L'extension frenchmath*

Antoine Missier
antoine.missier@ac-toulouse.fr

4 mai 2024

Table des matières

1 Introduction

Cette extension, inspirée à l'origine de mafr de Christian Obrecht [10], permet le respect des règles typographiques des mathématiques françaises, en particulier l'obtention automatique des majuscules en romain (lettres droites) plutôt qu'en italique (voir [1] et [2]), et elle gère correctement les espacements pour les virgules, point-virgules et crochets. L'extension fournit en outre diverses macros francisées [1]. Depuis la version 2.0, frenchmath propose en outre des options permettant de composer les minuscules grecques du mode mathématique en forme droite.

D'autres solutions existent pour composer les majuscules mathématiques en romain, par exemple avec XeLaTeX ou LuaLaTeX et l'extension unicode-math [14]. Pour pdfLaTeX nous avons les extensions fourier de Michel Bovani [15] (avec la famille des fontes Adobe Utopia), mathdesign de Paul Pichaureau [16] (avec les polices Adobe Utopia, URW Garamond ou Bitstream Charter) ou encore kpfonts de

*Ce document correspond à frenchmath v3.0, dernière modification le 04/05/2024.

1. Contrairement à mafr nous avons choisi de ne pas conserver le même nom de commande pour substituer des symboles français aux symboles anglais.

Christophe Caignaert [17]. Mais frenchmath fournit une solution générique s'adaptant à n'importe quelle police de caractères (et compatible avec unicode-math).

Certaines 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], tout comme diverses commandes que l'on trouve dans mafr et qui ne sont pas spécifiques aux mathématiques françaises : c'est le cas de `\vect` [3], des ensembles de nombres `\R`, `\N`... (pour $\mathbf{R}, \mathbf{N}\ldots$) ainsi que celles relatives à la réalisation de feuilles d'exercices.

Mentionnons par ailleurs l'extension tdsfrmath de Yvon Henel [12] qui fournit également beaucoup de commandes francisées ou encore tablvar [13] qui permet de réaliser de jolis tableaux de variations, spécificité des mathématiques françaises.

2 Utilisation

2.1 Majuscules mathématiques

Dans les mathématiques françaises, pour l'alphabet latin, « les lettres majuscules sont toujours composées en romain » ($\mathrm{A}, \mathrm{B}, \mathrm{C}\ldots$) et non en italique (cf. [1] p.107, voir aussi [2]). En utilisant XeLaTeX ou LuaLaTeX, avec l'option `math-style=french` de l'extension unicode-math, cette convention est commode à mettre en œuvre, par contre, avec LaTeX ou pdfLaTeX, elle est peu respectée et les extensions comme mathdesign (avec `uppercase=upright`), fourier (avec `upright`) ou kpfonts (avec `uprightRoman`) ne fonctionnent qu'avec des polices particulières. Par défaut frenchmath compose automatiquement les majuscules mathématiques latines en romain, quelle que soit la police utilisée. Par exemple `\[ P(X)=\sum_{i=0}^{n} a_i X^i \]` donne avec frenchmath

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

`[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 ponctuellement l'aspect d'une lettre particulière, avec les macros LaTeX `\mathrm` et `\mathit`. D'autre part l'extension mismath [18] fournit deux bascules puissantes `\MathUp` et `\MathIt` qui agissent de manière globale (ou locale dans un environnement) et permettent à tout moment de changer la « famille » d'une lettre particulière ; une commande générique `\apply` permet y compris d'appliquer ces macros sur une liste. Ainsi `\apply\MathIt{F,G,X}` remettra en italique les lettres F, G et X.

