

L'extension `frenchmath`*

Antoine Missier
`antoine.missier@ac-toulouse.fr`

11 novembre 2022

1 Introduction

Cette extension, inspirée de `mafr` de Christian Obrecht [10], permet le respect des règles typographiques mathématiques françaises, en particulier la possibilité d'obtenir automatiquement les majuscules en romain (lettres droites) plutôt qu'en italique (voir [1] et [2]) ainsi que des espacements corrects pour les virgules et point-virgules¹. Depuis la version 2.0, une option permet également de composer les minuscules grecques du mode mathématique en forme droite plutôt qu'en italique.

D'autres solutions pour composer les majuscules mathématiques en romain sont proposées dans les extensions `fourier` de Michel Bovani [12] (avec la famille des fontes Adobe Utopia) ou encore `mathdesign` de Paul Pichaureau [13] (avec les fontes Adobe Utopia, URW Garamond ou Bitstream Charter). Mais `frenchmath` fournit une solution générique s'adaptant à n'importe quelle fonte.

D'autres préconisations, telles que composer en lettre droite et non en italique le symbole différentiel, les constantes mathématiques *i* et *e* [2], sont des règles internationales [5] [6] [7]. Elles ne sont donc pas implémentées dans `frenchmath`².

L'extension fournit en outre diverses macros francisées. Quelques différences sont à signaler avec `mafr` :

- nous avons choisi de ne pas substituer les symboles français aux symboles anglo-saxons avec le même nom de commande ;
- les macros présentées dans la section 2 correspondant à des macros de `mafr` sont signalées par un astérisque en fin d'item, les autres sont nouvelles ;
- enfin quelques commandes de `mafr` ne sont pas spécifiques aux mathématiques françaises et ne sont donc pas abordées ici : c'est le cas de `\vect`³, des ensembles de nombres `\R`, `\N`, ... (pour `\R`, `\N`, ...) ainsi que celles relatives à la réalisation de feuilles d'exercices.

Mentionnons aussi l'extension `tdsfrmath` [11] de Yvon Henel qui fournit beaucoup de commandes francisées.

*Ce document correspond à `frenchmath` v2.1, dernière modification le 11/11/2022.

1. Merci à Jean-François Burnol pour quelques améliorations proposées au code.

2. Nous proposons pour cela l'extension `mismath` [23] qui fournit diverses macros pour les mathématiques internationales.

3. Pour de jolis vecteurs on dispose de l'extension `esvect` d'Eddie Soudrais.

2 Utilisation

2.1 Majuscules mathématiques

En France, les lettres majuscules du mode mathématique doivent toujours être composées en romain (A, B, C, ...) et non en italique ([1] p.107, voir aussi [2]). En utilisant X_YLaTeX ou LuaLaTeX avec des fontes mathématiques OpenType, cette convention est assez commode à mettre en œuvre, mais, avec LaTeX ou pdfLaTeX, assez peu d’auteurs la respectent et les extensions précitées ne fonctionnent qu’avec des fontes précises. Par défaut **frenchmath** compose automatiquement toutes les majuscules mathématiques en romain, quelle que soit la fonte utilisée. Par exemple `\[P(X)=\sum_{i=0}^n a_i X^i \]` donne avec **frenchmath**

$$P(X) = \sum_{i=0}^n a_i X^i.$$

capsit L’extension **frenchmath** possède l’option **capsit**, qui permet de conserver la composition des majuscules du mode mathématique en italique (dans la famille de fonte par défaut). Que l’option soit activée ou pas, il est toujours possible de changer l’aspect d’une lettre particulière, avec les macros LaTeX `\mathrm` et `\mathit`.

2.2 Lettres grecques

La norme concernant l’usage des lettres grecques en italique ou en forme droite pour les mathématiques françaises ne semble pas très claire et les auteurs divergent sur ce point. Plusieurs recommandent l’usage des lettres grecques minuscules en forme droite [12] [13] [18], mais d’autres préconisent l’italique, comme pour toutes les variables mathématiques [3]. Le lexique des règles typographiques en usage à l’Imprimerie Nationale [1] les compose en forme droite et relativement grasses (p.108) sans préciser s’il s’agit vraiment d’une règle s’appliquant aux variables, au même titre que celles énoncées pour l’alphabet latin.

