

Documentation

Le package `Preamble.sty`¹ ou `HTMLPreamble.sty`² doit être chargé pour pouvoir utiliser les autres qui sont donnés ci-dessous.

Les fichiers `.sty` doivent être placés dans le même répertoire que le fichier `.tex` qui est utilisé.

Pour charger un package (par exemple `NomDuPackage.sty`), il faut utiliser la commande `\usepackage{nomdupackage}` avant `\begin{document}`.

En utilisant `Preamble.sty` ou `HTMLPreamble.sty`, les packages suivant seront chargés :

- `\usepackage[utf8]{inputenc}`
- `\usepackage[french]{babel}`
- `\usepackage[T1]{fontenc}`
- `\usepackage{amsmath, amsfonts, amssymb}`
- `\usepackage{stmaryrd}`
- `\usepackage{adjustbox}` (pour `HTMLPreamble.sty`)
- `\usepackage{xcolor}` (pour `Preamble.sty`)

Il est nécessaire que `cm-super` soit installé (disponible sur [CTAN](#)) pour pouvoir utiliser `Preamble.sty`. Pour ne pas avoir à installer `cm-super`, il est possible de commenter les lignes `\usepackage{sffont}` et `\renewcommand{\sfdefault}{cmssp}` du fichier `Preamble.sty` en mettant `%` au début de chacune de ces lignes (numéro 10 et 11).

Lors de l'utilisation de BEAMER (avec une police sans-sérif), il est possible d'utiliser les commandes avec les polices sans-serif, sauf pour les lettres grecques (Ω , ϕ , φ , ...), la redéfinition du ℓ en mathématiques, les alphabets `\mathcal` et `\mathbb` ainsi que les symboles.

Il est possibles de changer les polices de caractères/symboles en important des packages après `\usepackage{preamble}`. Il peut être nécessaire de placer l'importation avant d'importer les autres modules décrit ci-dessous.

Il n'est pas possible d'utiliser en simultanée le package `Dsfont.sty` disponible sur [CTAN](#) et `Dsft.sty` décrit ci-dessous. De plus, la commande `\1` ne sera pas modifiée si un package définissant `\mathbb{1}` est importé. Il est alors possible de redéfinir la commande en utilisant `\newcommand\1[1]{\mathbb{1}_{\#1}}` (si `Dsft.sty` n'est pas importé) ou `\renewcommand\1[1]{\mathbb{1}_{\#1}}`.

Il est possible de redéfinir le « ℓ » à sa version d'origine (ℓ) avec :

```
\mathcode`1="8000
\begingroup
\makeatletter
```

1. Pour utiliser avec BEAMER
2. Pour les documents autres que BEAMER

```

\lccode`~=`\l
\DeclareMathSymbol{\lsb@l}{\mathalpha}{letters}{\l}
\lowercase{\gdef~{\lsb@l}}%
\endgroup
\makeatother

```

Pour utiliser des commandes avec des parenthèses automatiques (comme pour `sup`), il est possible de faire³ :

```

\let\oldsup\sup
\renewcommand{\sup}[1]{\oldsup\l#1\r}

```

`\l` et `\r` sont définis dans `Preamble.sty` et `HTMLPreamble.sty`.

La commande `\sup{\left\{x\in\mathbb{Q}\;;\middle|\;;x^2<2\right\}}` donne : $\sup(\{x \in \mathbb{Q} \mid x^2 < 2\})$.

L'ensembles des titres des sections (et sous-sections si le fichier n'est pas déjà dans une section) sont des liens qui pointent vers les fichiers en ligne pour un téléchargement direct. Il est aussi possible de télécharger la version la plus récente de ce fichier en cliquant sur le titre en page 1.

3. Cette commande est déjà définie dans `Usuelles.sty`

Table des matières

0	Documentation	1
1	Flashcards.py et Htmlcards.py	3
2	Preamble.sty et HTMLPreamble.sty	5
3	AL.sty	6
4	Analyse.sty	7
5	Arithmetique.sty	9
6	BigOperators.sty	10
7	Complexes.sty	11
8	Dsft.sty	12
9	Equivalents.sty	13
10	Matrices.sty	14
11	Polynomes.sty	17
12	Probas.sty	18
13	Sffont.sty	19
14	Structures.sty	20
15	Tables.sty	21
16	Tools.sty	22
17	Trigo.sty	23
18	Usuelles.sty	24

1 Flashcards.py et Htmlcards.py

Les fichiers `Flashcards.py` et `Htmlcards.py` permettent d'exporter facilement des flashcards en `.pdf` et `.svg` (pour affichage dans le navigateur).

Pour pouvoir créer des fiches de révision, il faut mettre un fichier `.txt` (décrit plus bas) dans un dossier `input` et mettre les fichiers `.sty` nécessaires dans un dossier `output`.

1.1 Flashcards.py

Pour exporter la fiche `fiche.txt`, il faut soit lancer le fichier python et entrer le nom du fichier (`fiche`), soit utiliser la commande `python Flashcards.py --file=fiche` (ou `python3`), à laquelle il est possible de rajouter les paramètres optionnels `--n=nombre` (avec le nombre d'exemplaires), `--dest=dossier` (avec le dossier où il faut mettre le `.pdf` produit) et `--open=True/False` (pour ouvrir le dossier où le `.pdf` est produit).

