B$` est une application partielle, c'est à dire définie sur un sous-ensemble de  $A$ .

`$i: C \ponetoone D$` est une application partielle injective.

`$s: E \ponto F$` est une application partielle surjective.

`$b: G \pbiject H$` est une application partielle bijective.

$f: A \rightarrowtail B$  est une application partielle, c'est à dire définie sur un sous-ensemble de  $A$ .

$i: C \rightarrowtail D$  est une application partielle injective.

$s: E \twoheadrightarrow F$  est une application partielle surjective.

$b: G \twoheadrightarrow H$  est une application partielle bijective.

## 8 Géométrie

### 8.1 Points et lignes

#### 8.1.1 Points

##### Exemple d'utilisation 1

`$\pt{I}$`

$I$

##### Exemple d'utilisation 2

`$\pt*{I}{1}$` ou  
`$\pt*{I}{2}$`

$I_1$  ou  $I_2$

#### 8.1.2 Lignes

##### Exemple 1 – Les droites

Dans l'exemple suivant, le préfixe **g** est pour **g**-eometry, tandis que **p** est pour **p**-oint.

`$\gline{A}{B}$` ,  
`$\gline{\pt{A}}{\pt{B}}$` ou  
`$\pgline{A}{B}$`

$(AB)$  ,  $(AB)$  ou  $(AB)$

##### Exemple 2 – Les segments

Les macros `\segment` et `\psegment` ont un comportement similaire à `\gline` et `\pgline`.

`$\segment{A}{B}$` ,  
`$\segment{\pt{A}}{\pt{B}}$` ou  
`$\psegment{A}{B}$`

$[AB]$  ,  $[AB]$  ou  $[AB]$

### Exemple 3 – Les demi-droites

Dans l'exemple suivant, le préfixe `h` est pour **h**-alf soit moitié en anglais.

<code>\hgline{A}{B}\$ ,</code> <code>\hgline{\pt{A}}{\pt{B}}\$ ou</code> <code>\phgline{A}{B}\$</code>	$[AB)$ , $[AB)$ ou $[AB)$
--	---------------------------

### Exemple 4 – D'autres demi-droites

Ce qui suit nécessite d'utiliser l'argument optionnel de `\gline` et `\pgline`. La valeur `OC` provient de **O**-pened-**C**-losed, soit ouvert-fermé en anglais.

<code>\gline[OC]{A}{B}\$ ,</code> <code>\gline[OC]{\pt{A}}{\pt{B}}\$ ou</code> <code>\pgline[OC]{A}{B}\$</code>	$(AB]$ , $(AB]$ ou $(AB]$
---	---------------------------

**Remarque.** Les segments utilisent en fait l'option `C` et les demi-droites standard l'option `CO`. La valeur par défaut est `O`.

#### 8.1.3 Droites parallèles ou non

Les opérateurs `\parallel` et `\nparallel` utilisent des obliques au lieu de barres verticales comme le montre l'exemple qui suit où `\stdnparallel` est un alias de `\nparallel` fourni par le package `amssymb`, et `\stdparallel` est un alias de la version standard de `\parallel` proposée par L<sup>A</sup>T<sub>E</sub>X.

<code>\pgline{A}{B} \parallel \pgline{C}{D}\$</code> au lieu de <code>\pgline{A}{B}</code> <code>\stdparallel \pgline{C}{D}\$</code>	$(AB) \parallel (CD)$ au lieu de $(AB) \parallel (CD)$ $(EF) \nparallel (GH)$ au lieu de $(EF) \nparallel (GH)$
<code>\pgline{E}{F} \nparallel \pgline{G}{H}\$</code> au lieu de <code>\pgline{E}{F}</code> <code>\stdnparallel \pgline{G}{H}\$</code>	

## 8.2 Vecteurs

### 8.2.1 Les écrire

#### Exemple d'utilisation 1

<code>\vect{ABCDEFG}\$ ,</code> <code>\vect*{e}{rot}\$ ou</code> <code>\vect{e_{rot}}\$</code>	$\overrightarrow{ABCDEFG}$ , $\vec{e}_{rot}$ ou $\overrightarrow{e_{rot}}$
--	--

#### Exemple d'utilisation 2

<code>\vect{i}\$ ou</code> <code>\vect*{j}{2}\$</code>	$\vec{i}$ ou $\vec{j}_2$
---	--------------------------

## 8.2.2 Norme

### Exemple d'utilisation

<code><math>\ \vec{i}\ </math></code>	$\ \vec{i}\ $
<code><math>\ \frac{2}{7}\vec{e}_k\ </math></code>	$\ \frac{2}{7}\vec{e}_k\ $
<code><math>\ \frac{2}{7}\vec{e}_k\ </math></code>	$\ \frac{2}{7}\vec{e}_k\ $

**Remarque.** Le code L<sup>A</sup>T<sub>E</sub>X vient directement de ce message : <https://tex.stackexchange.com/a/43009/6880>.

## 8.2.3 Produit scalaire – Écriture minimaliste

### Exemple d'utilisation - Version longue

<code><math>\frac{1}{2}\vec{i} \cdot \vec{j}</math></code>	$\frac{1}{2}\vec{i} \cdot \vec{j}$
--	------------------------------------

### Exemple d'utilisation - Version courte mais restrictive

Dans l'exemple suivant, le préfixe v est pour **v**-ector.

<code><math>\vec{i} \cdot \vec{j}</math></code>	$\vec{i} \cdot \vec{j}$
---	-------------------------

## 8.2.4 Produit scalaire – Écriture « à la physicienne »

Dans l'exemple suivant, le préfixe a est pour **a**-ngle, et v pour **v**-ector. Les physiciens aiment bien cette notation.

<code><math>\langle \frac{1}{2}\vec{i}   \vec{j} \rangle</math></code>	$\langle \frac{1}{2}\vec{i}   \vec{j} \rangle$
<code><math>\langle \frac{1}{2}\vec{i}   \vec{j} \rangle</math></code>	$\langle \frac{1}{2}\vec{i}   \vec{j} \rangle$
<code><math>\langle \vec{i}   \vec{j} \rangle</math></code>	$\langle \vec{i}   \vec{j} \rangle$
<code><math>\langle \vec{i}   \vec{j} \rangle</math></code>	$\langle \vec{i}   \vec{j} \rangle$

## 8.2.5 Produit vectoriel

### Exemple d'utilisation - Version longue

```
\crossprod{\dfrac{1}{2} \vect{i}}%
{\vect{j}}$
```

$$\frac{1}{2} \vec{i} \wedge \vec{j}$$

### Exemple d'utilisation - Version courte mais restrictive

```
\vcrossprod{i}{j}$
```

$$\vec{i} \wedge \vec{j}$$

### Exemple d'utilisation - Explication des calculs

Dans l'exemple suivant, le préfixe **c**, qui double cette lettre au début, est pour **c**-ordinates, et **v** pour **v**-ector.

```
\ccrossprod{\vect{u}}{x }{y }{z }%
{\vect{v}}{x'}{y'}{z'}
=
\vcrossprod{u}{x }{y }{z }%
{v}{x'}{y'}{z'}$
```

$$\begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \\ z & z' \end{array} \right| \end{array} = \begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \\ x & x' \end{array} \right| \end{array}$$

## 8.2.6 Plan – Déterminant de deux vecteurs

### Exemple d'utilisation - Version décorée

Dans l'exemple suivant, le préfixe **c** est pour **c**-ordinates.

```
\cdetplane{\vect{u}}{x }{y }%
{\vect{v}}{x'}{y'}$
```

$$\begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \end{array} \right| \end{array}$$

### Exemple d'utilisation - Version non décorée

```
\cdetplane*{\vect{u}}{x }{y }%
{\vect{v}}{x'}{y'}$
```

$$\begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \end{array} \right| \end{array}$$

### Exemple d'utilisation - Versions courtes

Dans l'exemple suivant, le préfixe **v** est pour **v**-ector.

```
\vcdetplane{u}{x }{y }%
{v}{x'}{y'}
=
\vcdetplane*{u}{x }{y }%
{v}{x'}{y'}$
```

$$\begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \end{array} \right| \end{array} = \begin{array}{c} \vec{u} \quad \vec{v} \\ \left| \begin{array}{cc} x & x' \\ y & y' \end{array} \right| \end{array}$$

## 8.3 Coordonnées

### Exemple d'utilisation 1

```
$\displaystyle
\pt{I} \coord{\frac{1}{3} | -4 | 0}$
ou
$\displaystyle
\pt{I} \coord*{\frac{1}{3} | -4 | 0}$
```

$$I\left(\frac{1}{3}; -4; 0\right) \text{ ou } I\left(\frac{1}{3}; -4; 0\right)$$

### Exemple d'utilisation 2

Dans l'exemple suivant, le préfixe v est pour **v**-ertical.

```
$\vect{i} \vcoord{3 | -4 | 0}$ ,
$\vect{i} \vcoord*{3 | -4 | 0}$
à la place de
$\vect{i} \coord{3 | -4 | 0}$
```

$$\vec{i}\begin{pmatrix} 3 \\ -4 \\ 0 \end{pmatrix}, \vec{i}\begin{bmatrix} 3 \\ -4 \\ 0 \end{bmatrix} \text{ à la place de } \vec{i}(3; -4; 0)$$

## 8.4 Nommer un repère

### Exemple d'utilisation 1 – La méthode basique

Commençons par la manière la plus basique d'écrire un repère (*nous verrons d'autres méthodes qui peuvent être plus efficaces*).

```
$\axes{\pt{O} %
| \pt{I} | \pt{J}}$
```

$$(O; I, J)$$

### Exemple d'utilisation 2 – La méthode basique en version étoilée

Dans l'exemple ci-dessous, on voit que la version étoilée produit des petites parenthèses.

```
$\displaystyle
\axes{\pt{O} %
| \frac{7}{3} \vect{i} %
| \vect{j}}$
ou
$\displaystyle
\axes*{\pt{O} %
| \frac{7}{3} \vect{i} %
| \vect{j}}$
```

$$\left(O; \frac{7}{3} \vec{i}, \vec{j}\right) \text{ ou } \left(O; \frac{7}{3} \vec{i}, \vec{j}\right)$$

### Exemple d'utilisation 3 – La méthode basique en dimension quelconque

Il faut au minimum deux "morceaux" séparés par des barres |, cas de la dimension 1, mais il n'y a pas de maximum, cas d'une dimension quelconque  $n > 0$ .