Pour les physiciens (et chimistes) l’affaire est plus claire puisque les quantités doivent toujours être écrites en italique et les unités ou les constantes en romain (forme droite), conformément à la norme ISO [5] [6] [7]. Ainsi la constante $\pi \approx 3,14$ ne s’écrit pas de la même manière qu’une variable π . Dans la section « How to get upright small Greek letters », la documentation de **isomath** de Günter Milde [17] fait le point sur les différentes méthodes pour obtenir les lettres grecques en forme droite. Par exemple les extensions **mathdesign** [13], **fourier** [12] ou **kpfonts** [14] disposent d’options permettant l’écriture automatique des lettres grecques minuscules en forme droite aussi bien que des majuscules en italique. Citons également **newpx**, **newtx** et **libertinustlmath** de Michael Sharpe, **pxgreek**, **txgreek** et **libgreek** de Jean-François Burnol, qui donnent de beaux résultats pour une utilisation avec respectivement les fontes Palatino, Times et Libertine ⁴.

4. Attention ces trois dernières extensions doivent être chargées *après* **amsmath** (et donc aussi **mismath**). En effet, elles redéfinissent les commandes `\iint`, `\iiint`, `\iiiint`, `\idotsint`.

upgreek Avec la même philosophie, **frenchmath** fournit l’option **upgreek** basée sur l’extension **upgreek** de Walter Schmidt [16] qui donne accès aux lettres grecques minuscules en forme droite : `\upalpha`, `\upbeta`, ... L’extension **upgreek** sera chargée avec l’option par défaut, **Euler**. Si l’on veut, par contre, utiliser l’extension **upgreek** avec l’une de deux autres options disponibles, **Symbol** ou **Symbolsmallscale** (utilisant la fonte Adobe Symbol), il faut charger l’extension **upgreek** avec l’option souhaitée *avant* **frenchmath**⁵. L’option **upgreek** de **frenchmath** redéfinit ensuite les commandes `\alpha`, `\beta`, ... pour produire automatiquement les lettres en forme droite α , β , ..., π , ..., les formes italiques, α , β , ..., π , ..., restant cependant disponibles avec les commandes `\italpha`, `\itbeta`, ..., `\itpi`, ...

Uppgreek Avec L^AT_EX, les lettres grecques majuscules sont automatiquement composées en forme droite, néanmoins il peut être utile de disposer de toutes les lettres grecques dans le même style; ainsi l’extension **upgreek** propose aussi `\Uppgamma`, ..., `\Uppomega` : Γ , ..., Ω . L’option **Uppgreek** de **frenchmath** redéfinit les majuscules `\Gamma`, ..., `\Omega` pour correspondre à ces variantes. Par contre l’on n’a alors plus accès aux caractères d’origine : Γ , ..., Ω ⁶.

L’option **Uppgreek** reprend aussi les minuscules grecques de l’option **upgreek**, qu’il est donc inutile d’invoquer simultanément.

Signalons enfin l’extension **textalpha** de Günter Milde [15] qui donne accès aux lettres en forme droite α , β , ..., π , ..., mais en mode texte avec `\textalpha`, `\textbeta`, etc. Ces glyphes se marient particulièrement bien avec la fonte Latin Modern utilisée ici. Par contre le theta produit n’est pas vraiment celui attendu, ϑ , à moins d’utiliser la fonte **libertinus** ou de compiler par X_YL^AT_EX avec l’extension **fontspec** (et seulement avec les versions de **textalpha** postérieures à 2020); d’autre part le rendu de `\textpi` est parfois curieusement déformé.

Mentionnons ce commentaire de Walter Schmidt [16] que le μ utilisé pour le préfixe des unités physiques, μ , doit se composer avec `\textmu`⁷, disponible en mode texte dans beaucoup de fontes (ou avec **textcomp**); il diffère du μ de **upmu**.

2.3 Virgules, point-virgule et crochets

Dans le mode mathématique de L^AT_EX, la virgule est toujours, par défaut, un symbole de ponctuation et sera donc suivi d’une espace. Ceci est légitime dans un intervalle : `(a,b)` donne $[a, b]$, mais pas pour un nombre en français : `$12,5$` donne 12,5 au lieu de 12,5. L’extension **babel**, avec l’option **french** [19], fournit deux bascules : `\DecimalMathComma` et `\StandardMathComma`, qui permettent d’adapter le comportement de la virgule du mode mathématique. Deux autres extensions bien commodes permettent néanmoins de se passer de ces bascules⁸. En mode mathématique :

5. L’option **Symbol** de **upgreek** se marie mieux avec une fonte comme Times par exemple.

6. L’extension **textalpha** fournit également les lettres grecques majuscules, mais elles semblent vraiment identiques à celles par défaut de L^AT_EX. Il n’y a donc pas lieu d’envisager une option particulière comme **Uppgreek**.