1.2 Htmlcards.py

Pour exporter la fiche `fiche.txt`, il faut soit lancer le fichier python et entrer le nom du fichier (`fiche`), soit utiliser la commande `python Flashcards.py --file=fiche` (ou `python3`), à laquelle il est possible de rajouter les paramètres optionnels `--dest=dossier` (avec le dossier où il faut mettre le `.pdf` produit) et `--open=True/False` (pour ouvrir le dossier où le `.pdf` est produit).

Modifier la valeur de `--dest` peut rendre inutilisable certaines fonctions liées au site pour visualiser les fiches.

1.3 Les options spéciales

Il est possible de compiler l'ensemble des fichiers `.txt` du dossier `input` en mettant `__compile_all__` comme nom de fichier.

Il est également possible de recompiler les Flashcards en utilisant `__recompile__` comme nom de fichier.

1.4 Les fiches .txt

Pour faire des fiches, il faut créer un fichier `.txt` de la forme

TITRE

Shuffle questions : True/False

Q/R & R/Q : True/False

PACKAGES & COMMANDES SUPPLÉMENTAIRES

QUESTION;;RÉPONSE

...

QUESTION;;RÉPONSE

Le titre doit être de la forme `Thème -- Chapitre` ou `Chapitre`. On peut aussi spécifier un titre raccourci pour le nom du fichier avec `Titre_raccourci!!titleTitre classique` où le titre raccourci ne peut pas contenir d'espaces ou de caractères spéciaux, et titre classique étant de la forme des deux premiers.

La ligne 2 indique si le programme peut ou non mettre un ordre aléatoire pour les questions.

La ligne 3 indique si le programme peut échanger l'ordre des questions et des réponses pour les fiches. Avec cette option à `True`, il est possible de forcer une question à être avant la réponse en mettant `!!fst` devant la/les ligne(s) concernée(s).

Les packages et commandes supplémentaires (voir [overleaf](#)) doivent être placées sur une seule ligne.

S'il y a une erreur lors de la compilation L^AT_EX, le programme python affichera le message d'erreur affiché par L^AT_EX.

Exemple de fiches : <https://github.com/rfoxinter/revisions/tree/main/input>.

1.5 Visionner les flashcards en svg (Htmlcards)

Pour pouvoir visionner les flashcards exportées en svg, il faut disposer d'un serveur web (comme [github](#) avec [github pages](#)) sur lequel le programme va mettre le dossier généré par `Htmlcards.py` (on suppose que l'url est <https://example.fr/dossier>).

Il faut alors convertir l'url du dossier en base64 (cette conversion peut se faire sur le site <https://www.base64encode.org/>, avec la fonction `btoa` de JavaScript ou avec la fonction Python `base64.b64encode`) en enlevant les « = » à la fin. Dans l'exemple, en exécutant le code Python suivant

```
from base64 import b64encode # encoder en base64
from re import sub # remplacer tous les '=' finaux
url = "https://example.fr/dossier"
base64url = sub("=", "", b64encode(url.encode()).decode())
print(base64url)
```

on obtient `aHR0cHM6Ly9leGFtcGxlLmZyL2Rvc3NpZXI`.

Il faut alors aller sur le site <https://rfoxinter.github.io/revisions/flashcards/> en rajoutant à la fin de l'url `?file=nom_du_dossier` où le nom du dossier correspond à celui en base64.

Dans l'exemple, on obtient l'url suivante :

[https://rfoxinter.github.io/revisions/flashcards/](https://rfoxinter.github.io/revisions/flashcards/?file=aHR0cHM6Ly9leGFtcGxlLmZyL2Rvc3NpZXI)
`?file=aHR0cHM6Ly9leGFtcGxlLmZyL2Rvc3NpZXI`

Il est sinon possible de mettre un lien vers un fichier téléchargé et hébergé sur un serveur (encodé en base64, et sans les « = » finaux) en ajoutant `?card=nom_du_fichier` à la fin de l'url.

1.6 Télécharger des Htmlcards depuis le site

Il est possible de faire en sorte que les cartes téléchargées puissent être mises à jour en ajoutant un fichier `cards.txt` à la racine du dossier des `Htmlcards`.

Ce fichier doit contenir en première ligne la racine à partir de laquelle sont données les url des `Htmlcards` (si l'url n'est pas absolue), puis plusieurs lignes (2 pour chacune des `Htmlcards`) contenant en premier le chemin vers la `Htmlcard` (relatif ou absolu, sachant que la racine est <https://rfoxinter.github.io/revisions/flashcards/>), suivi de la date de dernière mise à jour du dossier de la `Htmlcard` concernée (au format `%YYYY%MM%dd%hh%mm%ss` (année, mois, jour, heure, minutes, secondes)).

Exemple : <https://rfoxinter.github.io/revisions/MP2I/flashcards/cards.txt>.

Un exemple de fichier python générant un tel fichier est disponible à l'adresse suivante : <https://rfoxinter.github.io/revisions/CardsList.py>.