<pre> <math>\backslash</math>axes{\pt{0} %       \vect*{i}{1} %       \vect*{i}{2} %       \vect*{i}{3} %       \dots %       \vect*{i}{9} %       \vect*{i}{10} %       \vect*{i}{11} %       \vect*{i}{12}}\$ </pre>	$(O; \vec{i}_1, \vec{i}_2, \vec{i}_3, \dots, \vec{i}_9, \vec{i}_{10}, \vec{i}_{11}, \vec{i}_{12})$
--	--

#### Exemple d'utilisation 4 – Repère affine

Dans l'exemple suivant, le préfixe p est pour p-oint.

<pre> <math>\backslash</math>paxes{0   I   J   K}\$ au lieu de <math>\backslash</math>axes{\pt{0} %       \pt{I}   \pt{J}   \pt{K}}\$ </pre>	$(O; I, J, K)$ au lieu de $(O; I, J, K)$
--	--

#### Exemple d'utilisation 5 – Repère vectoriel (méthode 1)

Dans l'exemple suivant, le préfixe v est pour v-ecteur.

<pre> <math>\backslash</math>vaxes{\pt{0}   i   j}\$ au lieu de <math>\backslash</math>axes{\pt{0}   \vect{i}   \vect{j}}\$ </pre>	$(O; \vec{i}, \vec{j})$ au lieu de $(O; \vec{i}, \vec{j})$
--	--

#### Exemple d'utilisation 6 – Repère vectoriel (méthode 2)

Dans l'exemple suivant, le préfixe pv permet de combiner ensemble les fonctionnalités proposées par les préfixes p et v.

<pre> <math>\backslash</math>pvaxes{0   i   j}\$ au lieu de <math>\backslash</math>axes{\pt{0}   \vect{i}   \vect{j}}\$ </pre>	$(O; \vec{i}, \vec{j})$ au lieu de $(O; \vec{i}, \vec{j})$
--	--

## 8.5 Arcs circulaires

#### Exemple d'utilisation 1

<pre> <math>\backslash</math>circarc{ABCDEF}\$ , <math>\backslash</math>circarc*{A}{rot}\$ ou <math>\backslash</math>circarc{A_{rot}}\$ </pre>	$\widehat{ABCDEF}$ , $\widehat{A}_{rot}$ ou $\widehat{A_{rot}}$
--	---

#### Exemple d'utilisation 2

<pre> <math>\backslash</math>circarc{i}\$ ou <math>\backslash</math>circarc*{j}{2}\$ </pre>	$\widehat{i}$ ou $\widehat{j}_2$
---	----------------------------------



## 8.6 Angles

### 8.6.1 Angles géométriques « intérieurs »

#### Exemple d'utilisation 1

```
$\anglein{ABCDEF}$ ,  
$\anglein*{A}{rot}$ ou  
$\anglein{A_{rot}}$
```

$\widehat{ABCDEF}$  ,  $\hat{A}_{rot}$  ou  $\widehat{A_{rot}}$

#### Exemple d'utilisation 2

```
$\anglein{i}$ ou  
$\anglein*{j}{2}$
```

$\hat{i}$  ou  $\hat{j}_2$

### 8.6.2 Angles orientés de vecteurs

#### Sans chapeau - Version longue

```
$\displaystyle  
 \angleorient{\frac{1}{2} \vect{i}}%  
           {\vect{j}}$  
ou  
$\displaystyle  
 \angleorient*{\frac{1}{2} \vect{i}}%  
           {\vect{j}}$ .
```

$\left( \frac{1}{2} \vec{i} ; \vec{j} \right)$  ou  $\left( \frac{1}{2} \vec{i} ; \vec{j} \right)$  .

#### Sans chapeau - Version courte mais restrictive

Dans l'exemple suivant, le préfixe **v** est pour **v**-ector qui permet de simplifier la saisie quand l'on a juste des vecteurs nommés avec des lettres (*notez que la version étoilée n'apporte rien de nouveau*).

```
$\vangleorient{i}{j}$  
ou  
$\vangleorient*{i}{j}$
```

$(\vec{i} ; \vec{j})$  ou  $(\vec{i} ; \vec{j})$

#### Avec un chapeau

Dans l'exemple suivant, le préfixe **h** est pour **h**-at, et **v** pour **v**-ector. Ceci permet d'avoir les angles orientés avec un chapeau (*la dernière écriture n'apporte rien de neuf*).

<code>\displaystyle</code> <code>\hangleorient{\frac{1}{2} \vect{i}}%</code> <code>{\vect{j}}\$</code>	$\overline{\left(\frac{1}{2}\vec{i}; \vec{j}\right)}$
<code>\displaystyle</code> <code>\hangleorient*{\frac{1}{2} \vect{i}}%</code> <code>{\vect{j}}\$</code>	$\overline{\left(\frac{1}{2}\vec{i}; \vec{j}\right)}$
<code>\hvangleorient{i}{j}\$</code>	$\overline{(\vec{i}; \vec{j})}$
<code>\hvangleorient*{i}{j}\$</code>	$\overline{(\vec{i}; \vec{j})}$

## 9 Analyse

### 9.1 Constantes

#### 9.1.1 Constantes classiques

La liste complète

<code>\gamma\$ , \$\ppi\$ , \$\tttau\$ ,</code> <code>\ee\$ , \$\ii\$ , \$\jj\$</code> et <code>\kk\$</code> où <code>\tttau = 2 \ppi\$</code>	$\gamma, \pi, \tau, e, i, j$ et $k$ où $\tau = 2\pi$
--	--

**Remarque.** Faites attention car `\Large $\ppi \neq \pi$` produit  $\pi \neq \pi$ . Comme vous le constatez, les symboles ne sont pas identiques. Ceci est vraie pour toutes les constantes grecques.

#### 9.1.2 Constantes latines personnelles

Exemple d'utilisation

<code>\ct{a} x^2 + \ct{b} x + \ct{c}\$</code> ou <code>\$a x^2 + b x + c\$</code>	$ax^2 + bx + c$ ou $ax^2 + bx + c$
---	------------------------------------

### 9.2 La fonction valeur absolue

Un exemple d'utilisation

<code>\abs{2}\$ ,</code> <code>\displaystyle \abs{\frac{3}{5}}\$</code> ou <code>\displaystyle \abs*{\frac{3}{5}}\$</code>	$ 2 , \left \frac{3}{5}\right $ ou $\left \frac{3}{5}\right $
--	---

**Remarque.** Le code L<sup>A</sup>T<sub>E</sub>X vient directement de ce poste : <https://tex.stackexchange.com/a/43009/6880>.

## 9.3 Fonctions nommées spéciales

### 9.3.1 Sans paramètre

#### Un exemple d'utilisation

Quelques fonctions nommées supplémentaires où **fch** est pour « f-rench » cosinus hyperbolique (*ce choix a été fait pour éviter des incompatibilités avec quelques autres packages*). La liste complète est donnée un peu plus bas dans une section dédiée.

$\$ \backslash fch x \backslash neq ch x$ ,$ $\$ \backslash ppcm(x;y)$$ ou $\$ \backslash lg x$$	$ch x \neq chx$ , $ppcm(x;y)$ ou $lg x$
--	---

### 9.3.2 Avec un paramètre

#### Un exemple d'utilisation

$\$ \backslash logb\{2\} x = \backslash lg x$$ ou $\$ \backslash expb\{6\} y = 6^y$$	$\log_2 x = lg x$ ou $\exp_6 y = 6^y$
---	---------------------------------------

### 9.3.3 Toutes les fonctions nommées en plus

pgcd : pgcd...	atanh : atanh...
ppcm : ppcm...	fch : ch...
acos : acos...	fsh : sh...
asin : asin...	fth : th...
atan : atan...	afch : ach...
arccosh : arccosh...	afsh : ash...
arcsinh : arcsinh...	afth : ath...
arctanh : arctanh...	expb{p} : $\exp_p$ ...
acosh : acosh...	logb{p} : $\log_p$ ...
asinh : asinh...	

## 9.4 Des notations complémentaires pour des suites spéciales

#### Exemple d'utilisation

$\$ \backslash seqplus\{F\}\{1\}\{2\}$$ $\$ \backslash seqhypergeo\{F\}\{1\}\{2\}$$ $\$ \backslash seqsuprageo\{F\}\{1\}\{2\}\{3\}\{4\}$$ pour les fous\dots :-)	$F_1^2$ ${}_1F_2$ ${}_1F_2^3$ pour les fous... :-)
---	--

## 9.5 Calcul différentiel

### 9.5.1 Les opérateurs $\partial$ et $d$

#### Exemple d'utilisation

```
\dd{f} , \pp{t} ,
\dd[5]{x} ou \pp[n]{x}
```

$df$  ,  $\partial t$  ,  $d^5x$  ou  $\partial^n x$

### 9.5.2 Dérivation totale

#### Exemple d'utilisation 1

```
\displaystyle
\derpow*{f} (a)
= \derpow{f} (a)
= \derfrac{f}{x} (a)
= \dersub{f}{x} (a)
```

$$f'(a) = f^{(1)}(a) = \frac{df}{dx}(a) = d_x f(a)$$

#### Exemple d'utilisation 2

```
\displaystyle
\derpow*[3]{f}(a)
= \derpow[3]{f} (a)
= \derfrac[3]{f}{x} (a)
= \dersub[3]{f}{x} (a)
```

```
\displaystyle
\derpow*[10]{\cos} a
= \derfrac[10]{\cos}{x} (a)
```

$$f'''(a) = f^{(3)}(a) = \frac{d^3 f}{dx^3}(a) = d_x^3 f(a)$$

$$\cos^{(10)} a = \frac{d^{10} \cos}{dx^{10}}(a)$$

#### Exemple d'utilisation 3

```
\displaystyle
\derpow[3]{f} (a)
= \derfrac*[3]{\left( \frac{1}{x}
\right)}{x} (a)
```

$$f^{(3)}(a) = \frac{d^3}{dx^3} \left( \frac{1}{x} \right) (a)$$

#### Exemple d'utilisation 4

```
\displaystyle
\derpar[3]{\dfrac{1}{2} uv}
= \derpar*[3]{\dfrac{1}{2} uv}$

\displaystyle
\sderpar[3]{\dfrac{1}{2} uv}
= \sderpar*[3]{\dfrac{1}{2} uv}$
```

$$\left( \frac{1}{2} uv \right)^{(3)} = \left( \frac{1}{2} uv \right)'''$$

$$\left( \frac{1}{2} uv \right)^{(3)} = \left( \frac{1}{2} uv \right)'''$$

### 9.5.3 Dérivation partielle

#### Exemple d'utilisation 1

```
\displaystyle
\partialfrac{f}{x} (a;b)
= \partialsub{f}{x} (a;b)
= \partialprime{f}{x} (a;b)$
```

$$\frac{\partial f}{\partial x}(a;b) = \partial_x f(a;b) = f'_x(a;b)$$

#### Exemple d'utilisation 2

```
\displaystyle
\partialfrac[3]{G}{f^2 | v} (a;b)
= \partialfrac{G}{f^2 | v} (a;b)
= \dots$

\displaystyle
\dots
= \partialsub{G}{f^2 | v} (a;b)
= \partialprime{G}{f^2 | v} (a;b)$
```

$$\frac{\partial^3 G}{\partial f^2 \partial v}(a;b) = \frac{\partial G}{\partial f^2 \partial v}(a;b) = \dots$$
$$\dots = \partial_{f(2)v} G(a;b) = G'_{f(2)v}(a;b)$$

