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

1 Introduction

LaTeX est un excellent langage, pour ne pas dire le meilleur, pour rédiger des documents contenant des formules mathématiques. Malheureusement toute la puissance de LaTeX 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 LaTeX suivant.

```
Sachant que $\frac{df}{dx}(x) = cos(x^2)$ sur $[a ; b]$ , nous avons :

$\int_a^b cos(x^2) dx = \left[ f(x) \right]_a^b$.
```

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

```
Sachant que $\derfrac{f}{x}(x) = cos(x^2)$ sur $\intervalC{a}{b}$, nous avons :

$\int_a^b cos(x^2) \dd{x} = \hook{f(x)}{a}{b}$.
```

Même si certaines commandes sont plus longues à écrire que ce que permet LaTeX, il y a trois avantages à utiliser des commandes sémantiques.

1. La mise en forme dans votre document devient 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 *(voir la section 14)*. Les exemples se présentent comme ci-dessous.

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

$$\int_a^b cos(x^2)dx = [f(x)]_a^b$$

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 [1].
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 soit* « non » *en anglais)*.

3.1.2 Les macros standards redéfinies

Certaines macros comme `\frac` sont un peu revues par `lymath`. Dans ce cas, les versions standard restent accessibles en utilisant le préfixe `std` ce qui donne ici la macro `\stdfrac`.

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 ... »*.

3.1.3 Les macros en mode `displaystyle`

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

3.2 Versions étoilées

Les versions étoilées proposent des mises en forme correspondant aux cas les moins usuels ou les moins pratiques : par exemple, la macro `\derpow` utilise un exposant entre des parenthèses pour rédiger une dérivée n$^{\text{e}}$ 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 Avec un nombre fixé d'arguments

Dans ce cas, c'est la syntaxe LaTeX usuelle qui sera à utiliser comme dans `\derfrac{f}{x}`.

3.3.2 Avec un 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 LaTeX, 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 LaTeX, sera analysé par `lymath` comme étant formé des quatre arguments `x` , `y` , `z` et `t`.

4 Couleurs

Certaines macros utilisent de la couleur. Les choix faits sont tels que l'impression en noir et blanc ne soit pas impactée.

5 Quelques modifications générales

5.1 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 $\boxed{;}$ et celle de `\lymathsubsep` est $\boxed{,}$. Sans ce choix, les valeurs de `\lymathsep` et `\lymathsubsep` seront $\boxed{,}$ et $\boxed{;}$ respectivement.

5.2 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 !

5.3 Espace et fractions

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 `\stddfrac`. Voici un exemple.

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

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

5.4 Espace et racines n-ièmes d'un réel

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

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

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

5.5 Espace après la négation logique

`\neg` a été redéfinie pour ajouter un peu d'espace après le symbole. Le comportement par défaut se retrouve en utilisant la macro `\stdneg`. Voici un exemple.

```
$\neg A = \stdneg A$

$\neg\neg A = \stdneg \stdneg A$
```

$\neg A = \stdneg A$

$\neg\neg A = \stdneg \stdneg A$

5.6 Sommes et produits en mode ligne

Pour limiter l'espace, LaTeX affiche $\sum_{k=0}^{n}$ et non $\displaystyle\sum_{k=0}^{n}$ sauf si l'on utilise la commande `\displaystyle`. Les macros `\dsum` et `\dprod` permettent de se passer de `\displaystyle`. Voici un exemple.

```
$\dsum_{k=0}^{n} 2^k =
 \sum_{k=0}^{n} 2^k$

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

$\displaystyle\dsum_{k=0}^{n} 2^k = \sum_{k=0}^{n} 2^k$

$\displaystyle\dprod_{k=1}^{n} k = \prod_{k=1}^{n} k$

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 [2] *(la section 6.1.8 explique comment est définit le texte « déf »).*

```
$f(x) \eqdef x^3 + 1$ ou
$f(x) \eqdef* x^3 + 1$
```

$f(x) \eqdef x^3 + 1$ ou $f(x) \eqdef* x^3 + 1$

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

6.1.2 Indiquer une identité

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

```
$(a + b)^2 \eqid a^2 + b^2 + 2 a b$ ou
$(a + b)^2 \eqid* a^2 + b^2 + 2 a b$ .
```

$(a+b)^2 \overset{\text{id}}{=} a^2+b^2+2ab$ ou $(a+b)^2 \overset{\text{id}}{\rightleftharpoons} a^2+b^2+2ab$.

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 »*.

```
$(a + b)^3 \eqtest a^3 + b^3 + 3 a b$

$(a + b)^3 \neqid a^3 + b^3 + 3 a b$

$x \neqhyp 0$ ou $x \neqcond 0$

$x \eqcond 0$ ou $x \eqhyp 0$ ?
```

$(a+b)^3 \overset{?}{=} a^3+b^3+3ab$

$(a+b)^3 \overset{\text{id}}{\neq} a^3+b^3+3ab$

$x \overset{\text{hyp}}{\neq} 0$ ou $x \overset{\text{cond}}{\neq} 0$

$x \overset{\text{cond}}{=} 0$ ou $x \overset{\text{hyp}}{=} 0$?

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.

```
$x \geqcond 4$ implique
$x^2 \geqcons 16$.

Alors $x \eqchoice 123$ donne
$123^2 \geqappli 16$.
```

$x \overset{\text{cond}}{\geq} 4$ implique $x^2 \overset{\text{cons}}{\geq} 16$.

Alors $x \overset{\text{choix}}{=} 123$ donne $123^2 \overset{\text{appli}}{\geq} 16$.

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)*.

```
$M \in \setgeo{C}: y \eqplot x^2 + 3$
donne
$y_M \eqappli x_M^2 + 3$.
```

$M \in \mathscr{C} : y \overset{\text{graph}}{=} x^2+3$ donne $y_M \overset{\text{appli}}{=} x_M^2+3$.

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.

```
$x \leqtest x^2$ ou $x \ltest x^2$ ou
$x \geqhyp 1$ ou $x \gcond 2$.
```

$x \overset{?}{\leq} x^2$ ou $x \overset{?}{<} x^2$ ou $x \overset{\text{hyp}}{\geq} 1$ ou $x \overset{\text{cond}}{>} 2$.

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.

`\textopappli` donne *« appli »*
`\textopchoice` donne *« choix »*
`\textopcond` donne *« cond »*
`\textopcons` donne *« cons »*
`\textopdef` donne *« déf »*
`\textophyp` donne *« hyp »*
`\textopid` donne *« id »*
`\textopplot` donne *« graph »*
`\textoptest` donne *« ? »*

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 LaTeX, il a été ajouté l'opérateur `\liesimp`, où l'on a inversé les groupes syllabiques de `\implies`, un opérateur pour pour obtenir $\Longleftarrow$[3], ainsi que des versions négatives. Voici un exemple d'utilisation.

```
$(A \implies B)
 \iff (B \liesimp A)$

$(A \implies B)
 \notiff (A \notimplies B)$
```

$$(A \implies B) \iff (B \Longleftarrow A)$$
$$(A \implies B) \not\iff (A \not\Longrightarrow B)$$

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.

```
$A \iffappli B \notiffchoice C$

$A \impliescond B \notimpliescons C$

$A \liesimphyp B \notliesimptest C$
```

$$A \overset{\text{appli}}{\iff} B \overset{\text{choix}}{\not\iff} C$$
$$A \overset{\text{cond}}{\implies} B \overset{\text{cons}}{\not\Longrightarrow} C$$
$$A \overset{\text{hyp}}{\Longleftarrow} B \overset{?}{\not\Longleftarrow} C$$

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 6.5.1 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)*. Bien entendu le préfixe `v` est pour `v`-ertical.

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

```
\begin{tabular}{cccccc}
  $A$             & $B$
& $C$             & $D$
& $E$             & $F$
\\
  $\viff$         & $\vimplies$
& $\vliesimp$     & $\notviff$
& $\notvimplies$ & $\notvliesimp$
\\
  $A$             & $B$
& $C$             & $D$
& $E$             & $F$
\end{tabular}
```

$$\begin{array}{cccccc} A & B & C & D & E & F \\ \Updownarrow & \Downarrow & \Uparrow & \not\Updownarrow & \not\Downarrow & \not\Uparrow \\ A & B & C & D & E & F \end{array}$$

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
\eq	×	×	×	×	×	×	×	×	×	×	×
\neq	×	×	×	×			×	×			×
\l	×	×	×	×			×			×	×
\g	×	×	×	×			×			×	×
\leq	×	×	×	×			×			×	×
\geq	×	×	×	×			×			×	×
\iff	×	×	×	×			×				×
\notiff	×	×	×	×			×				×
\implies	×	×	×	×			×				×
\notimplies	×	×	×	×			×				×
\liesimp	×	×	×	×			×				×
\notliesimp	×	×	×	×			×				×

6.4 Des versions alternatives du quantificateur ∃

Quantifier l'existence

Voici deux versions, l'une classique, et l'autre beaucoup moins, permettant de préciser le quantificateur ∃.