7. L’extension **textalpha** fournit à la place `\textmicro` (depuis 2020) car elle redéfinit `\textmu`.

8. Dans ce cas il ne faut pas utiliser les bascules, au risque de rendre ces extensions inopérantes.

- avec `icomma` (intelligent comma) de Walter Schmidt [20], la virgule se comporte comme un caractère de ponctuation si elle est suivie d’une espace, sinon c’est un caractère ordinaire,
- avec `nccomma` de Alexander I. Rozhenko [21], la virgule se comporte comme un caractère ordinaire si elle est suivie d’un chiffre (sans espace), sinon c’est un caractère de ponctuation.

Cette deuxième approche paraît meilleure, néanmoins `nccomma` ne fonctionne pas avec l’option `autolanguage`⁹ de l’extension `numprint`. Par contre `icomma` fonctionne et était utilisé jusqu’à la version 1.5 de `frenchmath`. Un code proposé par Jean-François Burnol permet, depuis la `frenchmath` 1.6, d’utiliser `nccomma` qui sera donc chargée par `frenchmath` à la place de `icomma`¹⁰.

Lorsque l’on utilise l’extension `pstricks-add` de `PSTricks` pour tracer des axes de coordonnées, l’appel `\psset{comma=true}` permet d’avoir les graduations avec une virgule au lieu du point décimal. Ce réglage est effectué par défaut ici.

;
Le symbole « ; » a été redéfini pour le mode mathématique car l’espace précédant le point-virgule est incorrecte en français `$x \in [0,25 ; 3,75]$` donne $x \in [0,25;3,75]$ sans `frenchmath` et $x \in [0,25 ; 3,75]$ avec `frenchmath`; le comportement de « ; » devient identique à celui de « : »

[]
Les crochets ouvrants et fermants [et] ont été redéfinis pour le mode mathématique car l’espace les précédant est souvent incorrecte¹¹ :

$$x \in]0, \pi[\cup]2\pi, 3\pi[\rightarrow \begin{cases} x \in]0, \pi[\cup]2\pi, 3\pi[& \text{sans frenchmath} \\ x \in]0, \pi[\cup]2\pi, 3\pi[& \text{avec frenchmath} \end{cases}$$

Dans notre code, les symboles [et] ne sont plus définis comme des délimiteurs. De ce fait, une coupure de ligne peut se produire entre les deux, mais il est toujours possible de transformer alors ces crochets en délimiteurs avec `\left` et `\right`. Un autre problème se pose lorsque la borne inférieure de l’intervalle contient un symbole d’opération car les espaces autour de l’opération seront alors trop grandes : `$[-1, 1]$` donne $[-1, 1]$ au lieu de $[-1, 1]$ ¹². Pour rétablir les espaces correctes, il faut entourer le $-$ ou le -1 par des accolades ou utiliser `\left` et `\right`.

2.4 Quelques macros et alias utiles

`\curs` Les lettres cursives (*A, B, C, D, ...*), provenant de l’extension `mathrsfs` chargée par `frenchmath`, sont composées avec `\curs` et sont différentes de celles obtenues

9. L’option `autolanguage` de `numprint` utilisée conjointement avec l’option `french` de `babel` garantit un espacement correct entre les groupes de trois chiffres dans les grands nombres, qui doit être une espace insécable et non dilatable [1], légèrement plus grande que l’espace que l’on obtient sans cette option.

10. Mentionnons aussi l’article *Intelligent commas* de Claudio Beccari [22] qui propose une solution simplifiée par rapport à `nccomma` mais dont l’avantage est discutable.

11. L’extension `interval` fournit une autre solution basée sur la macro `\interval`; citons aussi `mathtools` et sa commande `\DeclarePairedDelimiter`.