#### Exemple d'utilisation 3

```
\displaystyle
\partialfrac[2]{f}{x | y}
= \partialfrac*[2]{%
  \left( \frac{1}{\cos(x y)} \right)%
}{x | y}$
```

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2}{\partial x \partial y} \left( \frac{1}{\cos(xy)} \right)$$

## 9.6 Calcul intégral

### 9.6.1 L'opérateur crochet – 1<sup>ère</sup> version

#### Exemple d'utilisation 1

```
\displaystyle
\int_{a}^{b} f(x) \, \mathrm{d}x
= \hook{F(x)}{a}{b}$

\hook{F(x)}{a}{b}
= F(b) - F(a)$
```

$$\int_a^b f(x) \, \mathrm{d}x = [F(x)]_a^b$$
$$[F(x)]_a^b = F(b) - F(a)$$

#### Exemple d'utilisation 2

Par défaut, les crochets s'étirent verticalement si besoin, mais si cela vous dérange, vous pouvez faire appel à la version étoilée de la macro comme dans l'exemple suivant

```

 $\hook{\frac{x - 1}{5 + x^2}}{a}{b}
= \hook*{\frac{x - 1}{5 + x^2}}{a}{b}$$ 
```

$$\left[ \frac{x-1}{5+x^2} \right]_a^b = \left[ \frac{x-1}{5+x^2} \right]_a^b$$

### 9.6.2 L'opérateur crochet – 2<sup>de</sup> version

#### Exemple d'utilisation 1

```

 $\vhook{F(x)}{a}{b}$ 
= \hook{F(x)}{a}{b}$.

```

$$F(x)|_a^b = [F(x)]_a^b.$$

#### Exemple d'utilisation 2

Tout comme avec la première version de l'opérateur crochet, vous pouvez utiliser une version étoilée pour empêcher l'étirement verticalement du trait vertical. Voici un exemple.

```

 $\vhook{\frac{x - 1}{5 + x^2}}{a}{b}
= \vhook*{\frac{x - 1}{5 + x^2}}{a}{b}$$ 
```

$$\left. \frac{x-1}{5+x^2} \right|_a^b = \left. \frac{x-1}{5+x^2} \right|_a^b$$

### 9.6.3 Intégrales multiples

Le package réduit les espacements entre des symboles  $\int$  successifs. Voici un exemple.

```

 $\int \int \int F(x;y;z) \dd{x} \dd{y} \dd{z}
= \int_{a}^{b} \int_{c}^{d} \int_{e}^{f} F(x;y;z) \dd{x} \dd{y} \dd{z}$$ 
```

$$\iiint F(x;y;z) \, dx \, dy \, dz = \int_a^b \int_c^d \int_e^f F(x;y;z) \, dx \, dy \, dz$$

**Remarque.** Par défaut, L<sup>A</sup>T<sub>E</sub>X affiche  $\int \int \int F(x;y;z) \, dx \, dy \, dz = \int_a^b \int_c^d \int_e^f F(x;y;z) \, dx \, dy \, dz$ . Nous avons obtenu ce résultat en utilisant `\stdint` qui est l'opérateur proposé de façon standard par L<sup>A</sup>T<sub>E</sub>X.

## 9.7 Tableaux de variation et de signe

### Comment ça marche ?

Tout le boulot est fait par le package `tkz-tab` auquel on impose le choix d'une pointe de flèche plus visible. Nous vous demandons donc de vous reporter à la documentation de `tkz-tab` pour savoir comment s'y prendre.

## Un exemple de tableaux de signes

```
\begin{tikzpicture}
  \tkzTabInit{$x$ / 1 , $\cos(x)$ / 1}%
    {$0$, $\frac{\pi}{2}$, $\pi$}

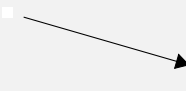
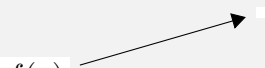
  \tkzTabLine{ , + , z , - , }
\end{tikzpicture}
```

$x$	0	$\frac{\pi}{2}$	$\pi$
$\cos(x)$	+	0	−

## Un exemple de tableaux de variation

```
\begin{tikzpicture}
  \tkzTabInit{$x$ / 1 , $f(x)$ / 1.5}%
    {$-\infty$, $p$, $+\infty$}

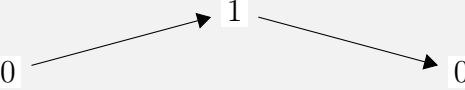
  \tkzTabVar{+ / , - / $f(p)$ , + / }
\end{tikzpicture}
```

$x$	$-\infty$	$p$	$+\infty$
$f(x)$		$f(p)$	

## Un exemple de tableaux de variation avec une dérivée

```
\begin{tikzpicture}
  \tkzTabInit{$x$ / 1 , $\cos(x)$ / 1, $\sin(x)$ / 1.5}%
    {$0$, $\frac{\pi}{2}$, $\pi$}

  \tkzTabLine{ , + , z , - , }
  \tkzTabVar{- / 0 , + / 1 , - / 0}
\end{tikzpicture}
```

$x$	0	$\frac{\pi}{2}$	$\pi$
$\cos(x)$	+	0	−
$\sin(x)$		1	0

## 9.8 Comparaison asymptotique de suites et de fonctions

### 9.8.1 Les notations $\mathcal{O}$ et $\mathcal{o}$

#### Exemple d'utilisation 1

Les notations suivantes sont dues à Landau.

$\backslash\bigO{\hspace{-1pt}}$ ou\backslash\smallO{\hspace{-1pt}}$$	$\mathcal{O}$ ou $\mathcal{o}$
---	--------------------------------

#### Exemple d'utilisation 2

$\backslash\bigO{x} \neq \backslash\smallO{x}$ oue^{t + \backslash\smallO{t}} = e^{\backslash\bigO{t}}$$	$\mathcal{O}(x) \neq \mathcal{o}(x)$ ou $e^{t+\mathcal{o}(t)} = e^{\mathcal{O}(t)}$
--	---

### 9.8.2 La notation $\Omega$

#### Exemple d'utilisation 1

La notation suivante est due à Hardy et Littlewood.

$\backslash\bigomega{\hspace{-1pt}}$$	$\Omega$
---------------------------------------	----------

#### Exemple d'utilisation 2

Dans l'exemple suivant,  $f(n) = \Omega(g(n))$  signifie :  $\exists(m, n_0)$  tel que  $n \geq n_0$  implique  $f(n) \geq mg(n)$ .

$f(n) = \backslash\bigomega{g(n)}$$	$f(n) = \Omega(g(n))$
-------------------------------------	-----------------------

### 9.8.3 La notation $\Theta$

#### Exemple d'utilisation 1

$\backslash\bigtheta{\hspace{-1pt}}$$	$\Theta$
---------------------------------------	----------

#### Exemple d'utilisation 2

Dans l'exemple suivant,  $f(n) = \Theta(g(n))$  signifie :  $\exists(m, M, n_0)$  tel que  $mg(n) \leq f(n) \leq Mg(n)$  dès que  $n \geq n_0$ .

$f(n) = \backslash\bigtheta{g(n)}$$	$f(n) = \Theta(g(n))$
-------------------------------------	-----------------------



## 10 Probabilité

### 10.1 Probabilité « simple »

Un exemple type

```
\proba{A}$ ou  
\proba[P]{A}$
```

$p(A)$  ou  $P(A)$

### 10.2 Probabilité conditionnelle

Un exemple type

La 1<sup>re</sup> notation, qui est devenue standard, permet de comprendre l'ordre des arguments.

```
\probacond{B}{A}  
=  
\probacond*{B}{A}  
=  
\probacond**{B}{A}  
=  
\dprobacond**{B}{A}$
```

$$p_B(A) = p(A | B) = \frac{p(A \cap B)}{p(B)} = \frac{p(A \cap B)}{p(B)}$$

Choisir le nom de la probabilité

```
\probacond[P]{B}{A}  
=  
\probacond*[P]{B}{A}  
=  
\probacond**[P]{B}{A}  
=  
\dprobacond**[P]{B}{A}$
```

$$P_B(A) = P(A | B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

### 10.3 Arbres pondérés

Que se passe-t-il en coulisse ?

Le gros du travail est fait par le package **forest** qui utilise **TikZ**. Ceci permet de faire des choses sympathiques comme dans le 2<sup>e</sup> exemple ci-dessous.

Un exemple type

Dans le code suivant l'environnement **probatree** utilise en coulisse celui nommé **forest** du package **forest**. Des réglages spécifiques sont faits pour obtenir le résultat ci-après. À cela s'ajoutent les styles spéciaux **pweight**, **apweight** et **bpweight** qui facilitent l'écriture des pondérations sur les branches<sup>6</sup>.