```
$\existsone x \in E$
pour \og il existe un seul $x$ \fg.

$\existmulti{\leq 1} y \in F$
pour \og il existe au plus un $y$ \fg.

$\existmulti{=1} \eqdef \existsone$
```

$\exists! x \in E$ pour « il existe un seul x ».
$\exists_{\leq 1} y \in F$ pour « il existe au plus un y ».
$\exists_{=1} \overset{\text{déf}}{=} \exists!$

Versions négatives

```
$\nexistsone$
pour \og il n'existe pas un unique \fg.

$\nexistmulti{>4}$
pour \og il n'existe pas plus de
quatre \fg.

$\nexists$
est proposé par \verb+amssymb+.
```

$\nexistsone$
pour \og il n'existe pas un unique \fg.

$\nexistmulti{>4}$
pour \og il n'existe pas plus de
quatre \fg.

$\nexists$
est proposé par \verb+amssymb+.

∄! pour « il n'existe pas un unique ».
$\nexists_{>4}$ pour « il n'existe pas plus de quatre ».
∄ est proposé par `amssymb`.

6.5 Détailler un raisonnement

6.5.1 Détailler un raisonnement simple – Version pour le lycée et après

Exemple 1 – Avec les réglages par défaut

L'environnement `explain` permet de détailler les étapes principales d'un calcul ou d'un raisonnement simple en s'appuyant sur la macro `\explnext` dont le nom vient de *« **expl**-ain **next** step »* soit *« expliquer la prochaine étape »* en anglais [4]. On dispose aussi de `\explnext*` pour des explications descendantes et/ou montantes [5].

Ci-dessous se trouve un exemple, un peu farfelu vers la fin, où l'on utilise les réglages par défaut. Notons au passage que ce type de présentation n'est sûrement pas bien adaptée à un jeune public pour lequel une 2e façon de détailler des calculs et/ou un raisonnement simple est proposée plus bas dans la section 6.5.2.

4. Le gros du travail est fait par l'environnement `flalign` du package `amsmath` qui est automatiquement chargé par `lymath`.
5. Les explications données ne doivent pas être trop longues car ce serait contre productif.

```
\begin{explain}
    (a + b)^2
        \explnext{On utilise $x^2 = x \cdot x$.}
    (a + b) (a + b)
        \explnext*{Double développement depuis la parenthèse gauche.}%
                  {Double factorisation non nécessairement évidente.}
    a^2 + a b + b a + b^2
        \explnext*{}%
                  {Commutativité du produit.}
    a^2 + 2 a b + b^2
        \explnext*{Commutativité de l'addition.}%
                  {}
    a^2 + b^2 + 2 a b
\end{explain}
```

$$
\begin{aligned}
&(a+b)^2 \\
&= \quad \{\ \textit{On utilise } x^2 = x \cdot x.\ \} \\
&(a+b)(a+b) \\
&= \quad \{\ \downarrow \textit{Double développement depuis la parenthèse gauche.} \downarrow\ \} \\
&\quad \{\ \uparrow \textit{Double factorisation non nécessairement évidente.} \uparrow\ \} \\
&a^2 + ab + ba + b^2 \\
&= \quad \{\ \uparrow \textit{Commutativité du produit.} \uparrow\ \} \\
&a^2 + 2ab + b^2 \\
&= \quad \{\ \downarrow \textit{Commutativité de l'addition.} \downarrow\ \} \\
&a^2 + b^2 + 2ab
\end{aligned}
$$

Remarque. Voici des petites choses à connaître sur les macros `\explnext` et `\explnext*`.

1. `\expltxt` est utilisée par `\explnext` pour mettre en forme le texte d'explication.
2. `\expltxtup` et `\expltxtdown` sont utilisées par `\explnext*` pour mettre en forme les textes d'explication. Ces macros s'appliquent sur `\expltxt`.
3. `\explnext` et `\explnext*` utilisent la macro constante `\expltxtspacein` pour l'espacement 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.

```
\begin{explain}[\viff]
    x^2 + 10 x + 25 = 0
        \explnext{Identité remarquable.}
    (x + 5)^2 = 0
        \explnext{$P^2 = 0$ si et
                  seulement si $P = 0$.}
    x = -5
\end{explain}
```

$$
\begin{aligned}
&x^2 + 10x + 25 = 0 \\
&\Updownarrow \quad \{\ \textit{Identité remarquable.}\ \} \\
&(x+5)^2 = 0 \\
&\Updownarrow \quad \{\ P^2 = 0 \textit{ si et seulement si } P = 0.\ \} \\
&x = -5
\end{aligned}
$$

Exemple 3 – Juste utiliser des symboles

Si l'argument obligatoire de la macro `\explnext` est vide alors seul le symbole est affiché *(ne pas oublier les accolades vides)*. Voici un court exemple de ceci.

```
\begin{explain}[\viff]
    a^2 = b^2
        \explnext{}
    a = \pm b
\end{explain}
```

$$
\begin{aligned}
& a^2 = b^2 \\
& \Updownarrow \\
& a = \pm b
\end{aligned}
$$

Exemple 4 – Utiliser un autre symbole localement

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

```
\begin{explain}[\viff]
    0 \leq a < b
        \explnext[\vimplies]%
                 {Croissance de $x^2$
                  sur $\RRp$.}
    a^2 < b^2
        \explnext{}
    a^2 - b^2 < 0
        \explnext{Identité remarquable.}
    (a - b)(a + b) < 0
        \explnext[\vimplies]{}
    a \neq b
\end{explain}
```

$$
\begin{aligned}
& 0 \leq a < b \\
& \Downarrow \qquad \{ \textit{Croissance de } x^2 \textit{ sur } \mathbb{R}_+. \} \\
& a^2 < b^2 \\
& \Updownarrow \\
& a^2 - b^2 < 0 \\
& \Updownarrow \qquad \{ \textit{Identité remarquable.} \} \\
& (a-b)(a+b) < 0 \\
& \Downarrow \\
& a \neq b
\end{aligned}
$$

Exemple 5 – Choisir la mise en forme des explications

Pour la mise en forme des explications à double sens, la macro `\explnext` fait appel à la macro `\expltxt`. Par défaut, le package utilise la définition suivante.

```
\newcommand\expltxt[1]{%
    \text{\color{blue} \footnotesize \itshape \{ #1 \}}%
}
```

Pour la mise en forme des explications à sens unique, la macro `\explnext*` fait appel aux macros `\expltxtup` et `\expltxtdown`. Par défaut, le package utilise les définitions suivantes qui fait appel à `\expltxt`.

```
\newcommand\expltxtup[1]{%
    \expltxt{$\uparrow$ #1 $\uparrow$}%
}

\newcommand\expltxtdown[1]{%
    \expltxt{$\downarrow$ #1 $\downarrow$}%
}
```

Voici un exemple un peu moche montrant que l'on peut adapter si besoin la mise en forme *(ne pas oublier de passer en mode texte via* `\text`*)*.

```
\newcommand\myexpltxt[2]{%
    \text{\color{#1} \footnotesize \itshape \bfseries #2}%
}

\renewcommand\expltxt[1]{%
    \myexpltxt{gray}{$\Downarrow$ #1 $\Uparrow$}%
}

\renewcommand\expltxtup[1]{%
    \myexpltxt{orange}{$\Uparrow$ #1 $\Uparrow$}%
}

\renewcommand\expltxtdown[1]{%
    \myexpltxt{red}{$\Downarrow$ #1 $\Downarrow$}%
}

\begin{explain}
    (a + b) (a + b)
        \explnext{Se souvenir de $P\cdot P = P^2$.}%
    (a + b)^2
        \explnext*{Identité remarquable pour développer.}%
                  {Identité remarquable pour factoriser.}
    a^2 + 2 a b + b^2
\end{explain}
```

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

$=$ $\Downarrow$ ***Se souvenir de*** $P\cdot P = P^2$. $\Uparrow$

$(a+b)^2$

$=$ $\Downarrow$ ***Identité remarquable pour développer.*** $\Downarrow$

$\Uparrow$ ***Identité remarquable pour factoriser.*** $\Uparrow$

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

6.5.2 Détailler un raisonnement simple – Version pour les collégiens

L'environnement `aexplain`[6] utilise des flèches pour indiquer les explications *(*`a` *est pour* `a`*-rrow soit* « flèche » *en anglais)*. Son fonctionnement est similaire à celui de `explain` si ce n'est que la macro `\explnext*` permet d'avoir une flèche unidirectionnelle, vers le haut ou le bas au choix, ou bien d'écrire deux indications dont l'une est montante et l'autre descendante.

Il existe aussi l'environnement `aexplain*` lorsque la toute 1re étape n'est pas expliquée. Par contre ceci nécessite au tout début de l'environnement l'usage de la macro très spéciale `\explnext` sans aucun contenu !

6. Cet environnement utilise le package `witharrows` qui est très sympathique pour expliquer des étapes de calcul.

Exemple 1 – Flèche à double sens ou sans flèche

```
\begin{aexplain}
    (a + b)^2
        \explnext{Identité remarquable}
    a^2 + 2 a b + b^2
        \explnext{}
    a^2 + b^2 + 2 a b
\end{aexplain}
```

$$
\begin{aligned}
&(a+b)^2 \\
&= a^2 + 2ab + b^2 \quad \text{⟆ Identité remarquable} \\
&= a^2 + b^2 + 2ab
\end{aligned}
$$

Exemple 2 – Des flèches unidirectionnelles

Ce qui suit est juste là comme démo. car les explications y sont un peu farfelues.