2. Nous proposons pour cela l'extension mismath [18] qui fournit diverses macros pour les mathématiques internationales.

3. Pour de jolis vecteurs on dispose de l'extension esvect [19] d'Eddie Saudrais.

2.2 Virgule, point-virgule et crochets

`virgule`

Dans le mode mathématique de LaTeX, la virgule est toujours, par défaut, un symbole de ponctuation et sera donc suivie d'une espace. Ceci est légitime dans une liste ou un intervalle : `$[a,b]$` donne $[a, b]$. Mais, en français, la virgule sert aussi de séparateur décimal pour les nombres et ne doit, dans ce cas, pas être suivie d'espace ; or `$12,5$` donne $12, 5$ au lieu de 12,5. L'extension babel, avec l'option `french` [11], 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 bascules [4]. En mode mathématique :

- 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 ncccomma 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 est plus souple, néanmoins ncccomma ne fonctionne pas avec babel-french utilisé conjointement avec l'option `autolanguage` [5] de l'extension numprint. En outre ncccomma ne fonctionne pas non plus avec l'extension unicode-math (et bugue à l'appel `\setmathfont`). Dans son article *Intelligent commas* [22], Claudio Beccari propose une autre solution, voisine de ncccomma, mais qui produit le même type d'incompatibilités. Le code a donc été revu afin de régler ces incompatibilités. Comme de nombreux pays utilisent la virgule comme séparateur décimal, il fait l'objet d'une extension séparée, decimalcomma [23], qui est chargée par frenchmath (depuis la version 2.7). Bien que decimalcomma ne fonctionne avec unicode-math que s'il est chargé après [6], frenchmath s'est affranchi de cette contrainte (depuis la version 2.11).

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 $]0, +\infty[$. Mais comme cela n'est pas prévu par LaTeX, les espaces seront souvent incorrectes. Nous avons redéfini les crochets dans l'extension ibrackets [24] qui est chargée

4. Dans ce cas il ne faut pas utiliser les bascules, au risque de rendre ces extensions inopérantes.
5. 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.
6. L'extension icomma présente la même limitation et doit être chargée après unicode-math.

[noibrackets] par frenchmath, sauf si on la désactive avec l'option `noibrackets`[7]. Le code `$x\in ]-\pi, 0[ \cup ]2\pi, 3\pi[$` produira

$$x \in]-\pi, 0[\cup]2\pi, 3\pi[\quad \text{avec ibrackets,}$$
$$\text{au lieu de} \quad x \in]-\pi, 0[\cup]2\pi, 3\pi[\quad \text{sans ibrackets.}$$

Avec ibrackets, 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 $x \in]-\infty, 0]$ au lieu de $x \in]-\infty, 0]$. Mais au contraire lorsque l'on veut faire de l'algèbre sur les intervalles, *il faut laisser une espace entre le second crochet et l'opération* + ou –, par exemple, `$[a,b] +[c,d]$` donne $[a, b] + [c, d]$ 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`.

2.3 Quelques macros et alias utiles

`\curs` Les lettres cursives ($\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D} \ldots$), sont composées avec `\mathscr`, ou son alias `\curs`, et sont différentes de celles obtenues avec `\mathcal`[8] ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D} \ldots$). En principe frenchmath charge l'extension mathrsfs qui fournit ces lettres cursives, sauf si la commande `\mathscr` est définie par ailleurs, en particulier si on utilise l'extension mathdesign [16] ou l'option `scr` de mathalpha [26][9].

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

`\vide` Le symbole de l'ensemble vide ? 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 classique Latin Modern : $\emptyset$.

`\AMSvarnothing` Avec unicode-math et la fonte mathématique Latin Modern Math (appelée avec `\setmathfont`), `\varnothing` produit le même glyphe laid que `\emptyset`. Pour corriger cela, on dispose de la commande `\AMSvarnothing`, qui doit être appelée dans le préambule et permet d'obtenir le même glyphe que celui de amssymb.

`\paral` La commande `\paral` fournit la *relation*[10] du parallélisme : $\mathscr{D} /\!/ \mathscr{D}'$, plutôt que sa version anglaise `\parallel` : $\mathscr{D} \parallel \mathscr{D}'$.

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

7. D'autres solutions existent, par exemple avec l'extension interval ou encore avec la macro `\DeclarePairedDelimiter` de mathtools [25], mais utilisée avec des crochets, cette dernière est incompatible avec ibrackets, d'où la possibilité de désactiver ibrackets.

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

9. L'extension mathalpha de Michael Sharpe permet d'accéder à différentes variantes élégantes de lettres calligraphiques, par exemple avec les options `scr=boondox`, `scr=rsfso` ou `scr=kp`.