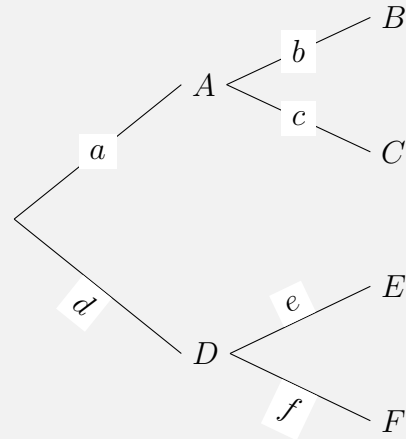
---

6. **pweight** vient de « *probability* » et « *weight* » soit « *probabilité* » et « *poids* » en anglais. Quant à **a** et **b** au début de **apweight** et **bpweight** respectivement, ils viennent de « *above* » et « *below* » soit « *dessus* » et « *dessous* » en anglais.

```

\begin{probatree}
[
  [$A$, pweight = $a$
    [$B$, pweight = $b$]
    [$C$, pweight = $c$]
  ]
  [$D$, bpweight = $d$
    [$E$, apweight = $e$]
    [$F$, bpweight = $f$]
  ]
]
\end{probatree}

```



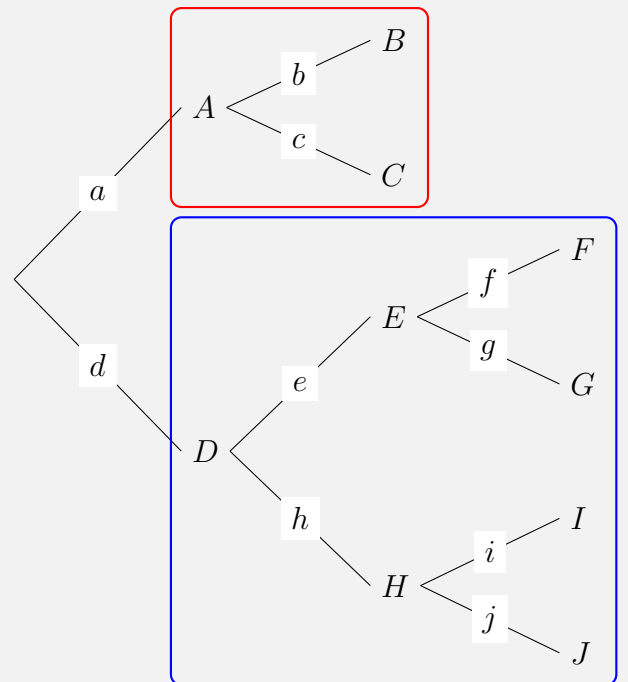
### Un exemple décoré facilement

Via la clé `frame`, il est très aisé d'encadrer un sous-arbre comme le montre l'exemple suivant. Dans l'exemple ci-après nous utilisons la bidouille `{},s sep = 1.3cm` qui évite que les cadres se superposent.

```

\begin{probatree}
[{},s sep = 1.3cm
  [$A$, pweight=$a$, frame=red
    [$B$, pweight=$b$]
    [$C$, pweight=$c$]
  ]
  [$D$, pweight=$d$, frame=blue
    [$E$, pweight=$e$
      [$F$, pweight=$f$]
      [$G$, pweight=$g$]
    ]
    [$H$, pweight=$h$
      [$I$, pweight=$i$]
      [$J$, pweight=$j$]
    ]
  ]
]
\end{probatree}

```



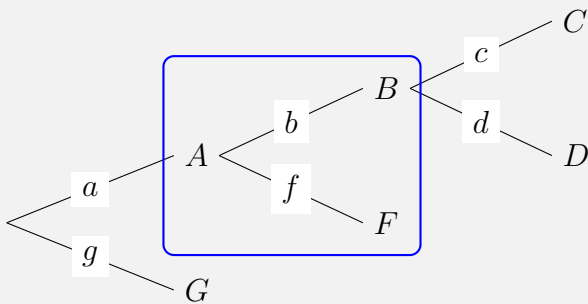
### Un exemple décoré à la main

En utilisant la machinerie de `TikZ` il est facile de décorer un arbre de probabilité comme ci-dessous où le cadre s'appuie sur trois noeuds nommés. Notons que cet exemple est tout simplement infaisable avec la clé `frame`.

```

\begin{probatree}
[
  [$A$, pweight = $a$, name = left
    [$B$, pweight = $b$, name = topright
      [$C$, pweight = $c$]
      [$D$, pweight = $d$]
    ]
  [$F$, pweight = $f$, name = bottomright]
]
[$G$, pweight = $g$]
]
\node[draw = blue,
  thick,
  rounded corners,
  fit = (left)(topright)(bottomright)] {};
\end{probatree}

```



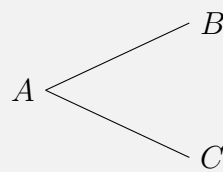
### Un exemple de poids cachés partout

On peut cacher tous les poids via l'environnement étoilé `probatree*` sans avoir à retaper un arbre où les pondérations ont déjà été indiquées.

```

\begin{probatree*}
[$A$, pweight = $a$
  [$B$, pweight = $b$]
  [$C$, pweight = $c$]
]
\end{probatree*}

```



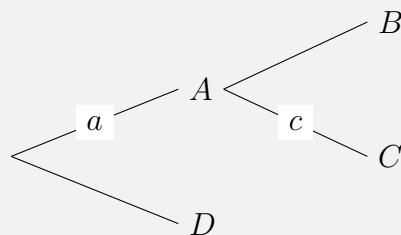
### Un exemple de poids cachés localement

Pour ne cacher que certains poids, il faudra utiliser, à la main, le style `pweight*` comme dans l'exemple ci-dessous.

```

\begin{probatree}
[
  [$A$, pweight = $a$
    [$B$, pweight* = $b$]
    [$C$, pweight = $c$]
  ]
  [$D$, pweight* = $d$]
]
\end{probatree}

```



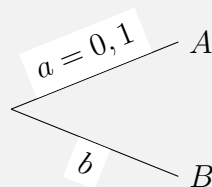
## Une astuce pour un signe = et une virgule dans les étiquettes

Vous ne pouvez pas utiliser directement un signe = ou une virgule dans les étiquettes des branches. L'astuce pour contourner cette limitation consiste juste à mettre le contenu de l'étiquette dans des accolades.

```

\begin{probatree}
[
  [$A$, apweight = {$a = 0,1$}]
  [$B$, bpweight = $b$]
]
\end{probatree}

```



# 11 Arithmétique

## 11.1 Opérateurs de base

Pour des raisons d'expressivité des codes L<sup>A</sup>T<sub>E</sub>X, les opérateurs binaires `\divides`, `\notdivides` et `\modulo` ont été ajoutés comme alias respectifs de `\mid`, `\nmid` et `\bmod` qui sont proposés par le package `amssymb`.

```

$10 \divides 150 = 10 \mid 150$ ,
$10 \notdivides 154 = 10 \not\mid 154$

$a \equiv b \pmod{p}$
\iff
p \divides (a - b)$ .

```

$10 \mid 150 = 10 \mid 150$  ,  $10 \nmid 154 = 10 \not\mid 154$   
 $a \equiv b \pmod{p} \iff p \mid (a - b)$ .

## 11.2 Fractions continuées

### 11.2.1 Fractions continuées standard

#### Exemple d'utilisation

Dans l'exemple suivant, la notation en ligne semble être due à Alfred Pringsheim. La notation à gauche utilise toujours le maximum d'espace pour améliorer la lisibilité.

```
$ \contfrac{u_0 | u_1 | u_2 | \dots | u_n}
= \contfrac*{u_0 | u_1 | u_2 | \dots | u_n}$
```

$$u_0 + \frac{1}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}} = u_0 + \left| \frac{1}{u_1} \right| + \left| \frac{1}{u_2} \right| + \left| \frac{1}{\dots} \right| + \left| \frac{1}{u_n} \right|$$

### 11.2.2 Fractions continuées généralisées

#### Exemple d'utilisation

Voici comment écrire une fraction continuée généralisée.

```
$\displaystyle
\contfracgene{a | b | c | d | e | f | \dots | y | z}
= \contfracgene*{a | b | c | d | e | f | \dots | y | z}$
```

$$a + \frac{b}{c + \frac{d}{e + \frac{f}{\dots + \frac{y}{z}}}} = a + \left| \frac{b}{c} \right| + \left| \frac{d}{e} \right| + \left| \frac{f}{\dots} \right| + \left| \frac{y}{z} \right|$$

### 11.2.3 Comme une fraction continuée isolée

#### Exemple d'utilisation

La raison d'être de la macro ci-dessous vient juste de son usage en interne.

```
$\singlecontfrac{a}{b}$
pour les fous\dots :-)
```

$$\left| \frac{a}{b} \right| \text{ pour les fous... :-)}$$

### 11.2.4 L'opérateur $\mathcal{K}$

#### Exemple d'utilisation 1

La notation suivante est proche de celle qu'utilisait Carl Friedrich Gauss.

```

 $\contfracope_{k=1}^n (b_k:c_k)
= \cfrac{b_1}{c_1 + \frac{b_2}{c_2 + \frac{b_3}{\dots + \frac{b_n}{c_n}}}}$ 

```

$$\mathcal{K}_{k=1}^n (b_k : c_k) = \frac{b_1}{c_1 + \frac{b_2}{c_2 + \frac{b_3}{\dots + \frac{b_n}{c_n}}}}$$

**Remarque.** La lettre  $\mathcal{K}$  vient de "kettenbruch" qui signifie "fraction continuée" en allemand.

### Exemple d'utilisation 2

```

 $u_0 + \contfracope_{k=1}^n (1:u_k)
= \contfrac{u_0}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}}$ 

```

$$u_0 + \mathcal{K}_{k=1}^n (1 : u_k) = u_0 + \frac{1}{u_1 + \frac{1}{u_2 + \frac{1}{\dots + \frac{1}{u_n}}}}$$

## 12 Algèbre

### 12.1 Polynômes, séries formelles et compagnie

#### 12.1.1 Polynômes et fractions polynômiales

##### Exemple d'utilisation 1 : Polynômes

<pre> <math display="style"> \setpoly{\RR}{X} \text{ ou } \setpoly{\RR}{X \mid Y \mid Z} </math> </pre>	$\mathbb{R}[X]$ ou $\mathbb{R}[X;Y;Z]$
---	--

##### Exemple d'utilisation 2 : Fractions polynômiales

<pre> <math display="style"> \setpolyfrac{\QQ}{T} \text{ ou } \setpolyfrac{\QQ}{\{S_1 \mid S_2 \mid \dots \mid S_k\}} </math> </pre>	$\mathbb{Q}(T)$ ou $\mathbb{Q}(S_1;S_2;\dots;S_k)$
--	--

### 12.1.2 Séries formelles et leurs corps de fractions

#### Exemple d'utilisation 1 : Séries formelles

$\text{\texttt{\$setserie\{CC\}\{X\}}}$ ou $\text{\texttt{\$setserie\{CC\}\{T   O   P\}}}$	$\mathbb{C}[[X]]$ ou $\mathbb{C}[[T; O; P]]$
---	--

#### Exemple d'utilisation 2 : Corps des fractions de séries formelles

$\text{\texttt{\$setseriefrac\{ZZ\}\{X\}}}$ ou $\text{\texttt{\$setseriefrac\{ZZ\}\{Z   T   O   P\}}}$	$\mathbb{Z}((X))$ ou $\mathbb{Z}((Z; T; O; P))$
---	---

### 12.1.3 Polynômes de Laurent et séries formelles de Laurent

#### Exemple d'utilisation 1 : Polynômes de Laurent

Ci-dessous, la notation  $\mathbb{R}\{X_1; X_2\}$  n'est pas standard.