```
\begin{aexplain}
    (a + b)^2
        \explnext*{Via $P^2 = P \cdot P$.}
                  {Via $P \cdot P = P^2$.}
    (a + b) (a + b)
        \explnext*{Double développement.}%
                  {Double factorisation (pas simple).}
    a^2 + a b + b a + b^2
        \explnext*{Commutativité du produit.}%
                  {}
    a^2 + 2 a b + b^2
        \explnext*{}%
                  {Commutativité de l'addition.}
    a^2 + b^2 + 2 a b
\end{aexplain}
```

$$
\begin{aligned}
&(a+b)^2 \\
&= (a+b)(a+b) && \text{⟅ Via } P^2 = P \cdot P. \quad \text{⟆ Via } P \cdot P = P^2. \\
&= a^2 + ab + ba + b^2 && \text{⟅ Double développement.} \quad \text{⟆ Double factorisation (pas simple).} \\
&= a^2 + 2ab + b^2 && \text{⟅ Commutativité du produit.} \\
&= a^2 + b^2 + 2ab && \text{⟆ Commutativité de l'addition.}
\end{aligned}
$$

Exemple 3 – Ne pas expliquer le tout début

Ci-dessous, la macro `\explnext` n'est à utiliser qu'une seule fois !

```
\begin{aexplain*}
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{Identité remarquable.}
    a^2 + b^2 + 2 a b
\end{aexplain*}
```

$$
\begin{aligned}
(a+b)(a+b) &= (a+b)^2 \\
&= a^2 + b^2 + 2ab \quad \text{⟆ Identité remarquable.}
\end{aligned}
$$

Exemple 4 – Choisir son symbole

Voici comment faire avec la version non étoilée où l'implication finale est juste là pour la démo. *(on notera une petite bidouille un peu sale à faire pour avoir un alignement à peu près correct).*

```
\begin{aexplain}[\iff]
    a^2 + 2 a b + b^2 = 0
        \explnext{}
    (a + b)^2 = 0
        \explnext[\:\implies]%
                 {$P^2 = 0$ ssi $P = 0$.}
    a + b = 0
\end{aexplain}
```

$$\begin{aligned} & a^2 + 2ab + b^2 = 0 \\ \iff\ & (a+b)^2 = 0 \\ \implies\ & a + b = 0 \end{aligned} \quad \left\} P^2 = 0 \text{ ssi } P = 0. \right.$$

Avec la version étoilée, on obtient ce qui suit.

```
\begin{aexplain*}[\iff]
    a^2 + 2 a b + b^2 = 0
        \explnext{}
    (a + b)^2 = 0
        \explnext[\:\implies]%
                 {$P^2 = 0$ ssi $P = 0$.}
    a + b = 0
\end{aexplain*}
```

$$\begin{aligned} a^2 + 2ab + b^2 = 0 \iff\ & (a+b)^2 = 0 \\ \implies\ & a + b = 0 \end{aligned} \quad \left\} P^2 = 0 \text{ ssi } P = 0. \right.$$

6.5.3 Un mini hack très utile pour des *« étapes alignées »*

Vous pouvez écrire très facilement des calculs ou raisonnement simples alignés comme suit sans trop vous fatiguez *(chacune des trois solutions présentées laisse toujours la possibilité de changer de symbole ainsi que d'expliquer une étape intermédiaire après coup).*

```
\begin{aexplain*}
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{}
    a^2 + b^2 + 2 a b
        \explnext{}
    a^2 + 2 a b + b^2
\end{aexplain*}
```

$$\begin{aligned} (a+b)(a+b) &= (a+b)^2 \\ &= a^2 + b^2 + 2ab \\ &= a^2 + 2ab + b^2 \end{aligned}$$

On a accès à une autre mise en forme *(ceci peut rendre aussi service).*

```
\begin{aexplain}
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{}
    a^2 + b^2 + 2 a b
        \explnext{}
    a^2 + 2 a b + b^2
\end{aexplain}
```

$$
\begin{aligned}
&(a+b)(a+b)\\
&= (a+b)^2\\
&= a^2+b^2+2ab\\
&= a^2+2ab+b^2
\end{aligned}
$$

Enfin dans le cadre de calculs à faire expliquer par des élèves, ce qui suit peut être utile.

```
\begin{explain}
    (a + b) (a + b)
        \explnext{}
    (a + b)^2
        \explnext{}
    a^2 + b^2 + 2 a b
        \explnext{}
    a^2 + 2 a b + b^2
\end{explain}
```

$$(a+b)(a+b)$$
$$=$$
$$(a+b)^2$$
$$=$$
$$a^2+b^2+2ab$$
$$=$$
$$a^2+2ab+b^2$$

6.5.4 Détailler un « vrai » raisonnement – Un tableau pour le post-bac

Exemple 1 – Le minimum avec les réglages par défaut

Prenons un exemple utile à la logique formelle en informatique théorique mais qui a complètement sa place en mathématiques plus classiques *(voir la section* **??** *pour un autre type de présentation plus adaptée à un public de collège ou de lycée)*. Ci-dessous l'environnement `demoexplain` facilite la mise en page via l'environnement `tabular` utilisé en coulisse, et la macro `\explbox` trace les cadres toujours de largeur `1.5em`[7]. Dans cet exemple en deux morceaux, pour montrer au passage comment continuer la numérotation là où elle s'était arrêtée, on utilise *« m.p. »* l'abréviation de *« modus ponens »*.

```
\begin{demoexplain}
    \demostep
          Hypothèse & $A$
    \demostep
          Axiome 1  & $A \implies B$
    \demostep
          m.p. sur
          \explref{1} et \explref{2}
        & $B$
    \demostep
          \explref{1} et \explref{3}
        & $A \wedge B$
\end{demoexplain}
```

1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B
4	1 et 3	$A \wedge B$

Il est possible de couper sa démonstration en morceaux en indiquant à l'environnement la valeur du 1er numéro de justification via la clé `start` : la valeur spéciale `last` indique de continuer la numérotation à la suite.

7. Ceci permet de numéroter les indications jusqu'à 99 ce qui est bien au-delà des besoins pratiques.

```
\begin{demoexplain}[start = last]
    \demostep
          Axiome 3
        & $(A \wedge B) \implies C$
    \demostep
          m.p. sur
          \explref{4} et \explref{5}
        & $C$
\end{demoexplain}
```

5	Axiome 3	$(A \wedge B) \implies C$
6	m.p. sur 4 et 5	C

Exemple 2 – Référencer une indication

L'argument optionnel de `\demostep` permet de définir un label qui ensuite facilitera le référencement d'une justification de façon pérenne via la macro étoilée `\explbox*`.

```
\begin{demoexplain}
    \demostep[demo-my-hyp]
          Hypothèse & $A$
    \demostep[demo-use-axiom-1]
          Axiome 1  & $A \implies B$
    \demostep
          m.p. sur
          \explref*{demo-my-hyp}
          et
          \explref*{demo-use-axiom-1}
        & $B$
\end{demoexplain}
```

1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B

Remarque. Prendre bien garde au fait que ce mécanisme utilise les macros `\label` et `\ref` de LaTeX. On travaille donc avec des références globalement au document compilé.

Exemple 3 – Indiquer ce que l'on cherche à faire

Les clés optionnelles `hyps` pour plusieurs hypothèses, `hyp` pour une seule hypothèse et `ccl` pour la conclusion permettent d'expliquer ce que l'on démontre et sous quel contexte.

```
\begin{demoexplain}[hyp = $A$, ccl = $B$]
    \demostep
          Hypothèse & $A$
    \demostep
          Axiome 1  & $A \implies B$
    \demostep
          m.p. sur
          \explref{1} et \explref{2}
        & $B$
\end{demoexplain}
```

Démonstration sous la seule hypothèse : A

1	Hypothèse	A
2	Axiome 1	$A \implies B$
3	m.p. sur 1 et 2	B

Conclusion : B

Remarque. Aucune des clés `hyps`, `hyp` et `ccl` n'est obligatoire. Par contre il n'est pas possible d'utiliser à la fois les clés `hyps` et `hyp`.

6.5.5 Détailler un « vrai » raisonnement – Un tableau pour le collège et le lycée

Exemple 1 – Avec les réglages par défaut

L'environnement étoilé `demoexplain*` est différent de l'environnement `demoexplain` puisqu'il sert à indiquer trois choses et non juste deux comme le montre l'exemple suivant. Par contre, la syntaxe est très similaire. Notez au passage la possibilité d'utiliser `\newline` pour forcer un retour à la ligne dans une cellule.