2 Preamble.sty et HTMLPreamble.sty

2.1 Commandes communes

Commande	Résultat
<code>\l</code> ⁴	(
<code>\r</code> ⁵)
<code>\llb</code> ⁶	[[
<code>\rrb</code> ⁷]]
<code>\oldfrac{a}{b}</code> ⁸	$\frac{a}{b}$
<code>\frac{a}{b}</code> ⁹	$\frac{a}{b}$
<code>\l</code> ¹⁰	ℓ
<code>\oldvec{x}</code> ¹¹	\vec{x}
<code>\vec{x}</code>	\vec{x}
<code>\overrightarrow{AB}</code> ¹²	\overrightarrow{AB}

2.2 Commandes de Preamble.sty

Commande	Résultat
<code>\slideq{Q1}{1}</code> ¹³	<div>Question 1</div> <div>Q1</div>
<code>\slider{R1}{1}</code> ¹⁴	<div>Réponse 1</div> <div>R1</div>

-
4. Correspond à la commande usuelle `\left(`
 5. Correspond à la commande usuelle `\right)`
 6. Correspond à la commande usuelle `\left[\llbracket`
 7. Correspond à la commande usuelle `\right[\rrbracket`
 8. Correspond à la commande usuelle `\frac`
 9. Correspond à la commande usuelle `\dfrac`
 10. Correspond à la commande usuelle `\ell`

Le ℓ peut être redéfini en l avec le code donné dans l'introduction

11. Correspond à la commande usuelle `\vec`
12. Le résultat est le même qu'avec `\vec`
13. Cette commande doit être utilisée entre `\begin{document}` et `\end{document}`
14. Cette commande doit être utilisée entre `\begin{document}` et `\end{document}`

3 AL.sty

Le package `Matrices.sty` sera importé automatiquement avec `AL.sty`.

Commande	Résultat
<code>\oldvect</code>	Vect
<code>\vect{E}</code>	$\text{Vect}(E)$
<code>\al{E}{}</code>	$\mathcal{L}(E)$
<code>\al{E}{F}</code>	$\mathcal{L}(E, F)$
<code>\oplus</code> ¹⁵	\oplus
<code>\matgl{n}{\mathbb{K}}</code> ¹⁶	$\text{GL}_n(\mathbb{K})$
<code>\gl{E}</code>	$\text{GL}(E)$
<code>\olddim</code> ¹⁷	\dim
<code>\dim{E}</code>	$\dim(E)$
<code>\oldrg</code>	rg
<code>\rg{u}</code>	$\text{rg}(u)$
<code>\oldtr</code>	tr
<code>\tr{u}</code>	$\text{tr}(u)$
<code>\oldmat</code>	Mat
<code>\almat{u}{\mathcal{B}}{}</code> ¹⁸	$\text{Mat}_{\mathcal{B}}(u)$
<code>\almat{u}{\mathcal{B}}{\mathcal{C}}</code>	$\text{Mat}_{\mathcal{B}, \mathcal{C}}(u)$
<code>\lc</code> ¹⁹	[
<code>\rc</code> ²⁰]
<code>\oldsl</code>	SL
<code>\sl{E}</code>	$\text{SL}(E)$
<code>\sl[n]{\mathbb{K}}</code>	$\text{SL}_n(\mathbb{K})$
<code>\oldorth</code>	O
<code>\orth{n}</code>	$\text{O}(n)$
<code>\orth[n]{\mathbb{R}}</code>	$\text{O}_n(\mathbb{R})$
<code>\oldso</code>	SO
<code>\so{n}</code>	$\text{SO}(n)$
<code>\so[n]{\mathbb{R}}</code>	$\text{SO}_n(\mathbb{R})$

15. Le `\oplus` utilisé est celui de `stmaryrd`

Pour récupérer celui de L^AT_EX, il est possible d'utiliser la commande `\let\oldoplus\oplus` avant `\usepackage{al}` puis de faire `\let\oplus\oldoplus` après importation
 Comparaison L^AT_EX - `stmaryrd` avec le plus normal $\oplus \oplus +$

16. Le `\matgl` de `AL.sty` correspond à la commande `\gl` de `Matrices.sty` qui a été renommé

17. Correspond à la commande usuelle `\dim`

18. Ne pas confondre cette commande avec `\mat` de `Matrices.sty`

19. Correspond à la commande usuelle `\left[`

20. Correspond à la commande usuelle `\right]`

4 Analyse.sty

Le package `BigOperators.sty` sera importé automatiquement avec `Analyse.sty`.