$\text{\texttt{\$setpolylaurent\{RR\}\{X\}}}$ $\text{\texttt{= \setpoly\{RR\}\{X   X^{-1}\}}}$ $\text{\texttt{\$setpolylaurent\{RR\}\{X_1   X_2\}}}$ $\text{\texttt{= \setpoly\{RR\}\{X_1   X_1^{-1}   X_2   X_2^{-1}\}}}$	$\mathbb{R}\{X\} = \mathbb{R}[X; X^{-1}]$ $\mathbb{R}\{X_1; X_2\} = \mathbb{R}[X_1; X_1^{-1}; X_2; X_2^{-1}]$
---	--

#### Exemple d'utilisation 2 : Séries formelles de Laurent

Ci-dessous, la notation  $\mathbb{Q}\{\{X_1; X_2\}\}$  n'est pas standard.

$\text{\texttt{\$setserielaurent\{QQ\}\{X\}}}$ $\text{\texttt{= \setserie\{QQ\}\{X   X^{-1}\}}}$ $\text{\texttt{\$setserielaurent\{QQ\}\{X_1   X_2\}}}$ $\text{\texttt{= \setserie\{QQ\}\{X_1   X_1^{-1}   X_2   X_2^{-1}\}}}$	$\mathbb{Q}\{\{X\}\} = \mathbb{Q}[[X; X^{-1}]]$ $\mathbb{Q}\{\{X_1; X_2\}\} = \mathbb{Q}[[X_1; X_1^{-1}; X_2; X_2^{-1}]]$
---	--

## 12.2 Matrices

### Comment ça marche ?

Tout le boulot est fait par le package `nicematrix` auquel on impose l'option `transparent`. Veuillez vous reporter à la documentation de `nicematrix` pour savoir comment s'y prendre.

### Exemple 1 tiré de la documentation de nicematrix

```

 $\begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$ 

```

$$\begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

### Exemple 2 tiré de la documentation de nicematrix

```

 $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ 

```

$\text{\tikz[remember picture, overlay] \draw[red] (mymatrix-2-2) circle (2.5mm);}$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

### Exemple 3 adapté de la documentation de nicematrix

```

 $\left( \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{array} \right)$ 

```

$$\left( \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{array} \right)$$



#### Exemple 4 proposé par l’auteur de nicematrix suite à une discussion par mail

```
% Besoin du package ‘‘ifthen’’.
\newcommand\aij{%
  a_{\arabic{iRow}\arabic{jCol}}%
}

$\begin{bNiceArray}%
  {*{5}{>{%
    \ifthenelse{\value{iRow}>0}{\aij}{}%
  }C}}%
  [first-col,
  first-row,
  code-for-first-row
  = \mathbf{\arabic{jCol}},
  code-for-first-col
  = \mathbf{\arabic{iRow}}]
  & & & & \\
  & & & & \\
  & & & & \\
\end{bNiceArray}$
```

$$\begin{matrix} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{1} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \mathbf{2} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{matrix}$$

#### Exemple 5 avec des calculs automatiques

```
\newcounter{cntaij}
\newcommand\aij{%
  \setcounter{cntaij}{\value{iRow}}%
  \addtocounter{cntaij}{\value{jCol}}%
  \addtocounter{cntaij}{-1}%
  \arabic{cntaij}%
}
Si $a_{ij} = i + j - 1$ alors
$(a_{ij})_{1 \leq i \leq 3, 1 \leq j \leq 5}$
=
\begin{bNiceArray}{*{5}{>{\aij}C}}
  & & & & \\
  & & & & \\
  & & & & \\
\end{bNiceArray}$
```

Si  $a_{ij} = i + j - 1$  alors

$$(a_{ij})_{1 \leq i \leq 3, 1 \leq j \leq 5} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

## 13 Historique

Nous ne donnons ici qu'un très bref historique récent de `lymath` à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier `change-log` : voir le code source de `lymath` sur [github](#).

### 2020-06-08 Nouvelle version mineure 0.7.0-beta.

- En analyse, il y a eu les changements suivants.
  - Pour éviter des conflits avec d'autres packages, les renommages suivants ont été faits.
    - `\ch` et `\ach` sont devenus `\fch` et `\afch` où `f` est pour `f`-rench.
    - `\sh` et `\ash` sont devenus `\fsh` et `\afsh`.
    - `\th` et `\ath` sont devenus `\fth` et `\afth`.
  - Les macros `\acosh`, `\asinh` et `\atanh` ont été ajoutées.
  - `\derpar` et `\derpar*` servent à rédiger des dérivées avec des parenthèses extensibles. En coulisse, `\derpow` et `\derpow*` sont appelées. Pour utiliser des parenthèses non extensibles, on passera par `\sderpar` et `\sderpar*` où `s` est pour `s`-mall.
  - Ajout de `\stdint` pour rendre public l'opérateur intégral proposé par défaut par L<sup>A</sup>T<sub>E</sub>X.
- En géométrie, il y a eu une suppression et trois ajouts.
  - La macro `\pts` a été supprimée car sans signification sémantique puisqu'un point peut être nommé avec deux lettres.
  - Les macros `\gline` et `\pgline` servent à indiquer des droites.
  - La macro `\hgline`, avec `h` pour `h`-alf, est pour les demi-droites.
  - La macro `\segment` est utile pour les segments.
- En logique, voici les améliorations apportées.
  - Pour les inégalités, on peut maintenant utiliser le décorateur `plot`.
  - Ajout des versions négatives des opérateurs logiques verticaux.
- Pour les probabilités, il y a eu les modifications ci-après.
  - Ajout de `\proba` pour écrire des probabilités.
  - Le comportement de `\probacond` a été modifié pour le rendre plus logique.

### 2019-10-21 Nouvelle version sous-mineure 0.6.3-beta.

- Pour les intervalles, `\CSinterval` a été déplacée dans le package `lyalgo` disponible à l'adresse <https://github.com/bc-latex/ly-algo>.
- En logique, il y a eu les modifications suivantes.
  - `\eqdef**` a été supprimé. Voir la macro `\Store*` du package `lyalgo`.
  - Différentes versions de l'opérateur  $\exists$  via `\existssone` et `\existmulti` avec leurs versions négatives `\nexistsone` et `\nexistmulti`.
  - Deux nouvelles macros `\eqplot` et `\eqappli` pour indiquer une équation de courbe et l'application d'une identité à des variables. Ceci s'accompagne de l'ajout des macros `\textopplot` et `\textopappli`.

- Ajout des formes négatives `\niff`, `\nimplies` et `\nliesimp`.
- Les décorations `cons`, `appli` et `choice` sont utilisables avec les opérateurs `\iff`, `\implies` et `\liesimp` et leurs formes négatives.
- Une macro `\textoptest` a été ajoutée afin de rendre personnalisable tous les textes décorant les symboles.
- En analyse, il y a eu les renommages suivants.
  - `\hypergeo` est devenu `\seqhypergeo`.
  - `\suprageo` est devenu `\seqsuprageo`.
- En géométrie, `\notparallel` est devenu `\nparallel`.

#### 2019-10-14 Nouvelle version sous-mineure 0.6.2-beta.

- En algèbre, il y a eu les renommages ci-dessous qui avaient été oubliés.
  - `\polyset` est devenu `\setpoly`.
  - `\polyfracset` est devenu `\setpolyfrac`.
  - `\serieset` est devenu `\setserie`.
  - `\seriefracset` est devenu `\setseriefrac`.
  - `\polylaurentset` est devenu `\setpolylaurent`.
  - `\serielaurentset` est devenu `\setserielaurent`.

#### 2019-10-13 Nouvelle version sous-mineure 0.6.1-beta.

- En logique, la macro `\explain` possède maintenant un argument optionnel pour indiquer l'espacement avant le symbole. Les macros obsolètes `\explain*` et `\textexplainspacebefore` ont été supprimées.
- En probabilité, voici ce qui a évolué.
  - Les macros `\probacond` et `\probacond*` n'ont plus d'argument optionnel. Pour obtenir l'écriture fractionnaire, il faut utiliser `\probacond**` ou `\dprobacond**`.
  - Les environnements `probatree` et `probatree*` ont trois nouvelles clés. La clé `frame` permet d'encadrer un sous-arbre, et les clés `apweight` et `bpweight` permettent d'écrire des poids dessus/dessous une branche.
- Pour les ensembles, il y a eu les renommages suivants par souci de cohérence.
  - `\algeset` est devenu `\setalge`.
  - `\geoset` est devenu `\setgeo`.
  - `\geneset` est devenu `\setgene`.
  - `\probaset` est devenu `\setproba`.
  - `\specialset` est devenu `\setspecial`.

#### 2019-10-10 Nouvelle version mineure 0.6.0-beta.

- Des nouveaux outils spécifiques aux probabilités.
  - Les macros `\probacond` et `\probacond*` servent à écrire des probabilités conditionnelles.
  - Les environnements `probatree` et `probatree*` simplifient la production d'arbres probabilistes pondérés ou non.

- En géométrie, la macro `\notparallel` a été rajoutée.
- Un nouveau type d'intervalle pour l'informatique théorique via la macro `\CSinterval` afin d'obtenir quelque chose comme `a..b`.
- En logique, il y a deux nouvelles macros sémantiques `\neqid` et `\eqchoice`.

#### 2019-09-27 Nouvelle version mineure 0.5.0-beta.

- Ajout des macros `\dsum` et `\dprod` qui sont vis à vis de `\sum` et `\prod` des équivalents de `\dfrac` pour `\frac`.
- En arithmétique, ajout des opérateurs `\divides`, `\notdivides` et `\modulo`.
- En géométrie, une nouvelle macro et un opérateur modifié.
  - `\pts` permet d'indiquer plusieurs points.
  - `\parallel` utilise des obliques pour symboliser le parallélisme au lieu de barres verticales.
- En logique, il y a les nouveautés suivantes.
  - La version doublement étoilée `\eqdef**` donne une deuxième écriture symbolique d'un symbole égal de type définition (*cette notation vient du langage B*).
  - Ajout de `\liesimp` comme alias de `\Longleftarrow`.
  - Les macros `\vimplies`, `\viff` et `\vliesimp` sont des versions verticales de `\implies`, `\iff` et `\liesimp`.
  - Comme pour les égalités, il existe les macros `\impliestest`, `\iffhyp` ... etc.

#### 2019-09-06 Nouvelle version mineure 0.4.0-beta.

- Dans « *Logique et fondements* », différents types de signes d'inéquation et de non égalité pour des cas de test, d'hypothèse faite et de condition à vérifier.
- Intégration du package `tkz-tab` pour rédiger des tableaux de variations et de signes.
- Intégration du package `nicematrix` pour écrire des matrices.