12. Problème signalé par J.F. Burnol, ce qui n’enlève pas l’intérêt de cette redéfinition.

avec `\mathcal`¹³ ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \dots$). La commande `\curs` permet aussi de composer ces lettres en mode texte, sans les délimiteurs du mode mathématique.*¹⁴

`\infeg` Les relations \leq et \geq s’obtiennent avec les commandes `\infeg` et `\supeg` et
`\supeg` diffèrent des versions anglaises de `\leq` (\leq) et `\geq` (\geq). Ce sont des alias de `\leqslant` et `\geqslant` de l’extension `amssymb`, chargée par `frenchmath`.*

`\vide` Le symbole de l’ensemble vide \emptyset s’obtient avec `\vide` (alias de la commande `\varnothing` de l’extension `amssymb`) ; il diffère de celui obtenu avec `\emptyset` (particulièrement laid dans la fonte par défaut Computer Modern : \emptyset).*

`\paral` La commande `\paral` fournit la *relation*¹⁵ du parallélisme : \parallel , plutôt
`\paral*` que sa version anglaise `\parallel` : \parallel .* Avec certaines fontes mathématiques (`libertinustlmath`, `mathastext+times`, etc.), les barres obliques peuvent se trouver trop serrées, on peut utiliser alors `\paral*` qui les écarte un peu plus.

`\ssi` La commande `\ssi` produit « si, et seulement si, ».

`\cmod` Le modulo se compose normalement entre parenthèses, avec `\pmod`, mais on rencontre aussi, en français, le modulo entre crochets, ce que permet la commande `\cmod` en respectant le bon espacement propre au modulo : $5 \equiv 53 \quad [12]$.

2.5 Identifiants de « fonctions » classiques

`\pgcd` En arithmétique, nous avons les classiques `\pgcd` et `\ppcm`, qui diffèrent de
`\ppcm` leur version anglo-saxonne `\gcd` et `\lcm`¹⁶.

`\card` Pour le cardinal d’un ensemble, nous proposons `\card`, cité dans [1] et [3], ou
`\Card` `\Card`, qui est aussi d’usage courant (cf. Wikipedia).

`\Ker` L^AT_EX fournit les macros `\ker` et `\hom`, alors que l’usage français est souvent
`\Hom` de commencer ces noms par une majuscule pour obtenir Ker ¹⁷ et Hom .

`\rg` Le rang d’une application linéaire ou d’une matrice (rg) s’obtient avec la com-
`\Vect` mande `\rg` et l’espace vectoriel engendré par une famille de vecteurs avec `\Vect`.

`\ch` En principe, les fonctions hyperboliques s’écrivent en français avec les macros
`\sh` L^AT_EX standard `\cosh`, `\sinh`, `\tanh`. Néanmoins les écritures $\text{ch } x$, $\text{sh } x$ et $\text{th } x$,
`\th` qui sont la norme avec les langues d’Europe de l’Est [8], sont aussi utilisées en France [1]. On les obtient avec les commandes `\ch`, `\sh` et `\th`¹⁸.

13. L’extension `calrsfs` fournit les mêmes cursives mais en redéfinissant la commande `\mathcal`.

14. Comme précisé dans l’introduction, l’astérisque en fin d’item signale une fonctionnalité similaire dans `mafr`.

15. Pour noter que deux objets sont perpendiculaires, on utilise `\perp`, défini comme une *relation* mathématique plutôt que `\bot` défini comme un *symbole* (les espacements diffèrent).

16. Cette dernière n’est pas implémentée en standard dans L^AT_EX (mais dans `mismath`).

17. La commande `\Im` existe déjà pour la partie imaginaire des nombres complexes et produit \Im ; elle est redéfinie en `\Im` par l’extension `mismath` et peut aussi être utilisée pour l’image.

18. La commande `\th` existe déjà, pour le mode texte uniquement, et produit p ; elle a été redéfinie, uniquement pour le mode mathématique, à la place de l’ancienne commande `\tgh`, désormais obsolète.

2.6 Bases et repères

`\Oij` Les repères classiques du plan ou de l'espace seront composés avec des hauteurs de flèches homogénéisées : `\Oij` compose (O, \vec{i}, \vec{j}) , `\Oijk` compose $(O, \vec{i}, \vec{j}, \vec{k})$ et `\Ouv` compose (O, \vec{u}, \vec{v}) (utilisé dans le plan complexe). On peut écrire ces commandes en mode texte, sans les délimiteurs du mode mathématique.
`\Oij*` Les versions étoilées utilisent le point-virgule et non la virgule comme séparateur après le point O, comme mentionné dans [1]. On obtient $(O ; \vec{i}, \vec{j})$, $(O ; \vec{i}, \vec{j}, \vec{k})$, `\Oijk*` $(O ; \vec{i}, \vec{j}, \vec{k})$, `\Ouv*` $(O ; \vec{u}, \vec{v})$.
`\ij` Enfin les macros `\ij`¹⁹ et `\ijk` composent les bases du plan et de l'espace, `\ijk` (\vec{i}, \vec{j}) et $(\vec{i}, \vec{j}, \vec{k})$, en homogénéisant la hauteur des flèches.