Commande	Résultat
<code>\dd</code> ²¹	d
<code>\der{}{f(x)}</code>	$\frac{d}{dx}(f(x))$
<code>\der[n]{}{f(x)}</code>	$\frac{d^n}{dx^n}(f(x))$
<code>\der[] [t]{f(t)}</code>	$\frac{df(t)}{dt}$
<code>\der[n] [t]{f}{t}</code> ²²	$\frac{d^n f}{dt^n}(t)$
<code>\pder{}{f(x,y)}</code> ²³	$\frac{\partial}{\partial x}(f(x,y))$
<code>\mpder[x,y,z]{}{f(x,y,z)}</code> ²⁴	$\frac{\partial^3}{\partial x \partial y \partial z}(f(x,y,z))$
<code>\mpder[x_1,x_2,x_3,x_3]{f(X)}</code> ²⁵	$\frac{\partial^4 f(X)}{\partial x_1 \partial x_2 \partial x_3^2}$
<code>\oldint</code> ²⁶	\int
<code>\int{f}</code>	$\int (f)$
<code>\int[t]{f(t)}</code>	$\int (f(t)) dt$
<code>\int[t] [{[a,b]}]{f(t)}</code> ²⁷	$\int_{[a,b]} (f(t)) dt$
<code>\int[t] [a] [b]{f(t)}</code>	$\int_a^b (f(t)) dt$
<code>\int[] [a] [b]{f'}</code>	$\int_a^b (f')$
<code>\eval[{[a,b]}]{f(t)}</code>	$[f(t)]_{[a,b]}$
<code>\eval[a] [b]{f(t)}</code>	$[f(t)]_a^b$

21. d de dérivation

22. Le parenthésage de l'expression dans le second argument se fait automatiquement si ce dernier est non vide

23. On peut appliquer les mêmes arguments optionnels que pour `\der` et les arguments obligatoires sont les mêmes que pour `\der`

24. Sans argument optionnel, `\mpder` agit comme `\pder` et les arguments obligatoires sont les mêmes que pour `\der`

25. `\mpder` peut ne pas marcher avec des variables de plusieurs lettres

Si la commande n'affiche pas le résultat attendu avec des variables de plusieurs lettres, il faut utiliser `\usepackage[dvar]{analyse}`

Cette option importe automatiquement `pgffor` (qui est utilisé importé par `TikZ` et `PGF`)

26. Correspond à la commande usuelle `\int`

27. L'argument `[a,b]` doit être mis entre accolades pour être traité correctement par \LaTeX

<code>\serie{a_n}</code> ²⁸	$\sum a_n$
<code>\oldesc</code>	Esc
<code>\esc{\left[a,b\right]}</code>	$\text{Esc}([a, b])$
<code>\esc{+}{f}</code>	$\text{Esc}_+(f)$
<code>\nrm{f}</code>	$\ f\ _\infty$
<code>\nrm[1]{g}</code>	$\ g\ _1$
<code>\oldva</code>	VA
<code>\va{u}</code>	$\text{VA}(u)$
<code>\oldepi</code>	Epi
<code>\epi{f}</code>	$\text{Epi}(f)$

28. Comme `BigOperators.sty` est chargé, la commande `\sum` est remplacée et il est nécessaire de taper `\oldsum` pour obtenir \sum

5 Arithmetique.sty

Commande	Résultat
<code>\olddiv</code> ²⁹	\div
<code>\div</code> ³⁰	$ $
<code>\cgr{a}{b}{n}</code>	$a \equiv b [n]$
<code>\oldphi</code> ³¹	ϕ
<code>\phi</code> ³²	φ

29. Correspond à la commande usuelle `\div`

30. Correspond à la commande usuelle `\mid`

31. Correspond à la commande usuelle `\phi`

32. Correspond à la commande usuelle `\varphi`

6 BigOperators.sty

Commande	Résultat
<code>\oldsum</code> ³³	\sum
<code>\sum{n=0}{+\infty}{u_n}</code>	$\sum_{n=0}^{+\infty} (u_n)$
<code>\oldprod</code> ³⁴	\prod
<code>\prod{n=0}{+\infty}{u_n}</code>	$\prod_{n=0}^{+\infty} (u_n)$
<code>\oldcap</code> ³⁵	\cap
<code>\bigcap{n=0}{+\infty}{A_n}</code>	$\bigcap_{n=0}^{+\infty} (A_n)$
<code>\oldcup</code> ³⁶	\cup
<code>\bigcup{n=0}{+\infty}{A_n}</code>	$\bigcup_{n=0}^{+\infty} (A_n)$
<code>\olduplus</code> ³⁷	\uplus
<code>\biguplus{n=0}{+\infty}{A_n}</code>	$\biguplus_{n=0}^{+\infty} (A_n)$
<code>\bigoplus{n=0}{+\infty}{E_n}</code>	$\bigoplus_{n=0}^{+\infty} (E_n)$

33. Correspond à la commande usuelle `\sum`

34. Correspond à la commande usuelle `\prod`

35. Correspond à la commande usuelle `\bigcap`

36. Correspond à la commande usuelle `\bigcup`

37. Correspond à la commande usuelle `\biguplus`

7 Complexes.sty

Commande	Résultat
<code>\oldbar{z}</code> ³⁸	\bar{z}
<code>\bar{z}</code> ³⁹	\bar{z}
<code>\e</code> ⁴⁰	e
<code>\i</code> ⁴¹	i
<code>\j</code> ⁴²	j
<code>\oldIm</code> ⁴³	\Im
<code>\Im</code>	\Im
<code>\pIm{x}</code>	$\Im(x)$
<code>\oldRe</code> ⁴⁴	\Re
<code>\Re</code>	\Re
<code>\pRe{x}</code>	$\Re(x)$

7.1 Dessiner des arcs

Il est aussi possible de faire des arcs en important le package `Complexes.sty` avec l'option `arc`, en utilisant `\usepackage[arc]{complexes}` au lieu de `\usepackage{complexes}`. Cette option importe automatiquement le package `graphics` (disponible sur [CTAN](#)).

