

L’extension frenchmath*

Antoine Missier

antoine.missier@ac-toulouse.fr

22 janvier 2023

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, point-virgules et crochets. Depuis la version 2.0, des options permettent de composer les minuscules grecques du mode mathématique en forme droite.

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 polices Adobe Utopia, URW Garamond ou Bitstream Charter). Mais `frenchmath` fournit une solution générique s’adaptant à n’importe quelle police de caractères.

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`¹.

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 anglais 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`², des ensembles de nombres \mathbb{R} , \mathbb{N} , ... (pour \mathbf{R} , \mathbf{N} , ...) ainsi que celles relatives à la réalisation de feuilles d’exercices.

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

*Ce document correspond à `frenchmath` v2.4, dernière modification le 22/01/2023.

1. Nous proposons pour cela l’extension `mismath` [25] qui fournit diverses macros pour les mathématiques internationales.

2. Pour de jolis vecteurs on dispose de l’extension `esvect` d’Eddie Sautrais.

2 Utilisation

2.1 Majuscules mathématiques

En France, les lettres majuscules du mode mathématique doivent toujours être composées en romain (A, B, C, \dots) et non en italique ([1] p.107, voir aussi [2]). En utilisant \LaTeX ou \LuaTeX avec des polices mathématiques OpenType, cette convention est assez commode à mettre en œuvre; par contre, avec \LaTeX ou \pdfLaTeX , assez peu d’auteurs la respectent et les extensions précitées ne fonctionnent qu’avec des polices particulières. 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’option `capsit` de `frenchmath` permet de désactiver la composition des majuscules du mode mathématique en romain pour conserver la composition par défaut (en italique) : `\usepackage[capsit]{frenchmath}` 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 aussi claire et les auteurs divergent sur ce point. Plusieurs recommandent l’usage des lettres grecques minuscules en forme droite [12] [13] [19], 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 [18] expose 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 (ou des majuscules en italique). Citons également `newpxmath`, `newtxmath`³ et `libertinust1math` de Michael Sharpe, `pxgrecs`,

3. L’extension `newtxmath` doit être chargée après `frenchmath` qui utilise `amssymb` car la compilation produit sinon un message d’erreur pour la commande `\Bbbk`.

txgreek⁴ et libgreek de Jean-François Burnol, qui donnent de beaux résultats pour une utilisation avec respectivement les polices Palatino, Times et Libertinus.

[lgrmath] Jean-François Burnol a également développé l’extension lgrmath [16] qui permet d’utiliser, en mode mathématique, les différentes fontes de lettres grecques accessibles par L^AT_EX avec l’encodage LGR. La documentation de l’extension indique comment consulter et utiliser les fontes accessibles sur votre distribution.

En activant l’option lgrmath, frenchmath charge cette extension avec l’option style=french et la fonte fcm (de l’extension cm-lgc)⁵ qui se marie particulièrement bien avec la police usuelle Latin Modern. Les commandes `\alpha`, `\beta`, etc. produisent alors automatiquement les lettres en forme droite α , β , ..., π , etc., tandis que `\alphait`, `\betait`, etc. produisent des formes italiques α , β , ..., π , etc. Ces dernières sont peu à notre goût, mais elles n’ont pas vocation à être utilisées lorsque l’on active l’option lgrmath. Par contre, on peut choisir d’autres options de fontes en chargeant l’extension lgrmath indépendamment de frenchmath (voir par exemple avec l’option font=Alegreya-LF ou font=Cochineal-LF).

[upgreek] Avec la même philosophie, frenchmath fournit l’option upgreek basée sur l’extension upgreek de Walter Schmidt [17] qui donne accès aux lettres grecques minuscules en forme droite : `\upalpha`, `\upbeta`, etc. 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 police Adobe Symbol), il faut charger l’extension upgreek avec l’option souhaitée *avant* frenchmath⁶. L’option upgreek de frenchmath redéfinit ensuite les commandes `\alpha`, `\beta`, etc. pour produire automatiquement les lettres en forme droite α , β , ..., π , ...; les formes italiques, α , β , ..., π , etc. restant cependant disponibles avec les commandes `\italpha`, `\itbeta`, ..., `\itpi`, etc.