Signalons que les macros de ce paragraphe peuvent ne pas fonctionner avec certaines fontes mathématiques qui ne connaissent pas `\jmath`, par exemple avec l'extension `mathptmx` (basée sur la fonte de texte Times).

3 Le code

```

1 \RequirePackage{ifthen}
2 \newboolean{capsit}
3 \DeclareOption{capsit}{\setboolean{capsit}{true}}
4 \newboolean{upgreek}
5 \DeclareOption{upgreek}{\setboolean{upgreek}{true}}
6 \newboolean{Upgreek}
7 \DeclareOption{Upgreek}{\setboolean{Upgreek}{true}}
8 \setboolean{upgreek}{true}}
9 \ProcessOptions \relax
10
11 \AtBeginDocument{
12   \@ifpackageloaded{mathdesign}{
13     \PackageWarningNoLine{frenchmath}{Package mathdesign is loaded,
14       \MessageBreak
15       I don't load mathrsfs and amssymb packages}
16   }{
17     \RequirePackage{mathrsfs} % fournit les majuscules cursives
18     \RequirePackage{amssymb} % \leqslant, \geqslant, \varnothing
19   }
20 }
21 \RequirePackage{amsopn} % fournit \DeclareMathOperator
22 \ifthenelse{\boolean{upgreek}}{
23   \@ifpackageloaded{upgreek}{\RequirePackage{Euler}{upgreek}}
24 }{}
25 \RequirePackage{xspace} % utile pour les commandes \curs, \ssi, \Oij
26 % \RequirePackage{icomma} % intelligent comma
27 \RequirePackage{nccomma} % depuis frenchmath 1.6
28 \AtBeginDocument{\mathcode'\,="8000\relax
29   \@ifpackageloaded{babel}{

```

19. Notons que la macro `\ij` existait déjà (ligature entre i et j pour le hollandais) et a été redéfinie.

```

30      \addto\extrasfrench{\mathcode'\,="8000\relax}
31    }{}
32 }

```

Merci à J.F. Burnol pour ce patch permettant d'utiliser `ncccomma` à la place de `icomma`, ainsi que pour l'amélioration du code ci-après, redéfinissant les majuscules mathématiques.

Sauf si l'option `capsit` est activée, on redéfinit toutes les lettres majuscules du mode mathématique ; `\AtBeginDocument` est nécessaire pour que ces définitions soient prises en compte avec la classe `beamer` par exemple.

```

33 \ifthenelse{\boolean{capsit}}{}{
34   \AtBeginDocument{
35     \DeclareMathSymbol{A}\mathalpha{operators}{'A}
36     \DeclareMathSymbol{B}\mathalpha{operators}{'B}
37     \DeclareMathSymbol{C}\mathalpha{operators}{'C}
38     \DeclareMathSymbol{D}\mathalpha{operators}{'D}
39     \DeclareMathSymbol{E}\mathalpha{operators}{'E}
40     \DeclareMathSymbol{F}\mathalpha{operators}{'F}
41     \DeclareMathSymbol{G}\mathalpha{operators}{'G}
42     \DeclareMathSymbol{H}\mathalpha{operators}{'H}
43     \DeclareMathSymbol{I}\mathalpha{operators}{'I}
44     \DeclareMathSymbol{J}\mathalpha{operators}{'J}
45     \DeclareMathSymbol{K}\mathalpha{operators}{'K}
46     \DeclareMathSymbol{L}\mathalpha{operators}{'L}
47     \DeclareMathSymbol{M}\mathalpha{operators}{'M}
48     \DeclareMathSymbol{N}\mathalpha{operators}{'N}
49     \DeclareMathSymbol{O}\mathalpha{operators}{'O}
50     \DeclareMathSymbol{P}\mathalpha{operators}{'P}
51     \DeclareMathSymbol{Q}\mathalpha{operators}{'Q}
52     \DeclareMathSymbol{R}\mathalpha{operators}{'R}
53     \DeclareMathSymbol{S}\mathalpha{operators}{'S}
54     \DeclareMathSymbol{T}\mathalpha{operators}{'T}
55     \DeclareMathSymbol{U}\mathalpha{operators}{'U}
56     \DeclareMathSymbol{V}\mathalpha{operators}{'V}
57     \DeclareMathSymbol{W}\mathalpha{operators}{'W}
58     \DeclareMathSymbol{X}\mathalpha{operators}{'X}
59     \DeclareMathSymbol{Y}\mathalpha{operators}{'Y}
60     \DeclareMathSymbol{Z}\mathalpha{operators}{'Z}
61   }
62 }