Commande	Résultat
<code>\arc{AB}</code>	\widehat{AB}
<code>\arc{ABCDEFGFGH}</code>	$\widehat{ABCDEFGFGH}$

38. Correspond à la commande usuelle `\bar`

39. Se comporte comme `\overline`

40. e de la fonction exponentielle

41. i complexe

L'ancienne commande `\i` s'obtient avec `\ii`

42. $j = e^{\frac{2i\pi}{3}}$

L'ancienne commande `\j` s'obtient avec `\jj`

43. Correspond à la commande usuelle `\Im`

44. Correspond à la commande usuelle `\Re`

8 Dsft.sty

Ce package remplace le `\mathds{1}` du package `Dsfonts.sty` disponible sur [CTAN](#) et introduit quelques symboles.

Commande	Résultat
<code>\mathds{1}</code>	$\mathbb{1}$
<code>\1{E}(x)</code>	$\mathbb{1}_E(x)$
<code>\square</code>	\square
<code>\star</code>	\star
<code>\triangle</code>	\triangle

8.1 dsrom12.pfb et dsrom12.tfm

Pour utiliser ce package, il faut copier les fichiers `dsrom12.pfb` et `dsrom12.tfm` dans les dossiers où ils sont actuellement avec `dsfonts` (et éventuellement créer une copie des anciens fichiers).

9 Equivalents.sty

Commande	Résultat
$\backslash o\{x\}$	$o(x)$
$\backslash o[x\to 0]\{x\}$	$o(x)$ $x \rightarrow 0$
$\backslash O\{x\}$	$O(x)$
$\backslash O[x\to 0]\{x\}$	$O(x)$ $x \rightarrow 0$
$\backslash Th\{x\}$	$\Theta(x)$
$\backslash Th[x\to 0]\{x\}$	$\Theta(x)$ $x \rightarrow 0$
$\backslash Om\{x\}$	$\Omega(x)$
$\backslash Om[x\to 0]\{x\}$	$\Omega(x)$ $x \rightarrow 0$
$\backslash eq\{u_n\}\{v_n\}$	$u_n \sim v_n$
$\backslash eq[n\to+\infty]\{u_n\}\{v_n\}$	$u_n \underset{n \rightarrow +\infty}{\sim} v_n$
$\backslash eg\{u_n\}\{v_n+\backslash o\{v_n\}\}$	$u_n = v_n + o(v_n)$
$\backslash eg[n\to+\infty]\{u_n\}\{v_n+\backslash o\{v_n\}\}$	$u_n \underset{n \rightarrow +\infty}{=} v_n + o(v_n)$

10 Matrices.sty

Commande	Résultat
<code>\mat{n}{p}{\mathbb{K}}</code>	$\mathcal{M}_{n,p}(\mathbb{K})$
<code>\mat{n}{}{\mathbb{K}}</code>	$\mathcal{M}_n(\mathbb{K})$
<code>\sym{n}{\mathbb{K}}</code>	$\mathcal{S}_n(\mathbb{K})$
<code>\ant{n}{\mathbb{K}}</code>	$\mathcal{A}_n(\mathbb{K})$
<code>\diag{n}{\mathbb{K}}</code>	$\mathcal{D}_n(\mathbb{K})$
<code>\ts{n}{\mathbb{K}}</code>	$\mathcal{T}_n^+(\mathbb{K})$
<code>\ti{n}{\mathbb{K}}</code>	$\mathcal{T}_n^-(\mathbb{K})$
<code>\olddet</code> ⁴⁵	det
<code>\det{M}</code>	$\det(M)$
<code>\det[\mathcal{B}]{\mathcal{B}'}</code>	$\det_{\mathcal{B}}(\mathcal{B}')$
<code>\oldgl</code>	GL
<code>\gl{n}{\mathbb{K}}</code> ⁴⁶	$\mathrm{GL}_n(\mathbb{K})$
<code>\oldcom</code>	Com
<code>\com{M}</code>	$\mathrm{Com}(M)$
<code>\mdots</code>	...
<code>\ddots</code>	\ddots
<code>\idots</code>	\dotso
<code>\vdots</code>	\vdots
<code>\xdots</code>	\ddots
<code>\plusdots</code>	\cdots
<code>\tmatrix({1\&0\&0\&1})</code> ⁴⁷	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

10.1 Ne pas importer les commandes avec .

Il est possible de ne pas modifier les commandes usuelles en utilisant l'option `nodots` de ce package. Il faut alors importer le package avec `\usepackage[nodots]{matrices}`.

10.2 Modifier les séparations entre les .

Les commandes avec des points tel que \ddots ont des définitions qui dépendent de la taille de la police. Par défaut, celle pour L^AT_EX est adaptée pour 12pt, et celle de BEAMER pour 17pt. Pour avoir des points alignés correctement, il est possible de modifier la valeur de `\dotsep` en utilisant `\setlength{\dotsep}{Xpt}`.