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

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

2.4 Identifiants de « fonctions » classiques

\pgcd \ppcm En arithmétique, nous avons les classiques \pgcd et \ppcm, qui diffèrent de leur version anglaise \gcd et \lcm [11].

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

\Ker \Hom LaTeX fournit les macros \ker et \hom, alors que l'usage français est souvent de commencer ces noms par une majuscule pour obtenir Ker [12] et Hom.

\rg \Vect Le rang d'une application linéaire ou d'une matrice (rg) s'obtient avec la commande \rg et l'espace vectoriel engendré par une famille de vecteurs avec \Vect.

\ch \sh \th En principe, les fonctions hyperboliques s'écrivent en français avec les macros LaTeX standard \cosh, \sinh, \tanh. 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 (voir [32]), sont aussi utilisées en français [1]. On les obtient avec les commandes \ch, \sh et \th [13].

\cosec \cosech La fonction cosécante (inverse du sinus) s'obtient avec la macro \csc, mais en français, on utilise aussi \cosec [1] et \cosech pour la cosécante hyperbolique [14].

2.5 Bases et repères

\Oij \Oijk \Ouv Les repères classiques du plan ou de l'espace seront composés avec des hauteurs de flèches homogénéisées : \Oij compose $(\mathrm{O}, \vec{\imath}, \vec{\jmath})$, \Oijk compose $\left(\mathrm{O}, \vec{\imath}, \vec{\jmath}, \vec{k}\right)$ et \Ouv compose $(\mathrm{O}, \vec{u}, \vec{v})$ (utilisé dans le plan complexe). Ces commandes peuvent être utilisées sans expliciter les délimiteurs du mode mathématique.

\Oij* \Oijk* \Ouv* Les versions étoilées utilisent le point-virgule et non la virgule comme séparateur après le point O, comme mentionné dans [1]. On obtient $(\mathrm{O}\,;\vec{\imath},\vec{\jmath})$, $\left(\mathrm{O}\,;\vec{\imath},\vec{\jmath},\vec{k}\right)$, $(\mathrm{O}\,;\vec{u},\vec{v})$.

\ij \ijk Enfin les macros \ij [15] et \ijk composent les bases du plan et de l'espace, $(\vec{\imath},\vec{\jmath})$ et $\left(\vec{\imath},\vec{\jmath},\vec{k}\right)$, en homogénéisant la hauteur des flèches.

Signalons que, pour l'extension mathptmx (basée sur la police de texte Times), \jmath n'est pas disponible, mais frenchmath contourne ce problème en chargeant alors dotlessj de David Carlisle [27], ce qui permet aux macros ci-dessus de fonctionner normalement.

11. Cette dernière n'est pas implémentée en standard dans LaTeX (mais dans mismath [18]).
12. 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.
13. La commande \th existe déjà, pour le mode texte uniquement, et produit þ ; elle a été redéfinie, uniquement pour le mode mathématique.
14. La fonction sécante est définie en standard par LaTeX avec \sec et la sécante hyperbolique \sech est définie par mismath [18].
15. Notons que la macro \ij existe déjà (ligature entre i et j pour le hollandais) ; elle a été redéfinie pour le mode mathématique uniquement, qui doit donc ici être explicite : $\ij$.

2.6 Lettres grecques

Dans les mathématiques françaises, la norme concernant l'usage des lettres grecques minuscules en italique ou en forme droite ne semble pas aussi claire que pour les lettres romaines et il y a parfois divergence sur ce point. Beaucoup recommandent l'usage des lettres grecques minuscules en forme droite ([15] [16] [9]), mais certains auteurs 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 celle énoncée 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 [8] expose différentes méthodes pour obtenir les lettres grecques en forme droite. Par exemple les extensions fourier [15] mathdesign [16], ou kpfonts [17] disposent d'options permettant l'écriture automatique des lettres grecques minuscules en forme droite (et aussi des majuscules en italique) dans la police donnée. Citons également newpxmath, newtxmath et libertinust1math de Michael Sharpe ou pxgreeks, txgreeks et libgreek de Jean-François Burnol, qui fournissent les formes droites avec les polices Palatino, Times et Libertine respectivement.