```
\begin{demoexplain*}
    \demostep
          $ABC$ est un triangle \newline équilatéral
        & Dans un triangle équilatéral, les trois angles mesurent $60$\textdegree.
        & $\anglein{ABC} = 60$\textdegree
    \demostep
          Voir la conséquence \explref{1} .
        & Simple calcul avec conversion en radians.
        & $\dfrac{1}{3} \anglein{ABC} = \dfrac{\pi}{9}$
\end{demoexplain*}
```

Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
1	ABC est un triangle équilatéral	Dans un triangle équilatéral, les trois angles mesurent 60°.	$\widehat{ABC} = 60°$
2	Voir la conséquence [1] .	Simple calcul avec conversion en radians.	$\frac{1}{3}\widehat{ABC} = \frac{\pi}{9}$

Exemple 2 – Avec toutes les options

Le système de référence marche ici aussi. Par contre `demoexplain*` ne propose que `start` comme clé optionnelle avec le même fonctionnement que pour `demoexplain`.

```
\begin{demoexplain*}[start = last]
    \demostep[demo-first-geo-fact]
          $ABC$ est un triangle \newline équilatéral
        & Dans un triangle équilatéral, les trois angles mesurent $60$\textdegree.
        & $\anglein{ABC} = 60$\textdegree
    \demostep
          Voir la conséquence \explref*{demo-first-geo-fact} .
        & Simple calcul avec conversion en radians.
        & $\dfrac{1}{3} \anglein{ABC} = \dfrac{\pi}{9}$
\end{demoexplain*}
```

Réf.	Je sais que...	Propriété ou fait utilisé	Conséquence
3	ABC est un triangle équilatéral	Dans un triangle équilatéral, les trois angles mesurent 60°.	$\widehat{ABC} = 60°$
4	Voir la conséquence [3] .	Simple calcul avec conversion en radians.	$\frac{1}{3}\widehat{ABC} = \frac{\pi}{9}$

7 Ensembles et applications

7.1 Différents types d'ensembles

7.1.1 Ensembles versus accolades

Exemple d'utilisation 1

`$\setgene{1 ; 3 ; 5}$ .`	$\{1\,;3\,;5\}$.

Exemple d'utilisation 2

`$\displaystyle`
` \setgene{\frac{1}{3} ; \frac{5}{7} ;`
`  \frac{9}{11}}$`
`ou`
`$\displaystyle`
` \setgene*{\frac{1}{3} ; \frac{5}{7} ;`
`  \frac{9}{11}}$ .`
	$\displaystyle \left\{\frac{1}{3}\,;\frac{5}{7}\,;\frac{9}{11}\right\}$ ou $\displaystyle \{\frac{1}{3}\,;\frac{5}{7}\,;\frac{9}{11}\}$.

7.1.2 Ensembles pour la géométrie

Exemple d'utilisation 1

`$\setgeo{C}$ ,`
`$\setgeo{D}$ ou`
`$\setgeo{d}$`
	$\mathscr{C}$, $\mathscr{D}$ ou $\mathscr{d}$

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

Exemple d'utilisation 2 – Avec des indices

`$\setgeo*{C}{1}$ ou` `$\setgeo*{C}{2}$`	$\mathscr{C}_1$ ou $\mathscr{C}_2$

7.1.3 Ensembles probabilistes

Exemple d'utilisation 1

`$\setproba{E}$ ou` `$\setproba{G}$`	$\mathcal{E}$ 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

`$\setproba*{E}{1}$ ou` `$\setproba*{E}{2}$`	$\mathcal{E}_1$ ou $\mathcal{E}_2$

7.1.4 Ensembles pour l'algèbre générale

Exemple d'utilisation 1

`$\setalge{A}$ ,`
`$\setalge{K}$ ,`
`$\setalge{h}$ ou`
`$\setalge{k}$`
	$\mathbb{A}$, $\mathbb{K}$, $\mathbb{h}$ ou $\mathbb{k}$

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

Exemple d'utilisation 2 – Avec des indices

`$\setalge*{k}{1}$ ou $\setalge*{k}{2}$`	$\mathbb{k}_1$ ou $\mathbb{k}_2$

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)*.

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

7.2.2 Ensembles classiques suffixés

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

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

Nous avons utilisé les suffixes `n` pour `n`-égatif, `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 `$\CCn$` 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 2e 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.

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

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

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.

```
$\intervalPC{a}{b} = \intervalOC{a}{b}$
   et
$\intervalP{a}{b} = \intervalO{a}{b}$.
```

$(a\,;b] =]a\,;b]$ 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.

```
$\ZintervalC{-1}{4}
 = \{ -1 ; 0 ; 1 ; 2 ; 3 ; 4 \}$

$\ZintervalC{-1}{4}
 = \ZintervalO{-2}{5}$.
```

$[\![-1\,;4]\!] = \{-1\,;0\,;1\,;2\,;3\,;4\}$
$[\![-1\,;4]\!] =]\!]-2\,;5[\!]$.

7.5 Unions et intersections

Exemple d'unions

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

```
$A \cup B$

$\dcup_{k=1}^{n} A_k$

$\bigcup_{k=1}^{n} B_k$

$\displaystyle \bigcup_{k=1}^{n} C_k$
```

$A \cup B$
$\dcup_{k=1}^{n} A_k$
$\bigcup_{k=1}^{n} B_k$
$\displaystyle \bigcup_{k=1}^{n} C_k$

Exemple d'unions disjointes

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

```
$A\sqcup B$

$\dsqcup_{k=1}^{n} A_k$

$\bigsqcup_{k=1}^{n} B_k$

$\displaystyle \bigsqcup_{k=1}^{n} C_k$
```

$A \sqcup B$
$\dsqcup_{k=1}^{n} A_k$
$\bigsqcup_{k=1}^{n} B_k$
$\displaystyle \bigsqcup_{k=1}^{n} C_k$

Exemple d'intersections

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

```
$A \cap B$

$\dcap_{k=1}^{n} A_k$

$\bigcap_{k=1}^{n} B_k$

$\displaystyle \bigcap_{k=1}^{n} C_k$
```

$A \cap B$

$\dcap_{k=1}^{n} A_k$

$\bigcap_{k=1}^{n} B_k$

$\displaystyle \bigcap_{k=1}^{n} C_k$

7.6 Cardinal, image et compagnie

Cardinal – Un exemple

```
$\card* E = \card E$
```

$\card* E = \card E$

Image et compagnie – Un exemple

```
$\ker f$ , $\dom f$ ,
$\im f$ ou $\codom f$
```

$\ker f$, $\dom f$, $\im f$ ou $\codom f$

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 [8] *(on parle aussi d'applications, sans qualificatif, et de fonctions)*.

Applications totales – Un exemple complet

```
$f: A \to B$ est une application totale, c'est à dire définie sur $A$ tout entier.

$i: C \onetoone D$ est une application totale injective.

$s: E \onto F$ est une application totale surjective.

$b: G \biject H$ est une application totale bijective.
```

$f: A \to B$ est une application totale, c'est à dire définie sur A tout entier.

$i: C \onetoone D$ est une application totale injective.

$s: E \onto F$ est une application totale surjective.

$b: G \biject H$ est une application totale bijective.

8. $a : E \to F$ est une application totale si $\forall x \in E$, $\exists ! y \in F$ tel que $y = a(x)$. Plus généralement, $f : E \to F$ 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.
```

$f : A \pto B$ 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.

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`-éometrie 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.

```
$\hgline{A}{B}$ ,
$\hgline{\pt{A}}{\pt{B}}$ ou
$\phgline{A}{B}$
```

$[AB)$, [AB) ou [AB)

Exemple 4 – D'autres demi-droites

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

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

$(AB]$, (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 LaTeX.

```
$\pgline{A}{B} \parallel \pgline{C}{D}$
au lieu de
$\pgline{A}{B}
 \stdparallel \pgline{C}{D}$

$\pgline{E}{F} \nparallel \pgline{G}{H}$
au lieu de
$\pgline{E}{F}
 \stdnparallel \pgline{G}{H}$
```

(AB) // (CD) au lieu de (AB) ∥ (CD)
(EF) ⫽ (GH) au lieu de (EF) ∦ (GH)

8.2 Vecteurs

8.2.1 Les écrire

Exemple d'utilisation 1

```
$\vect{ABCDEF}$  ,
$\vect*{e}{rot}$ ou
$\vect{e_{rot}}$
```

$\overrightarrow{ABCDEF}$, $\vec{e}_{rot}$ ou $\overrightarrow{e_{rot}}$

Exemple d'utilisation 2

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

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

8.2.2 Norme

Exemple d'utilisation

```
$\norm{\vect{i}}$

$\displaystyle
 \norm{\frac{2}{7} \vect*{e}{k}}$

$\displaystyle
 \norm*{\frac{2}{7} \vect*{e}{k}}$
```

$\|\vec{i}\|$

$\left\|\frac{2}{7}\vec{e}_k\right\|$

$\|\frac{2}{7}\vec{e}_k\|$

Remarque. Le code LaTeX vient directement de ce message : `https://tex.stackexchange.com/a/43009/6880`.