Par exemple, avec 2pt, on obtient : « \ddots ».

⁴⁵. Correspond à la commande usuelle `\det`

⁴⁶. Si `AL.sty` est chargé, cette commande est remplacée et il faut utiliser `\matgl{n}{\mathbb{K}}` pour obtenir ce résultat

⁴⁷. Les caractères `\&` sont utilisés au lieu du `&` utilisé habituellement avec T_ikZ pour des raisons de compatibilité avec BEAMER

Il n'est pas nécessaire de mettre la `\tmatrix` dans une équation et les cellules sont par défaut des équations

Il est également possible de faire de même avec la hauteur des \dots en modifiant la longueur `\dotlift`. De même avec `\matmin` pour `minimum width` et `minimum height`, ou encore `\matsep` pour `row sep` et `column sep`.

10.3 La commande `\tmatrix`

`\tmatrix` est composé de deux arguments optionnels (les éléments à ajouter à la matrice TikZ et les éléments de mise en page de la matrice) ainsi que de trois arguments (le délimiteur d'ouverture, le contenu de la matrice et le délimiteur de fermeture).

Commande	Résultat
<code>\mtxvline{params}{n}</code>	Crée une ligne verticale après la colonne n (ou <code>left/right</code> pour les extrémités) avec les paramètres TikZ params
<code>\mtxhline{params}{n}</code>	Crée une ligne horizontale après la ligne n (ou <code>top/bottom</code> pour les extrémités) avec les paramètres TikZ params
<code>\mtxvpartial{params}{n}{a}{b}</code>	Crée une ligne verticale après la colonne n (ou <code>left/right</code> pour les extrémités), la ligne ayant pour extrémités la fin de la ligne a et b (ou <code>top/bottom</code>) avec les paramètres TikZ params
<code>\mtxhpartial{params}{n}{a}{b}</code>	Crée une ligne horizontale après la ligne n (ou <code>top/bottom</code> pour les extrémités), la ligne ayant pour extrémités la fin de la ligne a et b (ou <code>left/right</code>) avec les paramètres TikZ params
<code>\mtxbox{params}{x}{y}</code>	Crée une boîte autour de la case de coordonnées x et y (l'indexation commence à 1) avec les paramètres TikZ params

10.4 Exemples avec `\tmatrix`

$\det(M) = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$ est produit par `$\det{M}=\tmatrix|{a\&b\\c\&d}\\|`.

$I_{n,p,r} = \left(\begin{array}{c|c} I_r & 0_{r,p-r} \\ \hline 0_{n-r,r} & 0_{n-r,p-r} \end{array} \right)$ est produit par

```
$
I_{n,p,r}=\tmatrix
  [\mtxvline{line width = 0.05em}{1}\mtxhline{line width = 0.05em}{1}]
  [minimum height = 5ex, row sep = 1ex, minimum width = 5ex,
   column sep = 1ex,]
  ({I_r\&0_{r,p-r}\\0_{n-r,r}\&0_{n-r,p-r}\\})
$
```


$$\left[\begin{array}{cccc} A_1 & 0 & 0 & 0 \\ 0 & A_2 & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & A_n \end{array} \right] \text{ est produit par }$$

```

\tmatrix
[
  \mtxbox{red, dashed}{1}{1}
  \mtxbox{teal, dotted, ultra thick}{2}{2}
  \mtxbox{}{4}{4}
]
[
  minimum height = 5ex,
  minimum width = 5ex,
  row sep = 10pt,
  inner sep = 5pt,
  column sep = 10pt,
]
{{[]}} % Le crochet est entouré de deux paires d'accolades
{
  A_1\&0\&0\&0\\
  0\&A_2\&\ddots\&0\\
  0\&\ddots\&\ddots\&0\\
  0\&0\&0\&A_n\\
}
{\}

```

11 Polynomes.sty

Commande	Résultat
<code>\pol{K}{X}</code>	$\mathbb{K}[X]$
<code>\fr{K}{X}</code>	$\mathbb{K}(X)$
<code>\olddeg</code> ⁴⁸	deg
<code>\deg{P}</code>	$\deg(P)$
<code>\oldval</code>	val
<code>\val{P}</code>	$\text{val}(P)$
<code>\oldcar</code>	car
<code>\car{\mathbb{K}}</code>	$\text{car}(\mathbb{K})$

48. Correspond à la commande usuelle `\deg`

12 Probas.sty

Commande	Résultat
<code>\p{A}</code>	$\mathbb{P}(A)$
<code>\p[B]{A}</code>	$\mathbb{P}_B(A)$
<code>\oldOmega</code> ⁴⁹	Ω
<code>\Omega</code> ⁵⁰	Ω
<code>\sq</code> ⁵¹	$ $
<code>\bor</code> ⁵²	\mathcal{B}
<code>\esp{X}</code>	$\mathbb{E}(X)$
<code>\var{X}</code>	$\mathbb{V}(X)$
<code>\ect{X}</code>	$\sigma(X)$
<code>\oldcov</code>	cov
<code>\cov{X}{Y}</code>	$\text{cov}(X, Y)$
<code>\indep</code> ⁵³	\perp
<code>\unif{n}</code>	$\mathcal{U}(n)$
<code>\bin{p}</code>	$\mathcal{B}(p)$
<code>\bin[n]{p}</code>	$\mathcal{B}(n, p)$
<code>\geom{p}</code>	$\mathcal{G}(p)$
<code>\pasc{r}{p}</code>	$\mathcal{P}(r, p)$
<code>\nbin{r}{p}</code>	$\mathcal{J}(r, p)$
<code>\hypg{N}{n}{q}</code>	$\mathcal{H}(N, n, q)$
<code>\poiss{\lambda}</code>	$\mathcal{P}(\lambda)$