Jean-François Burnol a également développé l'extension lgrmath [28] qui permet d'utiliser, en mode mathématique, les différentes fontes de lettres grecques accessibles par LaTeX avec l'encodage LGR. La documentation de l'extension indique comment consulter et utiliser les fontes accessibles sur votre distribution.

Enfin, comme pour les majuscules, l'extension unicode-math réalise cette tâche automatiquement avec l'option `math-style=french`, mais nécessite une compilation XeLaTeX ou LuaLaTeX.

Avant la version 3.0, frenchmath disposait de deux options pour composer les lettres grecques en forme droite : `lgrmath` et `upgreek`[16]. Inspiré par le travail de Jean-François Burnol, nous avons développé, et intégré à frenchmath, l'extension mathgreeks [30] qui permet d'utiliser de nombreuses fontes de lettres grecques sans modifier les autres caractères et symboles. Il suffit d'appeler frenchmath avec une des options de fonte de mathgreeks, de la forme `clé=valeur`. Par exemple, les lettres grecques droites de la police Utopia, telles que fournies par l'extension mathdesign, s'obtiennent avec

```
\usepackage[mathdesign=Utopia]{frenchmath}.
```

L'extension mathdesign elle-même ne sera pas chargée. Lorsqu'une valeur n'est pas précisée, il y a toujours une valeur par défaut. Par exemple pour `mathdesign`,

16. En fait il y avait une troisième option, `Upgreek`, mais celle-ci est désormais obsolète.

nous avons choisi `Charter`. Pour `upgreek`, la fonte par défaut est `Symbol`. Pour `lgrmath`, nous avons choisi `fcm`[17] qui se marie particulièrement bien avec la police usuelle Latin Modern.

Les commandes `\alpha`, `\beta`, ... `\pi`, etc. produisent alors les lettres en forme droite α, β, ..., π, etc. tandis que `\italpha`, `\itbeta`, ..., `\itpi`, etc. produisent les formes italiques α, β, ..., π, etc. Si ces dernières déplaisent, on peut activer l'option `savegreeks` (option booléenne passée à mathgreeks) pour conserver les lettres d'origine avec `\backalpha`, `\backbeta`, ..., `\backpi`, etc. : α, β, ..., π, etc.

Nous présentons ci-après quelques unes des nombreuses options possibles. On se référera à la documentation de mathgreeks [30] pour d'autres exemples ou pour les formes italiques correspondantes. La seconde ligne de chaque exemple contient, outre les majuscules, des variantes obtenues avec `\varpepsilon`, `\vartheta`, etc. (sans effet avec la clé `lgrmath`, excepté pour `\varsigma`). Les clés sans valeur sont en fait des booléens dont la valeur par défaut est `true`.

Option `clé=valeur`	Résultat
lgrmath=Cochineal-LF	α β γ δ ε ζ η θ ι κ λ μ ν ξ π ρ σ τ υ φ χ ψ ω ε θ π ρ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
lgrmath=fcm	α β γ δ ε ζ η θ ι κ λ μ ν ξ π ρ σ τ υ φ χ ψ ω ε θ π ρ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
upgreek=Euler	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
upgreek=Symbol	**α β γ δ ε ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω** **υ ϑ ϖ ρ σ φ** | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
mathdesign=Charter	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ ϱ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
fourier	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ ϱ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
pxfonts	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ ϱ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
txfonts	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ ϱ ς φ | Γ Δ Θ Λ Ξ Π Σ ϒ Φ Ψ Ω
libertinus	α β γ δ ϵ ζ η θ ι κ λ μ ν ξ π ρ σ τ υ ϕ χ ψ ω ε ϑ ϖ ϱ ς φ | Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω

17. Il faut que l'extension cm-lgc (développée par Alexej Kryukov) soit installée sur votre distribution, sans quoi la fonte de substitution LGR/lmr/m/n sera utilisée. Les formes italiques du thêta ϑ ou du rho ϱ de `fcm` ne sont pas vraiment ceux d'usage en mathématiques, mais les formes droites sont bien adaptées. `Alegreya-LF` est une autre fonte assez élégante pour `lgrmath`, mais produit un `\phi` ϕ moins courant et sans la variante φ.