[Upgreek] Avec L^AT_EX, les lettres grecques majuscules sont automatiquement composées en forme droite et l’option upgreek ne concerne que les minuscules. Cependant l’extension upgreek propose aussi `\Upgamma`, ..., `\Upomega` : Γ , ..., Ω . Si l’on veut conserver majuscules et minuscules dans le même style, l’option Upgreek 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 .

L’option Upgreek 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 également bien avec la police Latin Modern. Par contre le theta produit, ϑ , n’est pas vraiment celui qui est d’usage en mathématiques.

4. Si on utilise amsmath (ou mismath), pxgreek ou txgreek doit être chargée *après* amsmath (ou mismath), pour éviter une erreur de compilation due à la redéfinition des commandes `\iint`, `\iiint`, `\idotsint`.

5. Évidemment il faut que cm-lgc soit installée sur votre distribution sans quoi la fonte de substitution LGR/cmr/m/n sera utilisée.

6. L’option Symbol de upgreek se marie mieux avec une police comme Times par exemple.

Mentionnons ce commentaire de Walter Schmidt [17] 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

virgules Dans le mode mathématique de \LaTeX , 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` [20], 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 :

- avec `icomma` (intelligent comma) de Walter Schmidt [21], 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 [22], 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`. Depuis la version 1.6, `frenchmath` charge `nccomma`, grâce à un code proposé par Jean-François Burnol qui permet d’utiliser conjointement `nccomma` avec l’option `autolanguage` de `numprint`¹⁰. Signalons que, si l’on compile avec \LuaTeX , `nccomma` (et donc `frenchmath`) doit être chargé *après* `babel-french` (ce qui est, somme toute, la pratique normale), sinon `babel` génère une erreur.

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.

point-virgule 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 « : ».

crochets Alors que les Anglais utilisent généralement les parenthèses pour les intervalles ouverts $(0, +\infty)$, l’usage en français est d’utiliser les crochets $]0, +\infty[$.

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.

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 dilatée [1], légèrement plus grande que l’espace que l’on obtient sans cette option.

10. Mentionnons aussi l’article *Intelligent commas* de Claudio Beccari [23] qui propose une solution simplifiée par rapport à `nccomma` mais dont l’avantage semble discutable.

Mais comme cela n'est pas prévu par \LaTeX , les espaces seront souvent incorrectes. Nous avons redéfini les crochets dans l'extension `ibackets` [24] qui est chargée par `frenchmath`, sauf si l'on active l'option `noibackets`¹¹. Le code `$\$x\in]-\pi, 0[\cup]2\pi, 3\pi[$` produira

$$x \in]-\pi, 0[\cup]2\pi, 3\pi[\quad \text{avec } \text{ibackets},$$

au lieu de

$$x \in]-\pi, 0[\cup]2\pi, 3\pi[\quad \text{sans } \text{ibackets}.$$

Avec `ibackets`, un crochet devient un caractère ordinaire, sauf s'il est immédiatement suivi par un signe + ou - (sans espace), auquel cas c'est un délimiteur ouvrant. Si la borne de gauche possède un signe - (ou +), *il ne faut pas laisser d'espace entre le premier crochet et le signe* : par exemple `$\$x \in]-\infty, 0[$` produit $x \in]-\infty, 0[$ avec des espaces trop grandes autour du signe. Mais au contraire lorsque l'on souhaite faire de l'algèbre sur les intervalles, *il faut laisser une espace entre le second crochet et l'opération + ou -*, par exemple, `$\$[a, b] + [c, d]$` produit $[a, b] + [c, d]$ mais `$\$[a, b] + [c, d]$` produit $[a, b] + [c, d]$.

En cas de comportement problématique, par exemple si une coupure de ligne se produit entre les deux crochets d'un intervalle, il est toujours possible de transformer alors ces crochets en délimiteurs avec `\left` et `\right`.