## 14 Toutes les fiches techniques

### 14.1 Deux séparateurs d'arguments par défaut

`\lymathsep <macro>` (Sans argument)  
`\lymathsubsep <macro>` (Sans argument)

### 14.2 Quelques gestions d'espaces

#### 14.2.1 Fractions

`\frac <macro>` (2 Arguments)  
`\dfrac <macro>` (2 Arguments)  
`\stdfrac <macro>` (2 Arguments)  
`\stddfraction <macro>` (2 Arguments)

- Argument 1: le numérateur.
- Argument 2: le dénominateur.

#### 14.2.2 Racines n-ièmes

`\sqrt <macro>` [1 Option] (1 Argument)  
`\stdsqrt <macro>` [1 Option] (1 Argument)

- Option: l'indice à indiquer pour une racine n-ième.
- Argument: le radicande, c'est à dire ce qui sera écrit sous le radical.

#### 14.2.3 Sommes et produits

Les macros suivantes sans argument ont un comportement spécifique vis à vis des mises en index et en exposant.

---

`\dprod <macro>` (Sans argument)  
`\dsum <macro>` (Sans argument)

### 14.3 Logique et fondements

#### 14.3.1 Les textes pour les opérateurs de « comparaison algébrique » et de logique

`\textopappli <macro>` (Sans argument)  
`\textopchoice <macro>` (Sans argument)  
`\textopcond <macro>` (Sans argument)  
`\textopcons <macro>` (Sans argument)  
`\textopdef <macro>` (Sans argument)  
`\textophyp <macro>` (Sans argument)  
`\textopid <macro>` (Sans argument)  
`\textopplot <macro>` (Sans argument)  
`\textoptest <macro>` (Sans argument)

### 14.3.2 Les opérateurs de « comparaison algébrique »

```
\eqappli <macro> (Sans argument)
\eqchoice <macro> (Sans argument)
\eqcond <macro> (Sans argument)
\eqcons <macro> (Sans argument)
\eqdef <macro> (Sans argument)
\eqdef* <macro> (Sans argument)
\eqhyp <macro> (Sans argument)
\eqid <macro> (Sans argument)
\eqid* <macro> (Sans argument)
\eqplot <macro> (Sans argument)
\eqtest <macro> (Sans argument)
```

---

```
\neqappli <macro> (Sans argument)
\neqchoice <macro> (Sans argument)
\neqcond <macro> (Sans argument)
\neqcons <macro> (Sans argument)
\neqhyp <macro> (Sans argument)
\neqid <macro> (Sans argument)
\neqtest <macro> (Sans argument)
```

---

```
\lappli <macro> (Sans argument)
\lchoice <macro> (Sans argument)
\lcond <macro> (Sans argument)
\lcons <macro> (Sans argument)
\lhyp <macro> (Sans argument)
\lplot <macro> (Sans argument)
\ltest <macro> (Sans argument)
```

---

```
\gappli <macro> (Sans argument)
\gchoice <macro> (Sans argument)
\gcond <macro> (Sans argument)
\gcons <macro> (Sans argument)
\ghyp <macro> (Sans argument)
\gplot <macro> (Sans argument)
\gtest <macro> (Sans argument)
```

---

```
\leqappli <macro> (Sans argument)
\leqchoice <macro> (Sans argument)
\leqcond <macro> (Sans argument)
\leqcons <macro> (Sans argument)
\leqhyp <macro> (Sans argument)
\leqplot <macro> (Sans argument)
\leqtest <macro> (Sans argument)
```

### 14.3.3 Les opérateurs de logique

`\iff <macro> (Sans argument)`  
`\iffappli <macro> (Sans argument)`  
`\iffchoice <macro> (Sans argument)`  
`\iffcond <macro> (Sans argument)`  
`\iffcons <macro> (Sans argument)`  
`\iffhyp <macro> (Sans argument)`  
`\ifftest <macro> (Sans argument)`

---

`\notiff <macro> (Sans argument)`  
`\notiffappli <macro> (Sans argument)`  
`\notiffchoice <macro> (Sans argument)`  
`\notiffcond <macro> (Sans argument)`  
`\notiffcons <macro> (Sans argument)`  
`\notiffhyp <macro> (Sans argument)`  
`\notifftest <macro> (Sans argument)`

---

`\implies <macro> (Sans argument)`  
`\impliesappli <macro> (Sans argument)`  
`\implieschoice <macro> (Sans argument)`  
`\impliescond <macro> (Sans argument)`  
`\impliescons <macro> (Sans argument)`  
`\implieshyp <macro> (Sans argument)`  
`\impliestest <macro> (Sans argument)`

---

`\notimplies <macro> (Sans argument)`  
`\notimpliesappli <macro> (Sans argument)`  
`\notimplieschoice <macro> (Sans argument)`  
`\notimpliescond <macro> (Sans argument)`  
`\notimpliescons <macro> (Sans argument)`  
`\notimplieshyp <macro> (Sans argument)`  
`\notimpliestest <macro> (Sans argument)`

---

`\liesimp <macro> (Sans argument)`  
`\liesimpappli <macro> (Sans argument)`  
`\liesimpchoice <macro> (Sans argument)`  
`\liesimpcond <macro> (Sans argument)`  
`\liesimpcons <macro> (Sans argument)`  
`\liesimphyp <macro> (Sans argument)`  
`\liesimptest <macro> (Sans argument)`

---

`\notliesimp <macro> (Sans argument)`  
`\notliesimpappli <macro> (Sans argument)`  
`\notliesimpchoice <macro> (Sans argument)`  
`\notliesimpcond <macro> (Sans argument)`  
`\notliesimpcons <macro> (Sans argument)`

`\notliesimphyp` <macro> (Sans argument)  
`\notliesimptest` <macro> (Sans argument)

#### 14.3.4 Les opérateurs de logique « verticaux »

`\viff` <macro> (Sans argument)  
`\notviff` <macro> (Sans argument)  
`\vimplies` <macro> (Sans argument)  
`\notvimplies` <macro> (Sans argument)  
`\vliesimp` <macro> (Sans argument)  
`\notvliesimp` <macro> (Sans argument)

#### 14.3.5 Les textes pour détailler un raisonnement

`\textexplainleft` <macro> (Sans argument)  
`\textexplainright` <macro> (Sans argument)  
`\textexplainspacein` <macro> (Sans argument)

#### 14.3.6 Détailler un raisonnement

`explain` <env> [1 Option]

— Option: le symbole à utiliser dans l’environnement, la valeur par défaut étant = .

---

`\explnext` <macro> [1 Option] (1 Argument)

— Option: le symbole à utiliser pour une explication, la valeur par défaut étant celle du symbole de l’environnement `explain` où `\explnext` est utilisé.

— Argument: le texte de l’explication qui peut être vide si aucune explication n’est à afficher.

#### 14.3.7 Des versions alternatives du quantificateur $\exists$

`\existmulti` <macro> (1 Argument)  
`\nexistmulti` <macro> (1 Argument)

— Argument 1: une écriture mathématique servant à préciser la portée du quantificateur.

---

`\existstone` <macro> (Sans argument)  
`\nexiststone` <macro> (Sans argument)

### 14.4 Ensembles et applications

#### 14.4.1 Ensembles versus accolades

`\setgene` <macro> (1 Argument)  
`\setgene*` <macro> (1 Argument)

— Argument: la définition de l’ensemble.



### 14.4.2 Ensembles pour la géométrie

`\setgeo <macro> (1 Argument)`

— Argument: un seul caractère ASCII indiquant un ensemble géométrique.

---

`\setgeo* <macro> (2 Arguments)`

— Argument 1: un seul caractère ASCII indiquant  $\mathcal{U}$  dans le nom  $\mathcal{U}_d$  d'un ensemble géométrique.

— Argument 2: un texte donnant  $d$  dans le nom  $\mathcal{U}_d$  d'un ensemble géométrique.

### 14.4.3 Ensembles probabilistes

`\setproba <macro> (1 Argument)`

— Argument: un seul caractère ASCII majuscule indiquant un ensemble probabiliste.

---

`\setproba* <macro> (2 Arguments)`

— Argument 1: un seul caractère ASCII majuscule indiquant  $\mathcal{U}$  dans le nom  $\mathcal{U}_d$  d'un ensemble probabiliste.

— Argument 2: un texte donnant  $d$  dans le nom  $\mathcal{U}_d$  d'un ensemble probabiliste.

### 14.4.4 Ensembles pour l'algèbre générale

`\setalge <macro> (1 Argument)`

— Argument: soit l'une des lettres  $h$  et  $k$ , soit un seul caractère ASCII majuscule indiquant un ensemble de type anneau ou corps.

---

`\setalge* <macro> (2 Arguments)`

— Argument 1: un seul caractère ASCII indiquant  $\mathbb{U}$  dans le nom  $\mathbb{U}_d$  d'un ensemble de type anneau ou corps.

— Argument 2: un texte donnant  $d$  dans le nom  $\mathbb{U}_d$  d'un ensemble de type anneau ou corps.

### 14.4.5 Ensembles classiques

`\NN <macro> (Sans argument)`

`\NNs <macro> (Sans argument)`

---

`\PP <macro> (Sans argument)`

---

`\ZZ <macro> (Sans argument)`

`\ZZn <macro> (Sans argument)`

`\ZZp <macro> (Sans argument)`

`\ZZs <macro> (Sans argument)`

`\ZZsn <macro> (Sans argument)`

`\ZZsp <macro> (Sans argument)`

---

`\DD <macro> (Sans argument)`  
`\DDn <macro> (Sans argument)`  
`\DDp <macro> (Sans argument)`  
`\DDs <macro> (Sans argument)`  
`\DDsn <macro> (Sans argument)`  
`\DDsp <macro> (Sans argument)`

---

`\QQ <macro> (Sans argument)`  
`\QQn <macro> (Sans argument)`  
`\QQp <macro> (Sans argument)`  
`\QQs <macro> (Sans argument)`  
`\QQsn <macro> (Sans argument)`  
`\QQsp <macro> (Sans argument)`

---

`\RR <macro> (Sans argument)`  
`\RRn <macro> (Sans argument)`  
`\RRp <macro> (Sans argument)`  
`\RRs <macro> (Sans argument)`  
`\RRsn <macro> (Sans argument)`  
`\RRsp <macro> (Sans argument)`

---

`\CC <macro> (Sans argument)`  
`\CCs <macro> (Sans argument)`

---

`\HH <macro> (Sans argument)`  
`\HHs <macro> (Sans argument)`

---

`\OO <macro> (Sans argument)`  
`\OOS <macro> (Sans argument)`

#### 14.4.6 Des suffixes à la carte

`\setspecial <macro> (2 Arguments)`  
`\setspecial* <macro> (2 Arguments)`

- Argument 1: l'ensemble à "suffixer".
- Argument 2: l'un des suffixes `n`, `p`, `s`, `sn` ou `sp`.