8.2.3 Produit scalaire – Écriture minimaliste

Exemple d'utilisation - Version longue

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

$\dfrac{1}{2}\vec{i}\cdot\vec{j}$

Exemple d'utilisation - Version courte mais restrictive

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

```
$\vdotprod{i}{j}$
```

$\vec{i}\cdot\vec{j}$

8.2.4 Produit scalaire – Écriture « à la physicienne »

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

```
$\displaystyle
 \adotprod{\frac{1}{2} \vect{i}}%
          {\vect{j}}$

$\displaystyle
 \adotprod*{\frac{1}{2} \vect{i}}%
           {\vect{j}}$

$\vadotprod{i}{j}$

$\vadotprod*{i}{j}$
```

$\left\langle \frac{1}{2}\vec{i} \middle| \vec{j} \right\rangle$

$\langle \frac{1}{2}\vec{i} \mid \vec{j} \rangle$

$\langle \vec{i} \mid \vec{j} \rangle$

$\langle \vec{i} \mid \vec{j} \rangle$

8.2.5 Produit vectoriel

Exemple d'utilisation - Version longue

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

$$\frac{1}{2}\vec{\imath} \wedge \vec{\jmath}$$

Exemple d'utilisation - Version courte mais restrictive

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

$$\vec{\imath} \wedge \vec{\jmath}$$

Exemple d'utilisation - Explication des calculs

Dans l'exemple suivant, le préfixe `c`, qui double cette lettre au début, est pour `c`-oordonnées et `v` pour `v`-ecteur.

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

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix} = \begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \\ z & z' \\ x & x' \end{vmatrix}$$

8.2.6 Plan – Déterminant de deux vecteurs

Exemple d'utilisation - Version décorée

Dans l'exemple suivant, le préfixe `c` est pour `c`-oordonnées.

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

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$$

Exemple d'utilisation - Version non décorée

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

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$$

Exemple d'utilisation - Versions courtes

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

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

$$\begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix} = \begin{vmatrix} \vec{u} & \vec{v} \\ x & x' \\ y & y' \end{vmatrix}$$

8.3 Coordonnées

Exemple d'utilisation 1

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

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

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
$\vect{i} \coord{3 | -4 | 0}$
```

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

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)*.

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

$(\mathrm{O}\,;\mathrm{I},\mathrm{J})$

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

$\displaystyle \left(\mathrm{O}\,;\frac{7}{3}\vec{\imath},\vec{\jmath}\right)$ ou $\displaystyle \left(\mathrm{O}\,;\frac{7}{3}\vec{\imath},\vec{\jmath}\right)$

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 $n > 0$.

```
$\axes{\pt{O} %
     | \vect*{i}{1} %
     | \vect*{i}{2} %
     | \vect*{i}{3} %
     | \dots %
     | \vect*{i}{9} %
     | \vect*{i}{10} %
     | \vect*{i}{11} %
     | \vect*{i}{12}}$
```

$(\mathrm{O}\,;\vec{\imath}_1,\vec{\imath}_2,\vec{\imath}_3,\ldots,\vec{\imath}_9,\vec{\imath}_{10},\vec{\imath}_{11},\vec{\imath}_{12})$

Exemple d'utilisation 4 – Repère affine

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

```
$\paxes{O | I | J | K}$
au lieu de
$\axes{\pt{O} %
     | \pt{I} | \pt{J} | \pt{K}}$
```

$(\mathrm{O}\,;\mathrm{I},\mathrm{J},\mathrm{K})$ au lieu de $(\mathrm{O}\,;\mathrm{I},\mathrm{J},\mathrm{K})$

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

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

```
$\vaxes{\pt{O} | i | j}$
au lieu de
$\axes{\pt{O} | \vect{i} | \vect{j}}$
```

$(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$ au lieu de $(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$

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`.

```
$\pvaxes{O | i | j}$
au lieu de
$\axes{\pt{O} | \vect{i} | \vect{j}}$
```

$(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$ au lieu de $(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$

8.5 Arcs circulaires

Exemple d'utilisation 1

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

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

Exemple d'utilisation 2

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

$\widehat{\imath}$ ou $\widehat{\jmath}_2$

8.6 Angles

8.6.1 Angles géométriques « intérieurs »

Exemple d'utilisation 1

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

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

Exemple d'utilisation 2

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

$\widehat{i}$ ou $\widehat{j_2}$

8.6.2 Angles orientés de vecteurs

Sans chapeau - Version longue

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

$\left(\frac{1}{2}\vec{\imath}\,;\vec{\jmath}\right)$ ou $(\frac{1}{2}\vec{\imath}\,;\vec{\jmath})$.

Sans chapeau - Version courte mais restrictive

Dans l'exemple suivant, le préfixe `v` est pour `v`-ecteur 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
$\vangleorient*{i}{j}$
```

$(\vec{\imath}\,;\vec{\jmath}\,)$ ou $\left(\vec{\imath}\,;\vec{\jmath}\right)$

Avec un chapeau

Dans l'exemple suivant, le préfixe `h` est pour `h`-at soit *« chapeau »* en anglais, et `v` pour `v`-ecteurr. Ceci permet d'avoir les angles orientés avec un chapeau *(la dernière écriture n'apporte rien de neuf mais elle evite de tout reprendre en cas de changement de mise en forme dans un document)*.

```
$\displaystyle
 \hangleorient{\frac{1}{2} \vect{i}}%
              {\vect{j}}$

$\displaystyle
 \hangleorient*{\frac{1}{2} \vect{i}}%
               {\vect{j}}$

$\hvangleorient{i}{j}$

$\hvangleorient*{i}{j}$
```

$\displaystyle \overgroup{\left(\frac{1}{2}\vec{\imath};\vec{\jmath}\right)}$

$\displaystyle \overgroup*{\left(\frac{1}{2}\vec{\imath};\vec{\jmath}\right)}$

$\hvangleorient{i}{j}$

$\hvangleorient*{i}{j}$

9 Analyse

9.1 Constantes

9.1.1 Constantes classiques

La liste complète

```
$\ggamma$ , $\ppi$ , $\ttau$ ,
$\ee$ , $\ii$ , $\jj$
et $\kk$ où $\ttau = 2 \ppi$
```

$\boldsymbol{\gamma}$, $\boldsymbol{\pi}$, $\boldsymbol{\tau}$, $\mathbf{e}$, $\mathbf{i}$, $\mathbf{j}$ et $\mathbf{k}$ où $\boldsymbol{\tau} = 2\boldsymbol{\pi}$

Remarque. Faites attention car `{\Large $\ppi \neq \pi$}` produit $\boldsymbol{\pi} \neq \pi$. 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

```
$\ct{a} x^2 + \ct{b} x + \ct{c}$
ou
$a x^2 + b x + c$
```

$\mathbf{a}x^2 + \mathbf{b}x + \mathbf{c}$ ou $ax^2 + bx + c$

9.2 La fonction valeur absolue

Un exemple d'utilisation

```
$\abs{2}$ ,
$\displaystyle \abs{\frac{3}{5}}$ ou
$\displaystyle \abs*{\frac{3}{5}}$
```

$|2|$, $\left|\frac{3}{5}\right|$ ou $|\frac{3}{5}|$

Remarque. Le code LaTeX 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 soit *« français »* en anglais *(ce choix a été fait pour éviter des incompatibilités avec quelques autres packages)*. La liste complète des fonctions nommées est donnée un peu plus bas dans une section dédiée.

```
$\fch x \neq ch x$ ,
$\ppcm(x;y)$ ou
$\lg x$
```

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

9.3.2 Avec un paramètre

Un exemple d'utilisation

```
$\logb{2} x = \lg x$ ou
$\expb{6} y = 6^y$
```

$\logb{2} x = \lg x$ ou $\expb{6} y = 6^y$

9.3.3 Toutes les fonctions nommées en plus

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

9.4 Des notations complémentaires pour des suites spéciales

Exemple d'utilisation

```
$\seqplus{F}{1}{2}$

$\seqhypergeo{F}{1}{2}$

$\seqsuprageo{F}{1}{2}{3}{4}$
pour les fous\dots :-)
```

$\seqplus{F}{1}{2}$

$\seqhypergeo{F}{1}{2}$

$\seqsuprageo{F}{1}{2}{3}{4}$
pour les fous... :-)

9.5 Calcul différentiel

9.5.1 Les opérateurs ∂ et d

Exemple d'utilisation

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

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

9.5.2 Dérivation totale

Exemple d'utilisation 1

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

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

Exemple d'utilisation 2

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

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

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

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

Exemple d'utilisation 3

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

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

Exemple d'utilisation 4

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

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

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

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

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 – 1ère version

Exemple d'utilisation 1

Il faut savoir que `hook` signifie *« crochet »* en anglais mais la bonne traduction est en fait *« square bracket »*.