2.4 Quelques macros et alias utiles

<code>\curs</code>	Les lettres cursives ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \dots$), provenant de l'extension <code>mathrsfs</code> chargée par <code>frenchmath</code> , sont composées avec <code>\curs</code> et sont différentes de celles obtenues avec <code>\mathcal</code> ¹² ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \dots$). La commande <code>\curs</code> permet aussi de composer ces lettres en mode texte, sans les délimiteurs du mode mathématique.* ¹³
<code>\infeg</code> <code>\supeg</code>	Les relations \leq et \geq s'obtiennent avec les commandes <code>\infeg</code> et <code>\supeg</code> et diffèrent des versions anglaises de <code>\leq</code> (\leq) et <code>\geq</code> (\geq). Ce sont des alias de <code>\leqslant</code> et <code>\geqslant</code> de l'extension <code>amssymb</code> , chargée par <code>frenchmath</code> .*
<code>\vide</code>	Le symbole de l'ensemble vide \emptyset s'obtient avec <code>\vide</code> (alias de la commande <code>\varnothing</code> de l'extension <code>amssymb</code>) ; il diffère de celui obtenu avec <code>\emptyset</code> (particulièrement laid dans la fonte par défaut Computer Modern : \emptyset).*
<code>\paral</code>	La commande <code>\paral</code> fournit la relation ¹⁴ du parallélisme : $\mathcal{D} \parallel \mathcal{D}'$, plutôt que sa version anglaise <code>\parallel</code> : $\mathcal{D} \parallel \mathcal{D}'$.*
<code>\ssi</code>	La commande <code>\ssi</code> produit « si, et seulement si, ».
<code>\cmod</code>	Le modulo se compose normalement entre parenthèses, avec <code>\pmod</code> , mais on rencontre aussi, en français, le modulo entre crochets, ce que permet la commande <code>\cmod</code> en respectant le bon espacement propre au modulo : $53 \equiv 5 \pmod{12}$.

11. D'autres solutions existent, par exemple avec l'extension `interval` ou encore avec la macro `\DeclarePairedDelimiter` de `mathtools`, mais utilisée avec des crochets, cette dernière est incompatible avec `ibackets`, d'où la possibilité de désactiver `ibackets`.

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

13. L'astérisque en fin d'item signale une fonctionnalité similaire dans `mafr`.

14. Pour noter que deux objets sont perpendiculaires, on utilise `\perp` : $\mathcal{D} \perp \mathcal{D}'$, défini comme une relation mathématique plutôt que `\bot` défini comme un symbole (les espacements diffèrent).

2.5 Identifiants de « fonctions » classiques

<code>\pgcd</code> <code>\ppcm</code>	En arithmétique, nous avons les classiques <code>\pgcd</code> et <code>\ppcm</code> , qui diffèrent de leur version anglaise <code>\gcd</code> et <code>\lcm</code> ¹⁵ .
<code>\card</code> <code>\Card</code>	Pour le cardinal d'un ensemble, nous proposons <code>\card</code> , cité dans [1] et [3], ou <code>\Card</code> , qui est aussi d'usage courant (cf. Wikipedia).
<code>\Ker</code> <code>\Hom</code>	L ^A T _E X fournit les macros <code>\ker</code> et <code>\hom</code> , alors que l'usage français est souvent de commencer ces noms par une majuscule pour obtenir <code>Ker</code> ¹⁶ et <code>Hom</code> .
<code>\rg</code> <code>\Vect</code>	Le rang d'une application linéaire ou d'une matrice (<code>rg</code>) s'obtient avec la commande <code>\rg</code> et l'espace vectoriel engendré par une famille de vecteurs avec <code>\Vect</code> .
<code>\ch</code> <code>\sh</code> <code>\th</code>	En principe, les fonctions hyperboliques s'écrivent en français avec les macros L ^A T _E X standard <code>\cosh</code> , <code>\sinh</code> , <code>\tanh</code> . Néanmoins les écritures $\operatorname{ch} x$, $\operatorname{sh} x$ et $\operatorname{th} x$, qui sont la norme avec les langues d'Europe de l'Est [8], sont aussi utilisées en français [1]. On les obtient avec les commandes <code>\ch</code> , <code>\sh</code> et <code>\th</code> ¹⁷ .
<code>\cosec</code> <code>\cosech</code>	La fonction cosécante (inverse du sinus) s'obtient avec la macro <code>\csc</code> , mais en français, on utilise aussi <code>\cosec</code> [1] et <code>\cosech</code> pour la cosécante hyperbolique ¹⁸ .