#### 14.4.7 Intervalles réels - Notation française (?)

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

---

`\intervalCO <macro> (2 Arguments)`  
`\intervalCO* <macro> (2 Arguments)`

- Argument 1: borne inférieure  $a$  de l'intervalle  $[a; b[$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $[a; b[$ .

---

`\intervalC <macro> (2 Arguments)`

`\intervalC* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $[a; b]$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $[a; b]$ .

---

`\intervalO <macro> (2 Arguments)`

`\intervalO* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $]a; b[$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $]a; b[$ .

---

`\intervalOC <macro> (2 Arguments)`

`\intervalOC* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $]a; b]$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $]a; b]$ .

#### 14.4.8 Intervalles réels - Notation américaine

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

---

`\intervalCP <macro> (2 Arguments)`

`\intervalCP* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $[a; b)$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $[a; b)$ .

---

`\intervalP <macro> (2 Arguments)`

`\intervalP* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $(a; b)$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $(a; b)$ .

---

`\intervalPC <macro> (2 Arguments)`

`\intervalPC* <macro> (2 Arguments)`

— Argument 1: borne inférieure  $a$  de l'intervalle  $(a; b]$ .

— Argument 2: borne supérieure  $b$  de l'intervalle  $(a; b]$ .

#### 14.4.9 Intervalles discrets d'entiers

Pour toutes les macros ci-dessous, la version non étoilée produit des délimiteurs qui s'étirent si besoin verticalement, tandis que la version étoilée ne le fait pas.

---

`\ZintervalC0` <macro> (2 Arguments)  
`\ZintervalC0*` <macro> (2 Arguments)

- Argument 1: borne inférieure  $a$  de l'intervalle  $\llbracket a; b \rrbracket$ .
- Argument 2: borne supérieure  $b$  de l'intervalle  $\llbracket a; b \rrbracket$ .

---

`\ZintervalC` <macro> (2 Arguments)  
`\ZintervalC*` <macro> (2 Arguments)

- Argument 1: borne inférieure  $a$  de l'intervalle  $\llbracket a; b \rrbracket$ .
- Argument 2: borne supérieure  $b$  de l'intervalle  $\llbracket a; b \rrbracket$ .

---

`\Zinterval0` <macro> (2 Arguments)  
`\Zinterval0*` <macro> (2 Arguments)

- Argument 1: borne inférieure  $a$  de l'intervalle  $\llbracket a; b \rrbracket$ .
- Argument 2: borne supérieure  $b$  de l'intervalle  $\llbracket a; b \rrbracket$ .

---

`\Zinterval0C` <macro> (2 Arguments)  
`\Zinterval0C*` <macro> (2 Arguments)

- Argument 1: borne inférieure  $a$  de l'intervalle  $\llbracket a; b \rrbracket$ .
- Argument 2: borne supérieure  $b$  de l'intervalle  $\llbracket a; b \rrbracket$ .

#### 14.4.10 Unions et intersections

`\dcap` <macro> (Sans argument)  
`\dcup` <macro> (Sans argument)  
`\dsqcup` <macro> (Sans argument)

#### 14.4.11 Cardinal, image et compagnie

`\card` <macro> (Sans argument)  
`\card*` <macro> (Sans argument)

---

`\dom` <macro> (Sans argument)  
`\codom` <macro> (Sans argument)  
`\im` <macro> (Sans argument)  
`\ker` <macro> (Sans argument)

#### 14.4.12 Application totale, partielle, injective, surjective et/ou bijective

<code>\to &lt;macro&gt;</code> (Sans argument)	<code>\pto &lt;macro&gt;</code> (Sans argument)
<code>\onetoone &lt;macro&gt;</code> (Sans argument)	<code>\ponetoone &lt;macro&gt;</code> (Sans argument)
<code>\onto &lt;macro&gt;</code> (Sans argument)	<code>\ponto &lt;macro&gt;</code> (Sans argument)
<code>\bijet &lt;macro&gt;</code> (Sans argument)	<code>\pbijet &lt;macro&gt;</code> (Sans argument)

## 14.5 Géométrie

### 14.5.1 Points

`\pt <macro>` (1 Argument)

— Argument : un texte donnant le nom d'un point.

---

`\pt* <macro>` (2 Arguments)

— Argument 1 : un texte indiquant UP dans le nom  $UP_{down}$  d'un point.

— Argument 2 : un texte indiquant *down* dans le nom  $UP_{down}$  d'un point.

### 14.5.2 Lignes

`\gline <macro>` [1 Option] (2 Arguments)

`\pgline <macro>` [1 Option] (2 Arguments)

— Option : pour indiquer les parenthèses ou crochets à utiliser, les valeurs possibles étant 0, valeur par défaut, C, CO et OC.

— Argument 1 : le 1<sup>er</sup> point géométrique.

— Argument 2 : le 2<sup>e</sup> point géométrique.

---

`\hgline <macro>` (2 Arguments)

`\phgline <macro>` (2 Arguments)

`\segment <macro>` (2 Arguments)

`\psegment <macro>` (2 Arguments)

— Argument 1 : le 1<sup>er</sup> point géométrique.

— Argument 2 : le 2<sup>e</sup> point géométrique.

### 14.5.3 Droites parallèles ou non

`\parallel <macro>` (Sans argument)

`\nparallel <macro>` (Sans argument)

`\stdparallel <macro>` (Sans argument)

`\stdnparallel <macro>` (Sans argument)

### 14.5.4 Vecteurs

`\vect <macro>` (1 Argument)

— Argument : un texte donnant le nom d'un vecteur.

---

`\vect* <macro>` (2 Arguments)

- Argument 1: un texte indiquant *up* dans le nom  $\overrightarrow{up}_{down}$  d'un vecteur.
- Argument 2: un texte indiquant *down* dans le nom  $\overrightarrow{up}_{down}$  d'un vecteur.

#### 14.5.5 Norme

`\norm <macro> (1 Argument)`  
`\norm* <macro> (1 Argument)`

- Argument: le vecteur sur lequel appliquer la norme.

#### 14.5.6 Produit scalaire – Écriture minimaliste

`\dotprod <macro> (2 Arguments)`

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.

---

`\vdotprod <macro> (2 Arguments)` où `v = v-ector`

- Argument 1: le nom du 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: le nom du 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.

#### 14.5.7 Produit scalaire – Écriture « à la physicienne »

`\adotprod <macro> (2 Arguments)` où `a = a-ngle`  
`\adotprod* <macro> (2 Arguments)`

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.

---

`\vadotprod <macro> (2 Arguments)` où `a = a-ngle` et `v = v-ector`

- Argument 1: le nom du 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: le nom du 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.

#### 14.5.8 Produit vectoriel

`\crossprod <macro> (2 Arguments)`

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.

---

`\vcrossprod <macro> (2 Arguments)` où `v = v-ector`

- Argument 1: le nom du 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: le nom du 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.

---

`\ccrossprod <macro> (8 Arguments)` où `c = c-ordinates`

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.

- Argument 2: la 1<sup>re</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 3: la 2<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 4: la 3<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 5: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 6: la 1<sup>re</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 7: la 2<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 8: la 3<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.

---

`\vccrossprod` <macro> (8 Arguments) où `c` = c-coordinates et `v` = v-ector

- Argument 1: le 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: la 1<sup>re</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 3: la 2<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 4: la 3<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 5: le 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.
- Argument 6: la 1<sup>re</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 7: la 2<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 8: la 3<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.

#### 14.5.9 Plan – Déterminant de deux vecteurs

`\cdetplane` <macro> (6 Arguments) où `c` = c-coordinates  
`\cdetplane*` <macro> (6 Arguments) où `c` = c-coordinates

- Argument 1: le 1<sup>er</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 2: la 1<sup>re</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 3: la 2<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 4: le 2<sup>e</sup> vecteur qu'il faut taper via la macro `\vect`.
- Argument 5: la 1<sup>re</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 6: la 2<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.

---

`\vcdetplane` <macro> (6 Arguments) où `c` = c-coordinates et `v` = v-ector  
`\vcdetplane*` <macro> (6 Arguments) où `c` = c-coordinates et `v` = v-ector

- Argument 1: le 1<sup>er</sup> vecteur sans utiliser la macro `\vect`.
- Argument 2: la 1<sup>re</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 3: la 2<sup>e</sup> coordonnée du 1<sup>er</sup> vecteur.
- Argument 4: le 2<sup>e</sup> vecteur sans utiliser la macro `\vect`.
- Argument 5: la 1<sup>re</sup> coordonnée du 2<sup>e</sup> vecteur.
- Argument 6: la 2<sup>e</sup> coordonnée du 2<sup>e</sup> vecteur.

### 14.5.10 Coordonnées

`\coord <macro> (1 Argument)`

`\coord* <macro> (1 Argument)`

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | , chaque morceau étant une coordonnée. Il peut n'y avoir qu'un seul morceau.

---

`\vcoord <macro> (1 Argument)` où `v = v-ertical`

`\vcoord* <macro> (1 Argument)` pour des crochets à la place de parenthèses

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | , chaque morceau étant une coordonnée. Il peut n'y avoir qu'un seul morceau.

### 14.5.11 Nommer un repère

`\axes <macro> (1 Argument)`

`\axes* <macro> (1 Argument)`

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | .

- Le premier morceau est l'origine du repère.
  - Les morceaux suivants sont des points ou des vecteurs qui "définissent" chaque axe.
- 

`\paxes <macro> (1 Argument)` où `p = p-oint`

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | .

- Le premier morceau est le nom de l'origine du repère sur laquelle la macro-commande `\pt` sera automatiquement appliquée.
  - Viennent ensuite les noms des points "définissant" chaque axe. Pour chacun de ces points la macro-commande `\pt` sera automatiquement appliquée.
- 

`\vaxes <macro> (1 Argument)` où `v = v-ector`

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | .

- Le premier morceau est l'origine du repère.
  - Viennent ensuite les noms des vecteurs "définissant" chaque axe. Pour chacun de ces vecteurs la macro-commande `\vect` sera automatiquement appliquée.
- 

`\pvaxes <macro> (3 Arguments)` où `p v = p + v`

— **Argument**: l'argument est une suite de "morceaux" séparés par des barres | .