49. Correspond à la commande usuelle `\Omega`

50. Correspond à la commande usuelle `\varOmega`

51. Correspond à la commande usuelle `\middle|`

Doit être utilisé entre `\left` et `\right`, ou dans la commande `\p` : $\mathbb{P}\left(A \mid \bigcap_{k=1}^n (B_k)\right)$

52. Correspond à la commande usuelle `\mathcal{B}`

53. Ce symbole est obtenu avec la commande `\perp\!\!\!\perp`

13 Sffont.sty

Ce package définit une nouvelle police `cmssp` qui correspond à `cmss` en 10pt. Pour l'utiliser, il faut utiliser `\fontfamily{cmssp}\fontsize{Xpt}{\baselineskip}\selectfont`.

Comparaison entre `cmssp` et `cmss` :

cmssp	cmss
Exemple avec une police de taille 21pt en gras.	Exemple avec une police de taille 21pt en gras.
<i>Exemple avec une police de taille 17pt en italique.</i>	<i>Exemple avec une police de taille 17pt en italique.</i>
<i>Exemple avec une police de taille 12pt en gras italique.</i>	<i>Exemple avec une police de taille 12pt en gras italique.</i>

14 Structures.sty

Commande	Résultat
<code>\oldhom</code>	Hom
<code>\hom{E}</code>	$\text{Hom}(E)$
<code>\oldaut</code>	Aut
<code>\aut{E}</code>	$\text{Aut}(E)$
<code>\oldker</code> ⁵⁴	ker
<code>\ker{f}</code>	$\text{ker}(f)$
<code>\oldim</code>	im
<code>\im{f}</code>	$\text{im}(f)$
<code>\la</code> ⁵⁵	\langle
<code>\ra</code> ⁵⁶	\rangle
<code>\oldord</code>	ord
<code>\ord{x}</code>	$\text{ord}(x)$

54. Correspond à la commande usuelle `\ker`

55. Correspond à la commande usuelle `\left\langle`

56. Correspond à la commande usuelle `\right\rangle`

15 Tables.sty

Ce package sert à mettre en forme des tables an latex grâce à TikZ.

Pour insérer une table, il faut appeler `\setrowcol[width][height]{ncols}{nrows}` avec le nombre de colonnes et de lignes de la table, puis rentrer la table TikZ, les arguments optionnels étant la largeur de la table et sa hauteur.

Une table a une largeur par défaut de 10cm et une hauteur de 6,5cm (est est réinitialisée à chaque appel de `\setrowcol`).

Il est possible d'utiliser `[ampersand replacement=\&]` puis `\&` pour la matrice lorsque `&` est déjà défini par l'environnement (comme BEAMER).

Il est possible de récupérer la valeur de la largeur et de la hauteur avec `\tblw` et `\tblh`.

Par exemple, la table

	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
sin	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1
cos	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0
tan	0	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$	—
cot	—	$\sqrt{3}$	1	$\frac{1}{\sqrt{3}}$	0

est produite avec le code suivant

```
\LARGE
\setcolrow[15cm]{6}{5}
\begin{tikzpicture}
  \matrix[table]{
    &0&\oldfrac{\pi}{6}&\oldfrac{\pi}{4}&\oldfrac{\pi}{3}&\oldfrac{\pi}{2}\\
    sin&0&\oldfrac{1}{2}&\oldfrac{\sqrt{2}}{2}&\oldfrac{\sqrt{3}}{2}&1\\
    cos&1&\oldfrac{\sqrt{3}}{2}&\oldfrac{\sqrt{2}}{2}&\oldfrac{1}{2}&0\\
    tan&0&\oldfrac{1}{\sqrt{3}}&1&\sqrt{3}&--\\
    cot&--&\sqrt{3}&1&\oldfrac{1}{\sqrt{3}}&0
  };
  \draw [line width=0.5mm] (-\tblw/3,-\tblh/2) -- (-\tblw/3,\tblh/2);
  \draw [line width=0.5mm] (-\tblw/2,3*\tblh/10) --
    (\tblw/2,3*\tblh/10);
  \draw [line width=0.5mm] (-\tblw/2,-\tblh/2) rectangle
    (\tblw/2,\tblh/2);
\end{tikzpicture}
```

16 Tools.sty

Ce fichier fournit des commandes latex utiles pour créer des macros.