2.6 Bases et repères

<code>\0ij</code> <code>\0ijk</code> <code>\0uv</code>	Les repères classiques du plan ou de l'espace seront composés avec des hauteurs de flèches homogénéisées : <code>\0ij</code> compose (O, \vec{i}, \vec{j}) , <code>\0ijk</code> compose $(O, \vec{i}, \vec{j}, \vec{k})$ et <code>\0uv</code> 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.
<code>\0ij*</code> <code>\0ijk*</code> <code>\0uv*</code>	Les versions étoilées utilisent le point-virgule et non la virgule comme séparateur après le point O, comme mentionné dans [1]. On obtient $(O; \vec{i}, \vec{j})$, $(O; \vec{i}, \vec{j}, \vec{k})$, $(O; \vec{u}, \vec{v})$.
<code>\ij</code> <code>\ijk</code>	Enfin les macros <code>\ij</code> ¹⁹ et <code>\ijk</code> composent les bases du plan et de l'espace, (\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 l'extension `mathptmx` (basée sur la fonte de texte Times).

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

16. 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.

17. La commande `\th` existe déjà, pour le mode texte uniquement, et produit \mathfrak{t} ; elle a été redéfinie, uniquement pour le mode mathématique, à la place de l'ancienne commande `\tgh`, utilisée jusqu'à la version 1.6, désormais obsolète.

18. La fonction sécante est définie en standard par L^AT_EX avec `\sec` et la sécante hyperbolique `\sech` est définie par `mismath` [25].

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

3 Le code

```
1 \RequirePackage{ifthen}
2 \newboolean{capsit}
3 \DeclareOption{capsit}{\setboolean{capsit}{true}}
4 \newboolean{lgrmath}
5 \DeclareOption{lgrmath}{\setboolean{lgrmath}{true}}
6 \newboolean{upgreek}
7 \DeclareOption{upgreek}{\setboolean{upgreek}{true}}
8 \newboolean{Upgreek}
9 \DeclareOption{Upgreek}{\setboolean{Upgreek}{true}}
10 \setboolean{upgreek}{true}}
11 \newboolean{noibrackets}
12 \DeclareOption{noibrackets}{\setboolean{noibrackets}{true}}
13 \ProcessOptions \relax
14
15 \AtBeginDocument{
16   \@ifpackageloaded{mathdesign}{
17     \PackageWarningNoLine{frenchmath}{Package mathdesign
18       is loaded, \MessageBreak
19       I don't load mathrsfs and amssymb packages}
20   }{
21     \RequirePackage{mathrsfs} % fournit les majuscules cursives
22     \RequirePackage{amssymb} % \leqslant, \geqslant, \varnothing
23   }
24 }
25 \RequirePackage{amsopn} % fournit \DeclareMathOperator
26 \ifthenelse{\boolean{lgrmath}}{
27   \@ifpackageloaded{lgrmath}{
28     \RequirePackage[font=fcm,style=french]{lgrmath}
29   }{
30 \ifthenelse{\boolean{upgreek}}{
31   \@ifpackageloaded{upgreek}{\RequirePackage[Euler]{upgreek}}
32 }{
33 \RequirePackage{xspace} % utile pour les commandes \curs, \ssi, \0ij
34 \ifthenelse{\boolean{noibrackets}}{
35   \RequirePackage{ibrackets} % intelligent brackets
36 % \RequirePackage{icomma} % intelligent comma
37 \RequirePackage{nccomma} % depuis frenchmath 1.6
38 \@ifpackagewith{babel}{french}{
39   \RequirePackage{iftex}
40   \ifluatex
41     \PackageWarningNoLine{frenchmath}{With LuaTeX engine,
42       \MessageBreak
43       load babel before frenchmath}
44   \fi
45 }
46 \AtBeginDocument{\mathcode'\,="8000\relax
47   \@ifpackageloaded{babel}{
48     \addto\extrasfrench{\mathcode'\,="8000\relax}
```

```

49   }{}
50 }

```

La macro ci-dessus, qui m’a été proposée par Jean-François Burnol, permet d’utiliser `nccomma` à la place de `icomma`. L’extension `nccomma` doit être chargée après `babel-french` si on utilise `Lua®TeX`, sinon une erreur de compilation se produit.