Une option spéciale, `libsans` (booléen), permet d'obtenir une fonte sans empattement pour `libertinus`. Une autre option, non présentée dans le tableau, est **kpfonts**, avec deux valeurs possibles. Enfin deux autres options sont disponibles, mais pour une compilation avec LuaLaTeX (ou XeLaTeX) :

fontspec=... De nombreuses valeurs sont possibles, par exemple `FreeSerif`, `LiberationSerif`, `GFS Artemisia`, `GFS Didot` (valeur par défaut)... mais aussi des polices non libres comme `Cambria`, `Arial`, `Palatino Linotype`, `Times New Roman`... si elles sont installées sur votre système ; nécessite de charger l'extension fontspec [18].

unicode-math=... Donne un beau résultat avec `Latin Modern Math` (valeur par défaut) ou `STIX Two Math` ; nécessite le module unicode-math.

Par défaut LaTeX compose les majuscules grecques en forme droite et il n'est donc pas indispensable de recoder celles-ci. L'option `uppercase=false` permet de conserver les majuscules d'origine : Γ, Δ, ..., Ω (sans le préfixe `\back...`).

Mentionnons pour finir ce commentaire de Walter Schmidt [29] que le mu utilisé pour le préfixe des unités physiques, µ, doit se composer avec `\textmu` [19], disponible en mode texte dans beaucoup de fontes (ou avec textcomp) ; il diffère du μ droit mathématique, obtenu avec `$\mu$` si l'on a activé une des options de fonte précédentes.

3 Le code

Les seules options de frenchmath sont `capsit` et `noibrackets`. Tout le travail sur les lettre grecques est dorénavant géré par mathgreeks et toute option inconnue de frenchmath sera passée à mathgreeks. Auquel cas, l'on y ajoute automatiquement l'option `upright` pour composer les lettres grecques en forme droite. Si l'on donne une clé ou une valeur invalide, c'est alors mathgreeks qui produira un message d'erreur.

```
\newif\iffrenchmath@capsit
\DeclareOption{capsit}{\frenchmath@capsittrue}
\newif\iffrenchmath@noibrackets
\DeclareOption{noibrackets}{\frenchmath@noibracketstrue}
\newif\iffrenchmath@mathgreeks
\DeclareOption*{\frenchmath@mathgreekstrue
    \PassOptionsToPackage{\CurrentOption,upright}{mathgreeks}}
\ProcessOptions \relax

```

Assurer une bonne compatibilité avec unicode-math est un peu compliqué car unicode-math définit ou redéfinit de nombreux symboles, lettres, signes dans

18. L'extension fontspec est chargée automatiquement par unicode-math ou mathspec (cette dernière étant spécifique à XeLaTeX), mais l'option `fontspec` permet une variété de polices TrueType ou OpenType bien plus grande que l'option `unicode-math`.

19. L'extension textalpha fournit à la place `\textmicro` (depuis 2020) car elle redéfinit `\textmu`.

\AtBeginDocument. Les extensions decimalcomma et ibrackets font déjà appel à \AtBeginDocument mais pour garantir que leur action soit bien postérieure à celle de unicode-math leur appel a été placé dans un \AtEndPreamble, fourni par l'extension etoolbox. Il y a par contre un problème si on utilise une instruction comme \MakeShortVerb" du module shortvrb ou \lstMakeShortInline" du module listings. Ce type d'instruction doit agir *après* frenchmath, donc être placée après \begin{document} ou bien dans un \AtBeginDocument{...}.