```

Avec l'option `upgreek`, on charge l'extension `upgreek` (sauf si elle est déjà chargée, ce qui évite les incompatibilités d'option) et on redéfinit les commandes `\alpha`, `\beta`, ... `Uppgreek` transforme en outre les majuscules grecques pour garder le même style.

```

63
64 \ifthenelse{\boolean{upgreek}}{
65   \@ifundefined{italpha}{\let\italpha\alpha}{
66     \PackageWarningNoLine{frenchmath}{Italicized Greek
67       \bslash italpha already exist \MessageBreak

```

```

68         and will not be redefined, \MessageBreak
69         no more warning for the other letters, except pi}
70     }
71     \@ifundefined{itbeta}{\let\itbeta\beta}{}
72     \@ifundefined{itgamma}{\let\itgamma\gamma}{}
73     \@ifundefined{itdelta}{\let\itdelta\delta}{}
74     \@ifundefined{itepsilon}{\let\itepsilon\epsilon}{}
75     \@ifundefined{itzeta}{\let\itzeta\zeta}{}
76     \@ifundefined{iteta}{\let\iteta\eta}{}
77     \@ifundefined{ittheta}{\let\ittheta\theta}{}
78     \@ifundefined{itiota}{\let\itiota\iota}{}
79     \@ifundefined{itkappa}{\let\itkappa\kappa}{}
80     \@ifundefined{itlambda}{\let\itlambda\lambda}{}
81     \@ifundefined{itmu}{\let\itmu\mu}{}
82     \@ifundefined{itnu}{\let\itnu\nu}{}
83     \@ifundefined{itxi}{\let\itxi\xi}{}
84     \@ifundefined{itpi}{\let\itpi\pi}{
85         \PackageWarningNoLine{frenchmath}{Italicized Greek \bslash itpi
86             already exist \MessageBreak
87             and will not be redefined}
88     }
89     \@ifundefined{itrho}{\let\itrho\rho}{}
90     \@ifundefined{itsigma}{\let\itsigma\sigma}{}
91     \@ifundefined{ittau}{\let\ittau\tau}{}
92     \@ifundefined{itupsilon}{\let\itupsilon\upsilon}{}
93     \@ifundefined{itphi}{\let\itphi\phi}{}
94     \@ifundefined{itchi}{\let\itchi\chi}{}
95     \@ifundefined{itpsi}{\let\itpsi\psi}{}
96     \@ifundefined{itomega}{\let\itomega\omega}{}
97     \@ifundefined{itvarepsilon}{\let\itvarepsilon\varepsilon}{}
98     \@ifundefined{itvartheta}{\let\itvartheta\vartheta}{}
99     \@ifundefined{itvarpi}{\let\itvarpi\varpi}{}
100    \@ifundefined{itvarsigma}{\let\itvarsigma\varsigma}{}
101    \@ifundefined{itvarphi}{\let\itvarphi\varphi}{}
102 }{}
103
104 \ifthenelse{\boolean{upgreek}}{
105     \renewcommand\alpha{\upalpha}
106     \renewcommand\beta{\upbeta}
107     \renewcommand\gamma{\upgamma}
108     \renewcommand\delta{\updelta}
109     \renewcommand\epsilon{\upepsilon}
110     \renewcommand\zeta{\upzeta}
111     \renewcommand\eta{\upeta}
112     \renewcommand\theta{\uptheta}
113     \renewcommand\iota{\upiota}
114     \renewcommand\kappa{\upkappa}
115     \renewcommand\lambda{\uplambda}
116     \renewcommand\mu{\upmu}
117     \renewcommand\nu{\upnu}