- Le premier morceau est le nom de l'origine du repère sur laquelle la macro-commande `\pt` sera automatiquement appliquée.
- Viennent ensuite les noms des vecteurs "définissant" chaque axe. Pour chacun de ces vecteurs la macro-commande `\vect` sera automatiquement appliquée.

### 14.5.12 Arcs circulaires

`\circarc <macro> (1 Argument)` où `circarc = circ-ular arc`

— **Argument**: un texte donnant le nom d'un arc circulaire.



---

`\circarc* <macro> (2 Arguments)` où `circarc` = circ-ular arc

— Argument 1: un texte indiquant *up* dans le nom  $\widehat{up}_{down}$  d'un arc circulaire.

— Argument 2: un texte indiquant *down* dans le nom  $\widehat{up}_{down}$  d'un arc circulaire.

### 14.5.13 Angles géométriques « intérieurs »

`\anglein <macro> (1 Argument)`

— Argument: un texte donnant le nom d'un angle intérieur.

---

`\anglein* <macro> (2 Arguments)`

— Argument 1: un texte indiquant *up* dans le nom  $\widehat{up}_{down}$  d'un angle intérieur.

— Argument 2: un texte indiquant *down* dans le nom  $\widehat{up}_{down}$  d'un angle intérieur.

### 14.5.14 Angles orientés de vecteurs

`\angleorient <macro> (2 Arguments)`

`\hangleorient <macro> (2 Arguments)` où `h` = h-at

— Argument 1: le premier vecteur qu'il faut taper via la macro `\vect`.

— Argument 2: le second vecteur qu'il faut taper via la macro `\vect`.

---

`\vangleorient <macro> (2 Arguments)` où `v` = v-ector

`\hvangleorient <macro> (2 Arguments)` où `h` = h-at et `v` = v-ector

— Argument 1: le nom du premier vecteur sans utiliser la macro `\vect`.

— Argument 2: le nom du second vecteur sans utiliser la macro `\vect`.

---

## 14.6 Analyse

### 14.6.1 Constantes classiques

`\ggamma <macro> (Sans argument)`

`\ii <macro> (Sans argument)`

`\ppi <macro> (Sans argument)`

`\jj <macro> (Sans argument)`

`\ttau <macro> (Sans argument)`

`\kk <macro> (Sans argument)`

`\ee <macro> (Sans argument)`

### 14.6.2 Constantes latines personnelles

`\ct <macro> (1 Argument)`

— Argument: un texte utilisant l'alphabet latin. `\abs <macro> (1 Argument)`

`\abs* <macro> (1 Argument)`

— Argument: l'expression à laquelle on applique la fonction valeur absolue.

### 14.6.3 Sans paramètre

`\pgcd <macro>` (Sans argument)  
`\ppcm <macro>` (Sans argument)

---

`\acos <macro>` (Sans argument)  
`\asin <macro>` (Sans argument)  
`\atan <macro>` (Sans argument)

---

`\arccosh <macro>` (Sans argument)  
`\arcsinh <macro>` (Sans argument)  
`\arctanh <macro>` (Sans argument)

---

`\acosh <macro>` (Sans argument)  
`\asinh <macro>` (Sans argument)  
`\atanh <macro>` (Sans argument)

---

`\fch <macro>` (Sans argument)  
`\fsh <macro>` (Sans argument)  
`\fth <macro>` (Sans argument)

### 14.6.4 Avec un paramètre

`\expb <macro>` (1 Argument)  
— Argument : la base de l'exponentielle

---

`\logb <macro>` (1 Argument)  
— Argument : la base du logarithme

### 14.6.5 Des suites spéciales

`\seqplus <macro>` (2 Arguments)  
— Argument 1 : l'exposant à droite.  
— Argument 2 : l'indice à droite.

---

`\seqhypergeo <macro>` (2 Arguments)  
— Argument 1 : l'indice à gauche.  
— Argument 2 : l'indice à droite.

---

`\seqsuprgeo <macro>` (4 Arguments)  
— Argument 1 : l'indice à gauche.  
— Argument 2 : l'indice à droite.

- Argument 3: l'exposant à droite.
- Argument 4: l'exposant à gauche.

### 14.6.6 Calcul différentiel

`\dd <macro> [1 Option] (1 Argument)`  
`\pp <macro> [1 Option] (1 Argument)`

- Option: utilisée, cette option sera mise en exposant du symbole  $\partial$  ou  $d$ .
- Argument: la variable de différentiation à droite du symbole  $\partial$  ou  $d$ .

### 14.6.7 Dérivation totale

`\derpar <macro> [1 Option] (1 Argument)`  
`\derpar* <macro> [1 Option] (1 Argument)`  
`\sderpar <macro> [1 Option] (1 Argument)`  
`\sderpar* <macro> [1 Option] (1 Argument)`  
`\derpow <macro> [1 Option] (1 Argument)`  
`\derpow* <macro> [1 Option] (1 Argument)`

- Option: utilisée, cette option sera l'exposant de dérivation mis entre des parenthèses pour la version non étoilée, et le nombre de primes pour la version étoilée.
- Argument: la fonction à différencier.

---

`\derfrac <macro> [1 Option] (2 Arguments)`  
`\derfrac* <macro> [1 Option] (2 Arguments)`  
`\dersub <macro> [1 Option] (2 Arguments)`

- Option: utilisée, cette option sera l'exposant de dérivation.
- Argument 1: la fonction à dériver.
- Argument 2: la variable.

### 14.6.8 Dérivation partielle

`\partialfrac <macro> [1 Option] (2 Arguments)`  
`\partialfrac* <macro> [1 Option] (2 Arguments)`

- Option: utilisée, cette option sera l'exposant total de dérivation mis en exposant de  $\partial$ .
- Argument 1: la fonction à dériver partiellement.
- Argument 2: les variables utilisées pour la dérivation partielle en utilisant la syntaxe suivante :  
par exemple, `x | y^3 | ...` indique de dériver suivant  $x$  une fois, puis suivant  $y$  trois fois... etc.

---

`\partialsub <macro> (2 Arguments)`  
`\partialprime <macro> (2 Arguments)`

- Argument 1: la fonction à dériver partiellement.
- Argument 2: les variables utilisées pour la dérivation partielle en utilisant la syntaxe suivante :  
par exemple, `x | y^3 | ...` indique de dériver suivant  $x$  une fois, puis suivant  $y$  trois fois... etc.

### 14.6.9 L'opérateur crochet – 1<sup>ère</sup> version

`\hook <macro> (3 Arguments)`

`\hook* <macro> (3 Arguments)`

- Argument 1: le contenu entre les crochets.
- Argument 2: la borne inférieure affichée en indice.
- Argument 3: la borne supérieure affichée en exposant.

### 14.6.10 L'opérateur crochet – 2<sup>nde</sup> version

`\vhook <macro> (3 Arguments)`

`\vhook* <macro> (3 Arguments)`

- Argument 1: le contenu avant le trait vertical.
- Argument 2: la borne inférieure affichée en indice.
- Argument 3: la borne supérieure affichée en exposant.

### 14.6.11 L'opérateur d'intégration standard

`\stdint <macro> (Sans argument)`

### 14.6.12 Les notations $\mathcal{O}$ et $\mathcal{o}$

`\bigO <macro> (1 Argument)`

`\smallO <macro> (1 Argument)`

- Argument: si l'argument est vide, il est ignoré, sinon il est mis entre des parenthèses après  $\mathcal{O}$  ou  $\mathcal{o}$ .

### 14.6.13 La notation $\Omega$

`\bigomega <macro> (1 Argument)`

- Argument: si l'argument est vide, il est ignoré, sinon il est mis entre des parenthèses après  $\Omega$ .

### 14.6.14 La notation $\Theta$

`\bigtheta <macro> (1 Argument)`

- Argument: si l'argument est vide, il est ignoré, sinon il est mis entre des parenthèses après  $\Theta$ .

## 14.7 Probabilité

### 14.7.1 Probabilité « simple »

`\proba <macro> [1 Option] (1 Argument)`

- Option: le nom de la probabilité.
- Argument: l'ensemble dont on veut calculer la probabilité.

### 14.7.2 Probabilité conditionnelle

`\probacond <macro> [1 Option] (2 Arguments)`  
`\probacond* <macro> [1 Option] (2 Arguments)`  
`\probacond** <macro> [1 Option] (2 Arguments)`  
`\dprobacond** <macro> [1 Option] (2 Arguments)`

- Option: le nom de la probabilité.
- Argument 1: l'ensemble qui donne la condition.
- Argument 2: l'ensemble dont on veut calculer la probabilité.

### 14.7.3 Arbres pondérés

`probatree <env>`  
`probatree* <env>`

- Contenu: un arbre codé en utilisant la syntaxe supportée par le package `forest`.
- Option "*pweight*": pour écrire un poids sur le milieu d'une branche.
- Option "*apweight*": pour écrire un poids au-dessus le milieu d'une branche.
- Option "*bpweight*": pour écrire un poids en-dessous du milieu d'une branche.
- Option "*frame*": pour encadrer un sous-arbre depuis un noeud vers toutes les feuilles de celui-ci.

## 14.8 Arithmétique

### 14.8.1 Opérateurs de base

`\divides <macro> (Sans argument)`  
`\notdivides <macro> (Sans argument)`  
`\modulo <macro> (Sans argument)`

### 14.8.2 Fractions continuées standard

`\contfrac <macro> (1 Argument)`  
`\contfrac* <macro> (1 Argument)`

- Argument: tous les éléments de la fraction continuée séparés par des `|`.

### 14.8.3 Fractions continuées généralisées

`\contfracgene <macro> (1 Argument)`  
`\contfracgene* <macro> (1 Argument)`

- Argument: tous les éléments de la fraction continuée généralisée séparés par des `|`.

### 14.8.4 Comme une fraction continuée isolée

`\singlecontfrac <macro> (2 Arguments)`

- Argument 1: le pseudo numérateur.
- Argument 2: le pseudo dénominateur.

### 14.8.5 L'opérateur $\mathcal{K}$

La macro suivante sans argument a un comportement spécifique vis à vis des mises en index et en exposant.

---

`\contfracope <macro>` (Sans argument)

## 14.9 Algèbre

### 14.9.1 Polynômes, séries formelles et compagnie

`\setpoly <macro>` (2 Arguments)

`\setpolyfrac <macro>` (2 Arguments)

`\setserie <macro>` (2 Arguments)

`\setseriefrac <macro>` (2 Arguments)

`\setpolylaurent <macro>` (2 Arguments)

`\setserielaurent <macro>` (2 Arguments)

— **Argument 1**: l'ensemble auquel les coefficients appartiennent.

— **Argument 2**: cet argument est une suite de "morceaux" séparés par des barres |, chaque morceau étant une variable formelle.