L'extension mathptmx ne fournit pas de \jmath mais considère néanmoins cette commande comme existante. Dans ce cas, nous la rendons donc « indéfinie » pour la définir ensuite dans une commande générale faisant appel à l'extension dotlessj.

```
\RequirePackage{amsopn} % fournit \DeclareMathOperator
\RequirePackage{xspace} % utile pour les commandes \ssi, \Oij
\RequirePackage{etoolbox} % fournit \AtEndPreamble
\iffrenchmath@mathgreeks \RequirePackage{mathgreeks} \fi
\AtEndPreamble{
  \RequirePackage{decimalcomma}
  \iffrenchmath@noibrackets\else\RequirePackage{ibrackets}\fi
  \@ifpackageloaded{mathptmx}{\let\jmath\@undefined}{}
  \@ifundefined{jmath}{\RequirePackage{dotlessj}}{}
  \@ifpackageloaded{pstricks-add}{\psset{comma=true}}{}
}

```

\DeclareMathUp

Sauf si l'on a invoqué l'option capsup=false (ou capsit), on redéfinit toutes les lettres majuscules du mode mathématique grâce à la commande \DeclareMathUp. Contrairement aux bascules \MathUp et \MathIt de l'extension mismath, \DeclareMathUp ne fonctionne que dans le préambule, mais son code est bien plus simple et suffit à nos besoins ici. Pour balayer toutes les lettres majuscules de l'alphabet, les boucles usuelles \@for ou \foreach ne fonctionnent pas et produisent un message d'erreur « ! Improper alphabetic constant ». Ceci est dû au fait qu'une *séquence de contrôle* comme par exemple \@letter ne peut être utilisée comme argument de la macro \DeclareMathUp et ne sera pas gérée

\apply

de manière identique à un simple caractère. Mais la macro \apply ci-dessous va faire le travail [20]. \AtBeginDocument est nécessaire pour que ces définitions soient prises en compte avec la classe beamer par exemple, ou avec unicode-math.

```
\providecommand*\DeclareMathUp[1]{
    \DeclareMathSymbol{#1}{\mathalpha}{operators}{`#1}}

\def\apply#1#2{\apply@#1#2,\apply@,}
\def\apply@#1#2,{\ifx\apply@#2\empty
    \else #1{#2}\afterfi@{\apply@#1}\fi}
\def\afterfi@#1#2\fi{\fi#1}

\AtEndPreamble{\iffrenchmath@capsit\else\AtBeginDocument{
```

20. Cette puissante macro, postée le 26/02/2021 sous le pseudonyme de *wipet*, se trouve sur le forum de discussion iterate190.rssing.com, en réponse à « TeX How to iterate over a comma separated list? ». Que son auteur, Petr Olšák, soit remercié. Cette macro a d'ailleurs été utilisée de manière identique dans mismath, mais cela ne génère pas d'incompatibilité.

```
  \apply\DeclareMathUp{A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z}
  }\fi
}

```

En utilisant `\AtEndPreamble` (de l'extension etoolbox), avec `\AtBeginDocument` à l'intérieur, on garantit que certaines actions auront bien lieu après celles de unicode-math, même si frenchmath est chargé avant unicode-math.

Certains symboles, provenant de amssymb sont déjà définis par certaines extensions (mathdesign ou unicode-math) et ne seront pas redéfinis. Un avertissement sera affiché dans ce cas. Comme amssymb (qui utilise amsfonts) charge plusieurs fontes mathématiques et un grand nombre de symboles pas forcément nécessaires, nous n'avons extrait que les trois symboles dont nous avons besoin dans cette extension : `\leqslant`, `\geqslant` et `\varnothing`.

```
\AtEndPreamble{% au cas où frenchmath est chargé avant unicode-math
  \AtBeginDocument{% après actions de unicode-math
    \DeclareMathSymbol{;}{\mathbin}{operators}{'73} % et non \mathpunct
    \@ifundefined{mathscr}{\RequirePackage{mathrsfs}}{
        \PackageWarningNoLine{frenchmath}{Command \string\mathscr\space
            already defined, \MessageBreak
            I don't load mathrsfs}
    }
    \@ifundefined{leqslant}{
        \DeclareSymbolFont{AMSa}{U}{msa}{m}{n}% de amsfonts
        \DeclareMathSymbol{\leqslant}{\mathrel}{AMSa}{"36}% de amssymb
        \DeclareMathSymbol{\geqslant}{\mathrel}{AMSa}{"3E}
    }{
        \PackageWarningNoLine{frenchmath}{\string\leqslant\space
            and \string\geqslant\space already defined,
            \MessageBreak
            they will not be redefined}
    }
    \@ifundefined{varnothing}{% de amssymb
        \newcommand\varnothing{% \usefont évite de déclarer AMSb
            \mathord{\text{\usefont{U}{AMSb}{m}{n}\symbol{"3F}}}}
    }{
        \PackageWarningNoLine{frenchmath}{\string\varnothing\space
            already defined,
            \MessageBreak
            it will not be redefined,
            \MessageBreak
            except by using \string\AMSvarnothing}
    }
  }
}