Commande	Résultat
<code>\comparestring{a}{b}{if}{else}</code>	Compare les chaînes de caractères a et b ; puis exécute le if si a est égal à b et le else sinon

17 Trigo.sty

Commande	Résultat
<code>\oldcos</code> ⁵⁷	\cos
<code>\cos{x}</code>	$\cos(x)$
<code>\cos[n]{x}</code>	$\cos^n(x)$
<code>\oldsin</code> ⁵⁸	\sin
<code>\sin{x}</code>	$\sin(x)$
<code>\sin[n]{x}</code>	$\sin^n(x)$
<code>\oldtan</code> ⁵⁹	\tan
<code>\tan{x}</code>	$\tan(x)$
<code>\tan[n]{x}</code>	$\tan^n(x)$
<code>\oldcot</code> ⁶⁰	\cot
<code>\cot{x}</code>	$\cot(x)$
<code>\cot[n]{x}</code>	$\cot^n(x)$
<code>\acos{x}</code>	$\arccos(x)$
<code>\acos[n]{x}</code>	$\arccos^n(x)$
<code>\asin{x}</code>	$\arcsin(x)$
<code>\asin[n]{x}</code>	$\arcsin^n(x)$
<code>\atan{x}</code>	$\arctan(x)$
<code>\atan[n]{x}</code>	$\arctan^n(x)$
<code>\oldch</code>	ch
<code>\ch{x}</code>	$\operatorname{ch}(x)$
<code>\ch[n]{x}</code>	$\operatorname{ch}^n(x)$
<code>\oldsh</code>	sh
<code>\sh{x}</code>	$\operatorname{sh}(x)$
<code>\sh[n]{x}</code>	$\operatorname{sh}^n(x)$
<code>\oldth</code>	th
<code>\th{x}</code>	$\operatorname{th}(x)$
<code>\th[n]{x}</code>	$\operatorname{th}^n(x)$
<code>\oldach</code>	argch
<code>\ach{x}</code>	$\operatorname{argch}(x)$
<code>\ach[n]{x}</code>	$\operatorname{argch}^n(x)$
<code>\oldash</code>	argsh
<code>\ash{x}</code>	$\operatorname{argsh}(x)$
<code>\ash[n]{x}</code>	$\operatorname{argsh}^n(x)$
<code>\oldath</code>	argth
<code>\ath{x}</code>	$\operatorname{argth}(x)$
<code>\ath[n]{x}</code>	$\operatorname{argth}^n(x)$

57. Correspond à la commande usuelle `\cos`

58. Correspond à la commande usuelle `\sin`

59. Correspond à la commande usuelle `\tan`

60. Correspond à la commande usuelle `\cot`

18 Usuelles.sty

Commande	Résultat
<code>\oldmin</code> ⁶¹	\min
<code>\min{\llb0,n\rrb}</code>	$\min(\llbracket 0, n \rrbracket)$
<code>\min[\mathbb{N}^*]{\llb0,n\rrb}</code>	$\min_{\mathbb{N}^*}(\llbracket 0, n \rrbracket)$
<code>\oldmax</code> ⁶²	\max
<code>\max{\llb0,n\rrb}</code>	$\max(\llbracket 0, n \rrbracket)$
<code>\max[\mathbb{Z}_-]{\llb0,n\rrb}</code>	$\max_{\mathbb{Z}_-}(\llbracket 0, n \rrbracket)$
<code>\oldlim</code> ⁶³	\lim
<code>\lim{u_n}</code>	$\lim(u_n)$
<code>\lim[x\to+\infty]{f(x)}</code>	$\lim_{x \rightarrow +\infty}(f(x))$
<code>\limi{u_n}</code>	$\liminf(u_n)$
<code>\limi[x\to+\infty]{f(x)}</code>	$\liminf_{x \rightarrow +\infty}(f(x))$
<code>\lims{u_n}</code>	$\limsup(u_n)$
<code>\lims[x\to+\infty]{f(x)}</code>	$\limsup_{x \rightarrow +\infty}(f(x))$
<code>\oldexp</code> ⁶⁴	\exp
<code>\exp{x}</code>	$\exp(x)$
<code>\exp[n]{x}</code>	$\exp^n(x)$
<code>\oldln</code> ⁶⁵	\ln
<code>\ln{x}</code>	$\ln(x)$
<code>\ln[n]{x}</code>	$\ln^n(x)$
<code>\oldinf</code> ⁶⁶	\inf
<code>\inf{\varnothing}</code>	$\inf(\emptyset)$
<code>\inf[\bar{\mathbb{R}}]{\{u_n\}}</code>	$\inf_{\bar{\mathbb{R}}}(\{u_n\})$
<code>\oldsup</code> ⁶⁷	\sup
<code>\sup{\varnothing}</code>	$\sup(\emptyset)$
<code>\sup[\bar{\mathbb{R}}]{\{u_n\}}</code>	$\sup_{\bar{\mathbb{R}}}(\{u_n\})$

61. Correspond à la commande usuelle `\min`

62. Correspond à la commande usuelle `\max`

63. Correspond à la commande usuelle `\lim`

64. Correspond à la commande usuelle `\exp`

65. Correspond à la commande usuelle `\ln`

66. Correspond à la commande usuelle `\inf`

67. Correspond à la commande usuelle `\sup`