```



```

118 \renewcommand\xi{\upxi}
119 \renewcommand\pi{\uppi}
120 \renewcommand\rho{\uprho}
121 \renewcommand\sigma{\upsigma}
122 \renewcommand\tau{\uptau}
123 \renewcommand\upsilon{\upupsilon}
124 \renewcommand\phi{\upphi}
125 \renewcommand\chi{\upchi}
126 \renewcommand\psi{\uppsi}
127 \renewcommand\omega{\upomega}
128 \renewcommand\varepsilon{\upvarepsilon}
129 \renewcommand\vartheta{\upvartheta}
130 \renewcommand\varpi{\upvarpi}
131 \renewcommand\varrho{\upvarrho}
132 \renewcommand\varsigma{\upvarsigma}
133 \renewcommand\varphi{\upvarphi}
134 }{}
135
136 \ifthenelse{\boolean{Upgreek}}{
137 \renewcommand\Gamma{\Upgamma}
138 \renewcommand\Delta{\Updelta}
139 \renewcommand\Theta{\Uptheta}
140 \renewcommand\Lambda{\Uplambda}
141 \renewcommand\Xi{\Upxi}
142 \renewcommand\Pi{\Upipi}
143 \renewcommand\Sigma{\Upsigma}
144 \renewcommand\Upsilon{\Upupsilon}
145 \renewcommand\Phi{\Upphi}
146 \renewcommand\Psi{\Uppsi}
147 \renewcommand\Omega{\Upomega}
148 }{}
149
150 \AtBeginDocument{\ifpackageloaded{pstricks-add}{\psset{comma=true}}{}}
151 \DeclareMathSymbol{;}\mathbin{operators}{'73} % \mathpunct à l'origine
152 \DeclareMathSymbol{[]}\mathord{operators}{093} % \mathclose à l'origine
153 \DeclareMathSymbol{[]}\mathord{operators}{091} % \mathopen à l'origine
154
155 \newcommand*\curs[1]{\ensuremath{\mathscr{#1}}\xspace}
156 \newcommand\infeg{\leqslant}
157 \newcommand\supeg{\geqslant}
158 \newcommand\vide{\varnothing}
159 \newcommand\@@paral{\mathrel{/!\!/}}
160 \newcommand\@paral{\mathrel{/!\!/}} % version étoilée avec moins d'espace
161 \newcommand\paral{\@ifstar{\@paral}{\@@paral}}
162 \newcommand\ssi{si, et seulement si,\xspace}
163 \newcommand*\cmod[1]{\quad\#1}
164
165 \DeclareMathOperator{pgcd}{pgcd}
166 \DeclareMathOperator{ppcm}{ppcm}
167 \DeclareMathOperator{card}{card}

```

```

168 \DeclareMathOperator{\Card}{Card}
169 \DeclareMathOperator{\Ker}{Ker}
170 \DeclareMathOperator{\Hom}{Hom}
171 \DeclareMathOperator{\rg}{rg}
172 \DeclareMathOperator{\Vect}{\Vect}
173 \DeclareMathOperator{\ch}{ch}
174 \DeclareMathOperator{\sh}{sh}
175 \AtBeginDocument{\let\oldth\th %\th existe déjà (mode texte uniquement)
176 \renewcommand\th{\TextOrMath{\oldth}{\operatorname{th}}}}
177 {\catcode'\|=z@ \catcode'\=12 \gdef\bslash{}} % la commande \bslash
178 \PackageWarningNoLine{frenchmath}{
179 Command \bslash tgh in now obsolete, \MessageBreak
180 use \bslash th instead}

La commande \bslash provient du package doc.sty de Frank Mittelbach. Elle est
utilisée à la place de \textbackslash (qui ne fonctionne pas ici).

181
182 \newcommand\@Oij{%
183 \ensuremath{\left(0, \vec{\imath}, \vec{\jmath}\right)\xspace}
184 \newcommand\@Oij{%
185 \ensuremath{\left(0 ; \vec{\imath}, \vec{\jmath}\right)\xspace}
186 \newcommand\Oij{\@ifstar{\@Oij}{\@Oij}}
187
188 \newcommand\@Oijk{%
189 \ensuremath{%
190 \left(0, \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
191 \vec{\vphantom{t}\smash{k}}\right)%
192 \xspace}
193 \newcommand\@Oijk{%
194 \ensuremath{%
195 \left(0 ; \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
196 \vec{\vphantom{t}\smash{k}}\right)%
197 \xspace}
198 \newcommand\Oijk{\@ifstar{\@Oijk}{\@Oijk}}
199
200 \newcommand\@Ouv{%
201 \ensuremath{\left(0, \vec{u}, \vec{v}\right)\xspace}
202 \newcommand\@Ouv{%
203 \ensuremath{\left(0 ; \vec{u}, \vec{v}\right)\xspace}
204 \newcommand\Ouv{\@ifstar{\@Ouv}{\@Ouv}}
205
206 \AtBeginDocument{
207 \renewcommand\ij{%
208 \ensuremath{\left(\vec{\imath}, \vec{\jmath}\right)\xspace}}
209 \newcommand\ijk{%
210 \ensuremath{%
211 \left(\vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
212 \vec{\vphantom{t}\smash{k}}\right)%
213 \xspace}