```

Voyons les alias et quelques commandes particulières.

La commande `\varnothing` produit ? mais par contre, le symbole fournit par unicode-math, avec Latin Modern Math, est très laid et malheureusement identique

à \emptyset : ∅. Afin de récupérer le symbole de amssymb, sans être obligé de charger cette extension, nous proposons la commande \AMSvarnothing, à placer dans le préambule, après l'appel de unicode-math.

```
\newcommand\curs{\mathscr}
\newcommand\infeg{\leqslant}
\newcommand\supeg{\geqslant}
\newcommand\vide{\varnothing}
\newcommand\AMSvarnothing{% doit être appelé après \setmathfont ?
    \AtEndPreamble{\AtBeginDocument{% après actions de unicode-math
        \renewcommand\varnothing{%
            \mathord{\text{\usefont{U}{AMSb}{m}{n}\symbol{"3F}}}}
    }}
}
```

La définition de \paral 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.

La macro \cmod, copiée du \pod de amsmath (depuis v2.11), permet de gérer différemment l'espacement en mode « mathématiques centrées » (hors-ligne) ou en mode « en ligne ».

```
\newcommand\paral{\mathrel{\ooalign{$\mkern-1.75mu/\mkern1.75mu$\cr%
    $\mkern1.75mu/\mkern-1.75mu$}}}
\newcommand\ssi{si, et seulement si,\xspace}
%\newcommand*\cmod[1]{\quad[#1]}
\newif\if@display % provient de amsmath, peut être appelé 2 fois
\everydisplay\@xp{\the\everydisplay \@displaytrue}
\newcommand*\cmod[1]{\penalty \z@ % similaire à amsmath \allowbreak
    \if@display\mkern 18mu\else\mkern 8mu\fi [#1]}

```

Passons aux identifiants de fonctions classiques.

```
\DeclareMathOperator{\pgcd}{pgcd}
\DeclareMathOperator{\ppcm}{ppcm}
\DeclareMathOperator{\card}{card}
\DeclareMathOperator{\Card}{Card}
\DeclareMathOperator{\Ker}{Ker}
\DeclareMathOperator{\Hom}{Hom}
\DeclareMathOperator{\rg}{rg}
\DeclareMathOperator{\Vect}{Vect}
\DeclareMathOperator{\ch}{ch}
\DeclareMathOperator{\sh}{sh}
\AtBeginDocument{\let\oldth\th %\th existe déjà (mode texte)
    \renewcommand\th{\TextOrMath{\oldth}{\operatorname{th}}}}
\DeclareMathOperator{\cosec}{cosec}
\DeclareMathOperator{\cosech}{cosech}