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

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

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

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

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

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

9.6.2 L'opérateur crochet – 2nde version

Exemple d'utilisation 1

Ci-dessous `v` est pour v-ertical.

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

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

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.

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

$$\frac{x-1}{5+x^2}\bigg|_a^b = \frac{x-1}{5+x^2}\Big|_a^b$$

9.6.3 Intégrales multiples

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

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

$$\iiint F(x\,;y\,;z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \int_a^b\int_c^d\int_e^f F(x\,;y\,;z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$$

Remarque. Par défaut, LaTeX affiche $\int\int\int F(x\,;y\,;z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \int_a^b\int_c^d\int_e^f F(x\,;y\,;z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$. Nous avons obtenu ce résultat en utilisant `\stdint` qui est l'opérateur proposé de façon standard par LaTeX.

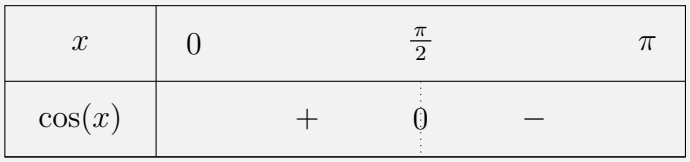
9.7 Tableaux de variation et de signe

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.

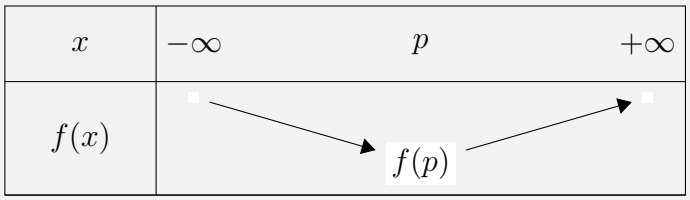
Un exemple de tableaux de signes

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

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



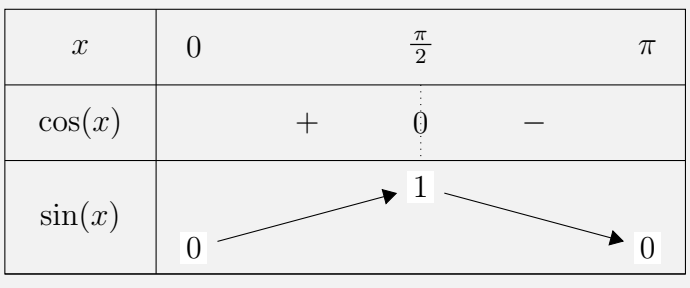
Un exemple de tableaux de variation

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

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



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}
```



9.8 Comparaison asymptotique de suites et de fonctions

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

Exemple d'utilisation 1

Les notations suivantes sont dues à Landau.

`$\bigO{}$ ou` `$\smallO{}$`	$\mathcal{O}$ ou o

Exemple d'utilisation 2

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

9.8.2 La notation Ω

Exemple d'utilisation 1

La notation suivante est due à Hardy et Littlewood.

`$\bigomega{}$`	$\boldsymbol{\Omega}$

Exemple d'utilisation 2

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

`$f(n) = \bigomega{g(n)}$`	$f(n) = \boldsymbol{\Omega}(g(n))$

9.8.3 La notation Θ

Exemple d'utilisation 1

`$\bigtheta{}$`	$\boldsymbol{\Theta}$

Exemple d'utilisation 2

Dans l'exemple suivant, $f(n) = \boldsymbol{\Theta}(g(n))$ signifie : $\exists(m, M, n_0)$ tel que $mg(n) \leqslant f(n) \leqslant Mg(n)$ dès que $n \geqslant n_0$.

`$f(n) = \bigtheta{g(n)}$`	$f(n) = \boldsymbol{\Theta}(g(n))$

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 1re notation, qui est devenue standard, permet de comprendre l'ordre des arguments.

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

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

Choisir le nom de la probabilité

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

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

10.3 Espérance

Un exemple type

`expval` vient de `exp`-ected `val`-ue soit *« espérance »* en anglais.

```
$\expval{X}$
```

$\mathrm{E}(X)$

Choisir le nom de l'espérance

```
$\expval[E_1]{X}$
```

$\mathrm{E}_1(X)$

10.4 Arbres pondérés

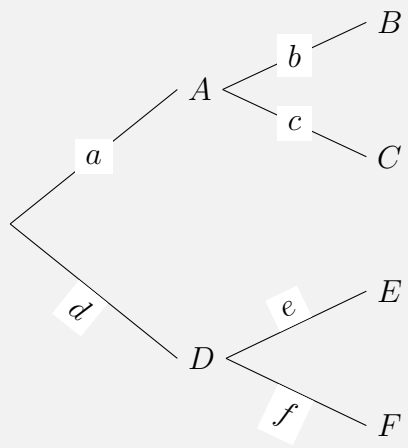
Que se passe-t-il en coulisse ?

Le gros du travail est fait par le package `forest` qui utilise `TiKz`. Ceci permet de faire des choses sympathiques comme dans le 2e exemple ci-dessous.

Exemple 1 – Le cas 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 [9].

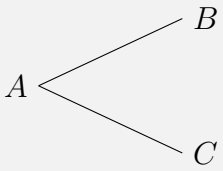
```
\begin{probatree}
    [
        [$A$, pweight = $a$
            [$B$, pweight = $b$]
            [$C$, pweight = $c$]
        ]
        [$D$, bpweight = $d$
            [$E$, apweight = $e$]
            [$F$, bpweight = $f$]
        ]
    ]
\end{probatree}
```



Exemple 2 – Des 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*}
```

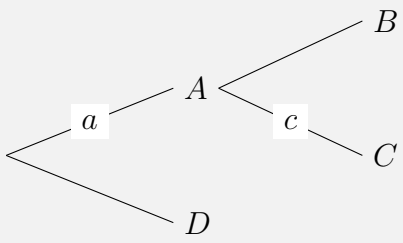


Exemple 3 – Des poids cachés localement

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

9. `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.

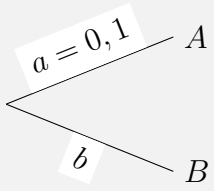
```
\begin{probatree}
    [
        [$A$, pweight = $a$
            [$B$, pweight* = $b$]
            [$C$, pweight  = $c$]
        ]
        [$D$, pweight* = $d$]
    ]
\end{probatree}
```



Exemple 4 – Un signe = et/ou 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.

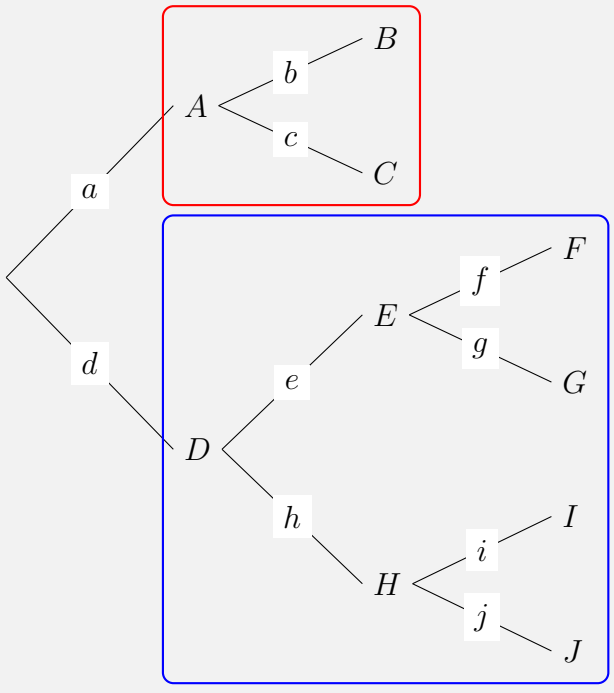
```
\begin{probatree}
    [
        [$A$, apweight = {$a = 0,1$}]
        [$B$, bpweight = $b$]
    ]
\end{probatree}
```



Exemple 5 – Des cadres 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.