```

Références

- [1] *Lexique des règles typographiques en usage à l'Imprimerie Nationale*. Édition du 26/08/2002.
- [2] *Composition des textes scientifiques*. Inspection générale de mathématiques (IGEN-DESCO), 06/12/2001.
http://mslp.ac-dijon.fr/IMG/pdf/typo_txt_sci.pdf.
- [3] *Règles françaises de typographie mathématique*. Alexandre André, 02/09/2015. http://sgalex.free.fr/typo-maths_fr.pdf.
- [4] *Le petit typographe rationnel*. Eddie Saudrais, 20/03/2000.
- [5] *Typesetting mathematics for science and technology according to ISO 31/XI*, Claudio Beccari, TUGboat Volume 18 (1997), N° 1.
<http://www.tug.org/TUGboat/tb18-1/tb54becc.pdf>.
- [6] *Typefaces for Symbols in Scientific Manuscripts*.
<https://www.physics.nist.gov/cuu/pdf/typefaces.pdf>.
- [7] *On the Use of Italic and up Fonts for Symbols in Scientific Text*, I.M. Mills and W.V. Metanomski, ICTNS (Interdivisional Committee on Terminology, Nomenclature and Symbols), dec 1999.
https://old.iupac.org/standing/idcns/italic-roman_dec99.pdf.
- [8] *L^AT_EX Companion*. Frank Mittelbach, Michel Goossens, 2^e édition, Pearson Education France, 2005.
- [9] *The Not So Short Introduction to L^AT_EX 2_ε*. Manuel L^AT_EX de Tobias Oetiker, Hubert Partl, Irene Hyna et Elisabeth Schlegl, CTAN, v6.4 09/03/2021. <http://tug.ctan.org/info/lshort/english/lshort.pdf>.
- [10] *La distribution mafr*. Christian Obrecht, CTAN, v1.0 17/09/2006.
- [11] *L'extension tdsfrmath*. Yvon Henel, CTAN, v1.3 22/06/2009.
- [12] *Fourier-GUTenberg*. Michel Bovani, CTAN, v1.3 30/01/2005.
- [13] *The mathdesign package*. Paul Pichaureau, CTAN, v2.31 29/08/2013.
- [14] *Kp-Fonts – The Johannes Kepler project*. Christophe Caignaert, CTAN, v3.34 20/09/2022.
- [15] *The textalpha package* (partie de l'extension greek-fontenc). Günter Milde, CTAN, v2.1 14/06/2022.
- [16] *The upgreek package for L^AT_EX 2_ε*. Walter Schmidt, CTAN, v2.0 12/03/2003.
- [17] *isomath – Mathematical style for science and technology*. Günter Milde, CTAN, v0.6.1 2012/09/04.
- [18] *PM-ISOMath, The Poor Man ISO math bundle*, Claudio Beccari, CTAN, v1.2.00 2021/08/04.
- [19] *A Babel language definition file for French*. Extension L^AT_EX babel-french de Daniel Flipo, CTAN, v3.5c 14/09/2018.
- [20] *The icomma package for L^AT_EX 2_ε*. Walter Schmidt, CTAN, v2.0 10/03/2002.
- [21] *The nccomma package*. Alexander I. Rozhenko, CTAN, v1.0 10/02/2005.

- [22] *Intelligent commas*. Claudio Beccari, The PracT_EX Journal, 2011, No.1.
<https://tug.org/pracjourn/2011-1/beccari/Intcomma.pdf>.
- [23] *mismath* – *Miscellaneous mathematical macros*. Antoine Missier, CTAN, v1.10
25/10/2022.