Je dois également à Jean-François Burnol une 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.

```

51
52 \ifthenelse{\boolean{capsit}}{}{
53   \AtBeginDocument{
54     \DeclareMathSymbol{A}\mathalpha{operators}{‘A}
55     \DeclareMathSymbol{B}\mathalpha{operators}{‘B}
56     \DeclareMathSymbol{C}\mathalpha{operators}{‘C}
57     \DeclareMathSymbol{D}\mathalpha{operators}{‘D}
58     \DeclareMathSymbol{E}\mathalpha{operators}{‘E}
59     \DeclareMathSymbol{F}\mathalpha{operators}{‘F}
60     \DeclareMathSymbol{G}\mathalpha{operators}{‘G}
61     \DeclareMathSymbol{H}\mathalpha{operators}{‘H}
62     \DeclareMathSymbol{I}\mathalpha{operators}{‘I}
63     \DeclareMathSymbol{J}\mathalpha{operators}{‘J}
64     \DeclareMathSymbol{K}\mathalpha{operators}{‘K}
65     \DeclareMathSymbol{L}\mathalpha{operators}{‘L}
66     \DeclareMathSymbol{M}\mathalpha{operators}{‘M}
67     \DeclareMathSymbol{N}\mathalpha{operators}{‘N}
68     \DeclareMathSymbol{O}\mathalpha{operators}{‘O}
69     \DeclareMathSymbol{P}\mathalpha{operators}{‘P}
70     \DeclareMathSymbol{Q}\mathalpha{operators}{‘Q}
71     \DeclareMathSymbol{R}\mathalpha{operators}{‘R}
72     \DeclareMathSymbol{S}\mathalpha{operators}{‘S}
73     \DeclareMathSymbol{T}\mathalpha{operators}{‘T}
74     \DeclareMathSymbol{U}\mathalpha{operators}{‘U}
75     \DeclareMathSymbol{V}\mathalpha{operators}{‘V}
76     \DeclareMathSymbol{W}\mathalpha{operators}{‘W}
77     \DeclareMathSymbol{X}\mathalpha{operators}{‘X}
78     \DeclareMathSymbol{Y}\mathalpha{operators}{‘Y}
79     \DeclareMathSymbol{Z}\mathalpha{operators}{‘Z}
80   }
81 }

```

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`, ... `Upgreek` transforme en outre les majuscules grecques pour garder le même style.

```

82
83 \ifthenelse{\boolean{upgreek}}{
84   \@ifundefined{italpha}{\let\italpha\alpha}{