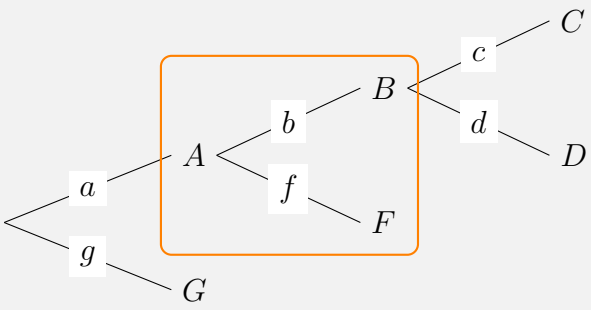
```
\begin{probatree}
    [{}, s sep = 1.3cm
     % Astuce pour espacer les cadres.
        [$A$, pweight = $a$,
              frame   = red
            [$B$, pweight = $b$]
            [$C$, pweight = $c$]
        ]
        [$D$, pweight = $d$,
              frame   = blue
            [$E$, pweight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



Exemple 6 – Des cadres faits 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    = nA
            [$B$, pweight = $b$,
                  name    = nB
                [$C$, pweight = $c$]
                [$D$, pweight = $d$]
              ]
            [$F$, pweight = $f$,
                  name    = nF]
        ]
        [$G$, pweight = $g$]
    ]
    \node[draw = orange,
          thick,
          rounded corners,
          fit = (nA)(nB)(nF)] {};
\end{probatree}
```



11 Arithmétique

11.1 Opérateurs de base

Pour des raisons d'expressivité des codes LaTeX, 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`.

```
$10 \divides 150$ au lieu de
$10 | 150$

$10 \notdivides 154$ au lieu de
$10 \not| 154$

$a \equiv b \modulo p
 \iff
 p \divides (a - b)$.
```

$10 \mid 150$ au lieu de $10|150$
$10 \nmid 154$ au lieu de $10 \not| 154$
$a \equiv b \bmod p \iff p \mid (a - b)$.

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}$
```

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

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}$
```

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

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 :-)
```

$\cfrac{a}{\left|b\right.}$ 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.

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

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

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

Exemple d'utilisation 2

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

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

12 Algèbre

12.1 Polynômes, séries formelles et compagnie

12.1.1 Polynômes et fractions polynômiales

Exemple d'utilisation 1 : Polynômes

```
$\setpoly{\RR}{X}$ ou
$\setpoly{\RR}{X | Y | Z}$
```

$\setpoly{\RR}{X}$ ou $\setpoly{\RR}{X | Y | Z}$

Exemple d'utilisation 2 : Fractions polynômiales

```
$\setpolyfrac{\QQ}{T}$ ou
$\setpolyfrac{\QQ}%
             {S_1 | S_2 | \dots | S_k}$
```

$\setpolyfrac{\QQ}{T}$ ou $\setpolyfrac{\QQ}{S_1 | S_2 | \dots | S_k}$

12.1.2 Séries formelles et leurs corps de fractions

Exemple d'utilisation 1 : Séries formelles

```
$\setserie{\CC}{X}$ ou
$\setserie{\CC}{T | O | P}$
```

$\setserie{\CC}{X}$ ou $\setserie{\CC}{T | O | P}$

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

```
$\setseriefrac{\ZZ}{X}$ ou
$\setseriefrac{\ZZ}{Z | T | O | P}$
```

$\setseriefrac{\ZZ}{X}$ ou $\setseriefrac{\ZZ}{Z | T | O | P}$

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.

```
$\setpolylaurent{\RR}{X}
 = \setpoly{\RR}{X | X^{-1}}$

$\setpolylaurent{\RR}{X_1 | X_2}
 = \setpoly{\RR}%
           {X_1 | X_1^{-1} %
            | X_2 | X_2^{-1}}$
```

$\setpolylaurent{\RR}{X} = \setpoly{\RR}{X | X^{-1}}$

$\setpolylaurent{\RR}{X_1 | X_2} = \setpoly{\RR}{X_1 | X_1^{-1} | X_2 | X_2^{-1}}$

Exemple d'utilisation 2 : Séries formelles de Laurent

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

```
$\setserielaurent{\QQ}{X}
 = \setserie{\QQ}{X | X^{-1}}$

$\setserielaurent{\QQ}{X_1 | X_2}
= \setserie{\QQ}%
           {X_1 | X_1^{-1} %
           | X_2 | X_2^{-1}}$
```

$\setserielaurent{\QQ}{X} = \setserie{\QQ}{X | X^{-1}}$

$\setserielaurent{\QQ}{X_1 | X_2} = \setserie{\QQ}{X_1 | X_1^{-1} | X_2 | X_2^{-1}}$

12.2 Matrices

Tout le boulot ou presque est fait par l'excellent package `nicematrix` auquel on impose l'option `transparent` avec l'ajout d'une macro *« maison »* à but pédagogique. Veuillez vous reporter à la documentation de `nicematrix` pour savoir comment s'y prendre en général.

12.2.1 Calculs expliqués des déterminants 2×2

Dans l'exemple suivant, le préfixe `c` est pour `c`-alculer et `two` est pour *« deux »* en anglais.

```
$\cdettwo{a}{b}%
         {c}{d}$
```

$\cdettwo{a}{b}{c}{d}$

12.2.2 Quelques exemples pour bien démarrer

Exemple 1 (cf. 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

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

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

Exemple 2

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

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

Exemple 3 (cf. la documentation de `nicematrix`)

```
$\begin{pNiceMatrix}[name = mymatrix]
    1 & 2 & 3 \\
    4 & 5 & 6 \\
    7 & 8 & 9
\end{pNiceMatrix}$

\tikz[remember picture,
      overlay]
\draw[red] (mymatrix-2-2) circle (2.5mm);
```

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

Exemple 4 (cf. la documentation de `nicematrix`)