```

Les commandes pour les bases et repères peuvent être utilisées en mode texte

grâce à \ensuremath (et \xspace qui garantit le bon espacement).

```
\newcommand\@Oij{%
    \ensuremath{\left(O, \vec{\imath}, \vec{\jmath}\,\right)}\xspace}
\newcommand\@@Oij{%
    \ensuremath{\left(O ; \vec{\imath}, \vec{\jmath}\,\right)}\xspace}
\newcommand\Oij{\@ifstar{\@@Oij}{\@Oij}}

\newcommand\@Oijk{%
    \ensuremath{%
        \left(O, \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
        \vec{\vphantom{t}\smash{k}}\,\right)}%
    \xspace}
\newcommand\@@Oijk{%
    \ensuremath{%
        \left(O ; \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
        \vec{\vphantom{t}\smash{k}}\,\right)}%
    \xspace}
\newcommand\Oijk{\@ifstar{\@@Oijk}{\@Oijk}}

\newcommand\@Ouv{%
    \ensuremath{\left(O, \vec{u}, \vec{v}\,\right)}\xspace}
\newcommand\@@Ouv{%
    \ensuremath{\left(O ; \vec{u}, \vec{v}\,\right)}\xspace}
\newcommand\Ouv{\@ifstar{\@@Ouv}{\@Ouv}}

\AtBeginDocument{\let\oldij\ij %\ij existe déjà (mode texte)
    \renewcommand\ij{\TextOrMath{\oldij}{%
        \left(\vec{\imath}, \vec{\jmath}\,\right)}}
}
\newcommand\ijk{%
    \ensuremath{%
        \left(\vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
        \vec{\vphantom{t}\smash{k}}\,\right)}%
    \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
https://euler.ac-versailles.fr/IMG/pdf/typo2.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] *isomath – Mathematical style for science and technology*, Günter Milde, CTAN, v0.6.1 2012/09/04.

[9] *PM-ISOmath – The Poor Man ISO math bundle*, Claudio Beccari, CTAN, v1.2.00 2021/08/04.

[10] *La distribution mafr*, Christian Obrecht, CTAN, v1.0 17/09/2006.

[11] *A Babel language definition file for French*, extension LaTeX babel-french de Daniel Flipo, CTAN, v3.5c 14/09/2018.

[12] *L'extension tdsfrmath*, Yvon Henel, CTAN, v1.3 22/06/2009.

[13] *L'extension tablvar*, Antoine Missier, CTAN, v2.0 23/12/2023.

[14] *Experimental Unicode mathematical typesetting : The unicode-math package*, Will Robertson, Philipp Stephani, Joseph Wright, Khaled Hosny, and others, CTAN, v0.8r 13/08/2023.

[15] Fourier-GUT*enberg*, Michel Bovani, CTAN, v1.3 30/01/2005.

[16] *The mathdesign package*, Paul Pichaureau, CTAN, v2.31 29/08/2013.

[17] *Kp-Fonts – The Johannes Kepler project*, Christophe Caignaert, CTAN, v3.34 20/09/2022.

[18] *Miscellaneous mathematical macros – The mismath package*, Antoine Missier, CTAN, v3.0 04/05/2024.

[19] *esvect – Typesetting vectors with beautiful arrow with LaTeX 2ε*, Eddie Saudrais, CTAN, v1.3 11/07/2013.

[20] *The icomma package for LaTeX 2ε*. Walter Schmidt, CTAN, v2.0 10/03/2002.

[21] *The ncccomma package*. Alexander I. Rozhenko, CTAN, v1.0 10/02/2005.

[22] *Intelligent commas*. Claudio Beccari, The PracTeX Journal, 2011, No.1.
https://tug.org/pracjourn/2011-1/beccari/Intcomma.pdf

[23] *The decimalcomma package*. Antoine Missier, CTAN, v1.4 30/12/2023.

[24] *Intelligent brackets – The ibrackets package*, Antoine Missier, CTAN, v1.2 26/07/2023.

[25] *The mathtools package*, Morten Høgholm, Lars Madsen and the LaTeX3 project, CTAN, v1.29 29/06/2022.

[26] *The* mathalpha, *AKA* mathalfa *package*, Michael Sharpe, CTAN, v1.143 18/11/2021.

[27] *dotlessj*, David Carlisle, CTAN, v0.03 09/12/1998.

[28] *The* lgrmath *package*, Jean-François B., CTAN, v1.0 16/11/2022.

[29] *The* upgreek *package for LaTeX 2ε*, Walter Schmidt, CTAN, v2.0 12/03/2003.

[30] *The* mathgreeks *package*, Antoine Missier, CTAN, v1.1 04/05/2024.

[31] *The* textalpha *package* (partie de l'extension greek-fontenc), Günter Milde, CTAN, v2.1 14/06/2022.

[32] *LaTeX Companion*, Frank Mittelbach, Michel Goossens, 2e édition, Pearson Education France, 2005.