```



```

85     \PackageWarningNoLine{frenchmath}{Command
86         \string\italpha\space already exist \MessageBreak
87         and will not be redefined, \MessageBreak
88         no more warning for the other Greek letters, \MessageBreak
89         except pi}
90 }
91 \@ifundefined{itbeta}{\let\itbeta\beta}{}
92 \@ifundefined{itgamma}{\let\itgamma\gamma}{}
93 \@ifundefined{itdelta}{\let\itdelta\delta}{}
94 \@ifundefined{itepsilon}{\let\itepsilon\epsilon}{}
95 \@ifundefined{itzeta}{\let\itzeta\zeta}{}
96 \@ifundefined{iteta}{\let\iteta\eta}{}
97 \@ifundefined{ittheta}{\let\ittheta\theta}{}
98 \@ifundefined{itiota}{\let\itiota\iota}{}
99 \@ifundefined{itkappa}{\let\itkappa\kappa}{}
100 \@ifundefined{itlambda}{\let\itlambda\lambda}{}
101 \@ifundefined{itmu}{\let\itmu\mu}{}
102 \@ifundefined{itnu}{\let\itnu\nu}{}
103 \@ifundefined{itxi}{\let\itxi\xi}{}
104 \@ifundefined{itpi}{\let\itpi\pi}{
105     \PackageWarningNoLine{frenchmath}{Command
106         \string\itpi\space already exist \MessageBreak
107         and will not be redefined}
108 }
109 \@ifundefined{itrho}{\let\itrho\rho}{}
110 \@ifundefined{itsigma}{\let\itsigma\sigma}{}
111 \@ifundefined{ittau}{\let\ittau\tau}{}
112 \@ifundefined{itupsilon}{\let\itupsilon\upsilon}{}
113 \@ifundefined{itphi}{\let\itphi\phi}{}
114 \@ifundefined{itchi}{\let\itchi\chi}{}
115 \@ifundefined{itpsi}{\let\itpsi\psi}{}
116 \@ifundefined{itomega}{\let\itomega\omega}{}
117 \@ifundefined{itvarepsilon}{\let\itvarepsilon\varepsilon}{}
118 \@ifundefined{itvartheta}{\let\itvartheta\vartheta}{}
119 \@ifundefined{itvarpi}{\let\itvarpi\varpi}{}
120 \@ifundefined{itvarsigma}{\let\itvarsigma\varsigma}{}
121 \@ifundefined{itvarphi}{\let\itvarphi\varphi}{}
122 }{}
123
124 \ifthenelse{\boolean{upgreek}}{
125     \renewcommand\alpha{\upalpha}
126     \renewcommand\beta{\upbeta}
127     \renewcommand\gamma{\upgamma}
128     \renewcommand\delta{\updelta}
129     \renewcommand\epsilon{\upepsilon}
130     \renewcommand\zeta{\upzeta}
131     \renewcommand\eta{\upeta}
132     \renewcommand\theta{\uptheta}
133     \renewcommand\iota{\upiota}
134     \renewcommand\kappa{\upkappa}

```

```

135 \renewcommand\lambda{\uplambda}
136 \renewcommand\mu{\upmu}
137 \renewcommand\nu{\upnu}
138 \renewcommand\xi{\upxi}
139 \renewcommand\pi{\uppi}
140 \renewcommand\rho{\uprho}
141 \renewcommand\sigma{\upsigma}
142 \renewcommand\tau{\uptau}
143 \renewcommand\upsilon{\upupsilon}
144 \renewcommand\phi{\upphi}
145 \renewcommand\chi{\upchi}
146 \renewcommand\psi{\uppsi}
147 \renewcommand\omega{\upomega}
148 \renewcommand\varepsilon{\upvarepsilon}
149 \renewcommand\vartheta{\upvartheta}
150 \renewcommand\varpi{\upvarpi}
151 \renewcommand\varrho{\upvarrho}
152 \renewcommand\varsigma{\upvarsigma}
153 \renewcommand\varphi{\upvarphi}
154 }{}
155
156 \ifthenelse{\boolean{Upgreek}}{
157 \renewcommand\Gamma{\Upgamma}
158 \renewcommand\Delta{\Updelta}
159 \renewcommand\Theta{\Uptheta}
160 \renewcommand\Lambda{\Uplambda}
161 \renewcommand\Xi{\Upxi}
162 \renewcommand\Pi{\Uppi}
163 \renewcommand\Sigma{\Upsigma}
164 \renewcommand\Upsilon{\Upupsilon}
165 \renewcommand\Phi{\Upphi}
166 \renewcommand\Psi{\Uppsi}
167 \renewcommand\Omega{\Upomega}
168 }{}
169
170 \AtBeginDocument{\@ifpackageloaded{pstricks-add}{\psset{comma=true}}{}}
171 \DeclareMathSymbol{;}{\mathbin}{operators}{'73} % \mathpunct à l'origine
172
173 \newcommand*\curs[1]{\ensuremath{\mathscr{#1}}\xspace}
174 \newcommand\infeg{\leqslant}
175 \newcommand\supeg{\geqslant}
176 \newcommand\vide{\varnothing}
177 \newcommand\paral{\mathrel{\ooalign{$\mkern-1.75mu/\mkern1.75mu$\cr%
178 \mkern1.75mu/\mkern-1.75mu$}}}
179 }