```
$\left(
    \begin{NiceArray}{CCCC:C}
        1  & 2  & 3  & 4  & 5  \\
        6  & 7  & 8  & 9  & 10 \\
        11 & 12 & 13 & 14 & 15
    \end{NiceArray}
\right)$
```

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

Exemple 5 proposé par l'auteur de `nicematrix` suite à une discussion par mail

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

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

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

Exemple 6 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 dû être 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` 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 LaTeX.
- 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 `\existsone` 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. Ceci s'accompagne de la suppression des macros obsolètes `\explain*` et `\textexplainspacebefore`.
- 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 Quelques modifications générales

14.1.1 Deux séparateurs d'arguments par défaut

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

14.1.2 Espace et fractions

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

`\stdfrac <macro> (2 Arguments)`
`\stddfrac <macro> (2 Arguments)`

— `Argument 1:` le numérateur.

— `Argument 2:` le dénominateur.

14.1.3 Espace et racines n-ièmes d'un réel

`\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.

14.1.4 Espace après la négation logique

`\neg <macro> (Sans argument)`
`\stdneg <macro> (Sans argument)`

14.1.5 Sommes et produits en mode ligne

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)`

14.2 Logique et fondements

14.2.1 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)

14.2.2 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)

14.2.3 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)`

14.2.4 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)`

14.2.5 Des versions alternatives du quantificateur ∃

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

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

`\existsone <macro> (Sans argument)`
`\nexistsone <macro> (Sans argument)`

14.2.6 Détailler un raisonnement simple

`explain <env> [1 Option]`
`aexplain <env> [1 Option]` où `a = a-rrow`
`aexplain* <env> [1 Option]` où `a = a-rrow`

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

ATTENTION ! *La macro* `\explnext` *est à utiliser sans argument au tout début de l'environnement* `aexplain*`.

`\explnext <macro> [1 Option] (1 Argument)` où `expl = expl-ain`

— `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.

`\explnext* <macro> [1 Option] (2 Arguments)` où `expl = expl-ain`

— `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 1:` le texte de l'explication pour la 1re ligne. Ce texte peut être vide *(voir l'environnement* `aexplain` *pour la raison de ceci).*

— `Argument 2:` le texte de l'explication pour la 2e ligne. Ce texte peut être vide *(voir l'environnement* `aexplain` *pour la raison de ceci).*

14.2.7 Détailler un raisonnement simple – Mise en forme du texte

Les macros suivantes sont juste utilisées par l'environnement `explain`.

`\expltxtspacein <macro> (Sans argument)`

`\expltxt <macro> (1 Argument)` où `expl = expl-ain`

— `Argument:` le texte de l'explication que l'on veut mettre en forme.

`\expltxtdown <macro> (1 Argument)` où `expl = expl-ain`

— `Argument:` le texte de l'explication du haut vers le bas que l'on veut mettre en forme.

`\expltxtup <macro> (1 Argument)` où `expl = expl-ain`

— `Argument:` le texte de l'explication du bas vers le haut que l'on veut mettre en forme.

14.2.8 Détailler un « vrai » raisonnement via un tableau

`demoexplain <env> [4 Options]`

— `Option` *`"start"`*: le début de la numérotation des identifiants des justifications. La valeur par défaut est `1` et la valeur spéciale `last` permet de reprendre la numérotation là où elle s'était arrêtée le dernier environnement `demoexplain` ou `demoexplain*` utilisé.

— `Option` *`"hyps"`*: les hypothèses, au format texte, vérifiées au départ. Cet argument peut être vide et ne doit pas rentrer en conflit avec l'option `hyp`.

— `Option` *`"hyp"`*: une unique hypothèse, au format texte, vérifiée au départ. Cet argument peut être vide et ne doit pas rentrer en conflit avec l'option `hyps`.

— `Option` *`"ccl"`*: la conclusion, au format texte, du raisonnement détaillé. Cet argument peut être vide.

`demoexplain* <env> [1 Option]`

— `Option` *`"start"`*: le début de la numérotation des identifiants des justifications. La valeur par défaut est `1` et la valeur spéciale `last` permet de reprendre la numérotation là où elle s'était arrêtée le dernier environnement `demoexplain` ou `demoexplain*` utilisé.

`\demostep <macro> [1 Option] (Sans argument)`

— `Option:` un texte qui sera utilisé comme label global référençant le numéro d'une justification.

`\explref <macro> (1 Argument)` où `expl = expl-ain` et `ref = ref-erence`

— `Argument:` un numéro de 1 ou 2 chiffres qui sera encadré comme le sont les numérotations des indications.

`\explref* <macro> (1 Argument)` où `expl = expl-ain` et `ref = ref-erence`

— `Argument:` un texte correspondant à un label global référençant le numéro d'une justification.

14.3 Ensembles et applications

14.3.1 Ensembles versus accolades

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

— `Argument:` la définition de l'ensemble.

14.3.2 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 $\mathscr{U}$ dans le nom $\mathscr{U}_d$ d'un ensemble géométrique.

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

14.3.3 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.

14.3.4 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.

14.3.5 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)`

14.3.6 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`.

14.3.7 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 $]a\,;b]$.

14.3.8 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 $(a\,;b]$.

14.3.9 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.

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

— `Argument 1:` borne inférieure a de l'intervalle $[\![a\,;b[\![$.

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

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

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

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

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

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

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

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

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

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

14.3.10 Unions et intersections

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

14.3.11 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)`

14.3.12 Application totale, partielle, injective, surjective et/ou bijective

`\to <macro> (Sans argument)`
`\onetoone <macro> (Sans argument)`
`\onto <macro> (Sans argument)`
`\biject <macro> (Sans argument)`
`\pto <macro> (Sans argument)`
`\ponetoone <macro> (Sans argument)`
`\ponto <macro> (Sans argument)`
`\pbiject <macro> (Sans argument)`

14.4 Géométrie

14.4.1 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.

14.4.2 Lignes

`\gline <macro> [1 Option] (2 Arguments)` où `g = g-eometry`
`\pgline <macro> [1 Option] (2 Arguments)` où `p = p-oint` et `g = g-eometry`

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

— `Argument 1:` le 1er point géométrique.

— `Argument 2:` le 2e point géométrique.

`\hgline <macro> (2 Arguments)` où `h = h-alf` et `g = g-eometry`
`\phgline <macro> (2 Arguments)` où `p = p-oint`, `h = h-alf` et `g = g-eometry`
`\segment <macro> (2 Arguments)`
`\psegment <macro> (2 Arguments)` où `p = p-oint`

— `Argument 1:` le 1er point géométrique.

— `Argument 2:` le 2e point géométrique.

14.4.3 Droites parallèles ou non

`\parallel <macro> (Sans argument)`
`\nparallel <macro> (Sans argument)`
`\stdparallel <macro> (Sans argument)`
`\stdnparallel <macro> (Sans argument)`

14.4.4 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.

14.4.5 Norme

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

— `Argument:` le vecteur sur lequel appliquer la norme.

14.4.6 Produit scalaire – Écriture minimaliste

`\dotprod <macro> (2 Arguments)`

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

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

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

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

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

14.4.7 Produit scalaire – Écriture « à la physicienne »

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

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

— `Argument 2:` le 2e 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 1er vecteur sans utiliser la macro `\vect`.

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

14.4.8 Produit vectoriel

`\crossprod <macro> (2 Arguments)`

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

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

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

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

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

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

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

— `Argument 2` : la 1re coordonnée du 1er vecteur.

— `Argument 3` : la 2e coordonnée du 1er vecteur.

— `Argument 4` : la 3e coordonnée du 1er vecteur.

— `Argument 5` : le 2e vecteur qu'il faut taper via la macro `\vect`.

— `Argument 6` : la 1re coordonnée du 2e vecteur.

— `Argument 7` : la 2e coordonnée du 2e vecteur.

— `Argument 8` : la 3e coordonnée du 2e vecteur.

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

— `Argument 1` : le 1er vecteur sans utiliser la macro `\vect`.

— `Argument 2` : la 1re coordonnée du 1er vecteur.

— `Argument 3` : la 2e coordonnée du 1er vecteur.

— `Argument 4` : la 3e coordonnée du 1er vecteur.

— `Argument 5` : le 2e vecteur sans utiliser la macro `\vect`.

— `Argument 6` : la 1re coordonnée du 2e vecteur.

— `Argument 7` : la 2e coordonnée du 2e vecteur.

— `Argument 8` : la 3e coordonnée du 2e vecteur.

14.4.9 Plan – Déterminant de deux vecteurs

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

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

— `Argument 2` : la 1re coordonnée du 1er vecteur.

— `Argument 3` : la 2e coordonnée du 1er vecteur.

— `Argument 4` : le 2e vecteur qu'il faut taper via la macro `\vect`.

— `Argument 5` : la 1re coordonnée du 2e vecteur.

— `Argument 6` : la 2e coordonnée du 2e vecteur.

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

— `Argument 1` : le 1er vecteur sans utiliser la macro `\vect`.

— `Argument 2` : la 1re coordonnée du 1er vecteur.

— `Argument 3` : la 2e coordonnée du 1er vecteur.

— `Argument 4` : le 2e vecteur sans utiliser la macro `\vect`.

— `Argument 5` : la 1re coordonnée du 2e vecteur.

— `Argument 6` : la 2e coordonnée du 2e vecteur.

14.4.10 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.

14.4.11 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 `pv = 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.

14.4.12 Arcs circulaires

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

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

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

— `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.

14.4.13 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.

14.4.14 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`.

14.5 Analyse

14.5.1 Constantes classiques

`\ggamma <macro> (Sans argument)`
`\ppi <macro> (Sans argument)`
`\ttau <macro> (Sans argument)`
`\ee <macro> (Sans argument)`
`\ii <macro> (Sans argument)`
`\jj <macro> (Sans argument)`
`\kk <macro> (Sans argument)`

14.5.2 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.

14.5.3 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)` où `f = f-rench`
`\fsh <macro> (Sans argument)` où `f = f-rench`
`\fth <macro> (Sans argument)` où `f = f-rench`

14.5.4 Avec un paramètre

`\expb <macro> (1 Argument)`

— `Argument:` la base de l'exponentielle

`\logb <macro> (1 Argument)`

— `Argument:` la base du logarithme

14.5.5 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.

`\seqsuprageo <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.

14.5.6 Calcul différentiel

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

— `Option:` utilisée, cette option sera mise en exposant du symbole ∂ ou d.

— `Argument:` la variable de différentiation à droite du symbole ∂ ou d.

14.5.7 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.

14.5.8 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 ∂.

— `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.

14.5.9 L'opérateur crochet – 1ère 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.

14.5.10 L'opérateur crochet – 2nde version

`\vhook <macro> (3 Arguments)` où `v = v-ertical`
`\vhook* <macro> (3 Arguments)` où `v = v-ertical`

— `Argument 1:` le contenu avant le trait vertical.

— `Argument 2:` la borne inférieure affichée en indice.

— `Argument 3:` la borne supérieure affichée en exposant.

14.5.11 L'opérateur d'intégration standard

`\stdint <macro> (Sans argument)`

14.5.12 Les notations $\mathcal{O}$ et 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 o.

14.5.13 La notation Ω

`\bigomega <macro> (1 Argument)`

— `Argument:` si l'argument est vide, il est ignoré, sinon il est mis entre des parenthèses après $\mathbf{\Omega}$.

14.5.14 La notation Θ

`\bigtheta <macro> (1 Argument)`

— `Argument:` si l'argument est vide, il est ignoré, sinon il est mis entre des parenthèses après $\mathbf{\Theta}$.

14.6 Probabilité

14.6.1 Probabilité « simple »

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

— `Option:` le nom de la probabilité.

— `Argument:` l'ensemble dont on veut calculer la probabilité.

14.6.2 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.

14.6.3 Espérance

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

— `Option`: le nom de la fonction espérance.

— `Argument`: la variable aléatoire dont on veut calculer l'espérance.

14.6.4 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.

14.7 Arithmétique

14.7.1 Opérateurs de base

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

14.7.2 Fractions continuées standard

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

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

14.7.3 Fractions continuées généralisées

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

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

14.7.4 Comme une fraction continuée isolée

`\singlecontfrac <macro> (2 Arguments)`

— `Argument 1:` le pseudo numérateur.

— `Argument 2:` le pseudo dénominateur.

14.7.5 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)`

14.8 Algèbre

14.8.1 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.

14.8.2 Calculs expliqués des déterminants 2×2

`\cdettwo <macro> (4 Arguments)` où `c = c-alculate`

— `Argument 1:` l'entrée à la position $(1, 1)$

— `Argument 2:` l'entrée à la position $(1, 2)$

— `Argument 3:` l'entrée à la position $(2, 1)$

— `Argument 4:` l'entrée à la position $(2, 2)$