```

Cette définition remplace, depuis la version 2.2, l'ancienne définition plus simple `\mathrel{/!\!/}`, mais qui donnait des barres trop serrées avec `mathastext + times` ou avec `libertinust1math`. Merci à Jean-François Burnol de me l'avoir fait remarquer et pour ses suggestions dans la mise au point d'une macro plus efficace.

```

180\newcommand\ssi{si, et seulement si,\xspace}
181\newcommand*\cmod[1]{\quad[#1]}
182
183\DeclareMathOperator{\pgcd}{pgcd}
184\DeclareMathOperator{\ppcm}{ppcm}
185\DeclareMathOperator{\card}{card}
186\DeclareMathOperator{\Card}{Card}
187\DeclareMathOperator{\Ker}{Ker}
188\DeclareMathOperator{\Hom}{Hom}
189\DeclareMathOperator{\rg}{rg}
190\DeclareMathOperator{\Vect}{Vect}
191\DeclareMathOperator{\ch}{ch}
192\DeclareMathOperator{\sh}{sh}
193\AtBeginDocument{\let\oldth\th %\th existe déjà (mode texte)
194  \renewcommand\th{\TextOrMath{\oldth}{\operatorname{th}}}}
195\DeclareMathOperator{\cosec}{cosec}
196\DeclareMathOperator{\cosech}{cosech}
197
198\newcommand\@Oij{%
199  \ensuremath{\left(0, \vec{\imath}, \vec{\jmath}\right)\xspace}
200\newcommand\@OOij{%
201  \ensuremath{\left(0 ; \vec{\imath}, \vec{\jmath}\right)\xspace}
202\newcommand\Oij{\@ifstar{\@OOij}{\@Oij}}
203
204\newcommand\@Oijk{%
205  \ensuremath{%
206    \left(0, \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
207    \vec{\vphantom{t}\smash{k}}\right)%
208  \xspace}
209\newcommand\@OOijk{%
210  \ensuremath{%
211    \left(0 ; \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
212    \vec{\vphantom{t}\smash{k}}\right)%
213  \xspace}
214\newcommand\Oijk{\@ifstar{\@OOijk}{\@Oijk}}
215
216\newcommand\@Ouv{%
217  \ensuremath{\left(0, \vec{u}, \vec{v}\right)\xspace}
218\newcommand\@OOuv{%
219  \ensuremath{\left(0 ; \vec{u}, \vec{v}\right)\xspace}
220\newcommand\Ouv{\@ifstar{\@OOuv}{\@Ouv}}
221
222\AtBeginDocument{
223  \renewcommand\ij{%
224    \ensuremath{\left(\vec{\imath}, \vec{\jmath}\right)\xspace}}
225\newcommand\ijk{%
226  \ensuremath{%
227    \left(\vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
228    \vec{\vphantom{t}\smash{k}}\right)%
229  \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.
<https://www.gutenberg-asso.fr/IMG/pdf/saudrais-typo.pdf>
- [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*, 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 lgrmath package*, Jean-François B., CTAN, v1.0 16/11/2022.
- [17] *The upgreek package for L^AT_EX 2_ε*, Walter Schmidt, CTAN, v2.0 12/03/2003.
- [18] *isomath – Mathematical style for science and technology*, Günter Milde, CTAN, v0.6.1 2012/09/04.
- [19] *PM-ISomath – The Poor Man ISO math bundle*, Claudio Beccari, CTAN, v1.2.00 2021/08/04.
- [20] *A Babel language definition file for French*, extension L^AT_EX babel-french de Daniel Flipo, CTAN, v3.5c 14/09/2018.

- [21] *The icomma package for L^AT_EX 2_ε*. Walter Schmidt, CTAN, v2.0 10/03/2002.
- [22] *The nccomma package*. Alexander I. Rozhenko, CTAN, v1.0 10/02/2005.
- [23] *Intelligent commas*. Claudio Beccari, The PracT_EX Journal, 2011, No.1.
<https://tug.org/pracjourn/2011-1/beccari/Intcomma.pdf>
- [24] *Intelligent brackets – The ibrackets package*, Antoine Missier, v1.0 19/12/2022.
- [25] *mismath – Miscellaneous mathematical macros*, Antoine Missier, CTAN, v2.1 26/12/2022.