L'extension frenchmath*

Antoine Missier
antoine.missier@ac-toulouse.fr

20 février 2024

1 Introduction

Cette extension, inspirée de mafr de Christian Obrecht [10], permet le respect des règles typographiques des mathématiques françaises, en particulier elle permet d'obtenir automatiquement les 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. Mais contrairement à mafr nous avons choisi de ne pas conserver le même nom de commande pour substituer des symboles français aux symboles anglais.

D'autres solutions existent pour composer les majuscules mathématiques en romain, par exemple l'extension unicode-math [11] mais qui doit être compilée avec XeLaTeX ou LuaLaTeX (et doit être chargée *avant* frenchmath) ; pour pdfLaTeX nous avons 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.

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 [1], 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` [2], 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 [16] qui fournit également beaucoup de commandes francisées ou encore tablvar [17] qui permet de réaliser de jolis tableaux de variations, spécificité des mathématiques françaises.

Depuis la version 2.0, frenchmath propose deux options permettant de composer les minuscules grecques du mode mathématique en forme droite.

*Ce document correspond à frenchmath v2.9, dernière modification le 20/02/2024.

1. Nous proposons pour cela l'extension mismath [14] qui fournit diverses macros pour les mathématiques internationales.

2. Pour de jolis vecteurs on dispose de l'extension esvect [15] d'Eddie Saudrais.

2 Utilisation

2.1 Majuscules mathématiques

Dans les mathématiques françaises, pour l'alphabet latin, « les lettres majuscules sont toujours composées en romain » (A, B, C...) et non en italique (cf. [1] p.107, voir aussi [2]). En utilisant XeLaTeX ou LuaLaTeX avec des polices mathématiques OpenType, cette convention est assez commode à mettre en œuvre [3] ; par contre, avec LaTeX ou pdfLaTeX, elle est peu respectée et les extensions précitées ne fonctionnent qu'avec des polices particulières. Par défaut frenchmath compose automatiquement les majuscules mathématiques latines en romain, quelle que soit la fonte 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 [14] 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.

2.2 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` [18], 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 [19], 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 [20], la virgule se comporte comme un caractère ordinaire si elle est suivie d'un chiffre (sans espace), sinon c'est un caractère de ponctuation.

3. Voir les options `math-style=upright` ou `math-style=french` de l'extension unicode-math.
4. Dans ce cas il ne faut pas utiliser les bascules, au risque de rendre ces extensions inopérantes.

Cette deuxième approche parait meilleure, néanmoins ncccomma ne fonctionne pas avec 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* [21], Claudio Beccari propose une autre solution, voisine de ncccomma, mais qui produit le même type d'incompatibilités. Le code a donc été revu et simplifié afin de régler proprement ces incompatibilités. Comme bien des pays utilisent la virgule comme séparateur décimal, il fait l'objet d'une extension séparée, decimalcomma [22], qui est chargée par frenchmath (depuis la version 2.7). Mais si on utilise unicode-math, il est alors impératif de le charger *avant* frenchmath[6].

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`

`[noibrackets]`

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 [23] qui est chargée par frenchmath, sauf si l'on active 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, `$[a,b] + [c,d]$` 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`.

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.

7. D'autres solutions existent, par exemple avec l'extension interval ou encore avec la macro `\DeclarePairedDelimiter` de mathtools [24], mais utilisée avec des crochets, cette dernière est incompatible avec ibrackets, d'où la possibilité de désactiver ibrackets.

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 [13] ou l'option `scr` de mathalpha [25][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, chargée par frenchmath.

`\vide` Le symbole de l'ensemble vide $\varnothing$ 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 : $\emptyset$.

`\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, ».

`\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 : $53 \equiv 5 \quad [12]$.

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 [31]), sont aussi utilisées en français [1]. On les obtient avec les commandes `\ch`, `\sh` et `\th`[13].

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).
11. Cette dernière n'est pas implémentée en standard dans LaTeX (mais dans mismath [14]).
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.

\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). On peut écrire ces commandes en mode texte, sans 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 [26] de David Carlisle, ce qui permet aux macros ci-dessus de fonctionner normalement.

2.6 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 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 [12] [13] [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 mathdesign [13], fourier [12] ou kpfonts [27] disposent d'options permettant l'écriture automatique des lettres grecques minuscules en forme droite (ou des majuscules en italique). Citons éga-

14. La fonction sécante est définie en standard par LaTeX avec `\sec` et la sécante hyperbolique `\sech` est définie par mismath [14].

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

lement newpxmath, newtxmath [16] et libertinust1math de Michael Sharpe, pxgreeks, txgreeks [17] et libgreek de Jean-François Burnol, qui donnent de beaux résultats pour une utilisation avec respectivement les polices Palatino, Times et Libertinus.

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 avec XeLaTeX ou LuaLaTeX.

L'extension frenchmath fournit les options décrites ci-dessous afin d'obtenir les lettres grecques en forme droite. Ces options peuvent être utilisées y compris avec unicode-math, mais seulement en compilant avec XeLaTeX et non pas LuaLaTeX (et unicode-math doit être chargé *avant* frenchmath).

`[lgrmath]` En activant l'option `lgrmath`, frenchmath charge l'extension du même nom avec son option `style=french` et la fonte fcm (de l'extension cm-lgc) [18]. Celle-ci se marie particulièrement bien avec la police usuelle Latin Modern. Les commandes `\alpha`, `\beta`, etc. produisent alors les lettres en forme droite α, β, …, π, etc., tandis que `\alphait`, `\betait`, etc. produisent des formes italiques a, β, …, π, etc. Ces dernières sont peu à notre goût, mais on conserve les lettres d'origine avec `\italpha`, `\itbeta` : α, β, …, π, etc.

`\SaveGreekItalics` On peut aussi choisir d'autres fontes en chargeant l'extension lgrmath indépendamment de frenchmath et sans son option `lgrmath` (voir par exemple les options `font=Alegreya-LF` ou `font=Cochineal-LF` de lgrmath). Dans ce cas, si on veut conserver les lettres italiques d'origine, il faut précéder le chargement de lgrmath de la commande `\SaveGreekItalics` ; et si on utilise unicode-math, il faut en outre placer cette commande et le `\usepackage[...]{lgrmath}` dans un `\AtBeginDocument`.

`[upgreek]` Avec la même philosophie, frenchmath fournit aussi l'option `upgreek` basée sur l'extension upgreek de Walter Schmidt [29] qui donne accès à d'autres fontes de lettres grecques minuscules en forme droite avec `\upalpha`, `\upbeta`, etc. L'extension upgreek sera chargée avec son option `Symbol` [19] utilisant la police Adobe Symbol et qui produit des lettres grecques assez grasses : $\upalpha$, $\upbeta$, …, $\uppi$, etc.

Si l'on veut, par contre, utiliser l'extension upgreek avec l'une des deux autres options disponibles, `Euler` (qui produit $\upalpha, \upbeta, \ldots, \uppi$, etc.) ou `Symbolsmallscale`, il faut charger l'extension upgreek avec l'option souhaitée indépendamment de

16. L'extension newtxmath doit être chargée après frenchmath qui utilise amssymb car la compilation produit sinon un message d'erreur ; ou sinon on peut ajouter `\let\Bbbk\relax` avant de charger frenchmath.

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

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

19. L'option `Symbol` de upgreek se marie bien avec une police comme Times par exemple.

frenchmath. Mais en conservant toutefois l'option upgreek, frenchmath redéfinira les commandes \alpha, \beta, etc. pour composer automatiquement les lettres en forme droite (comme alias de \upalpha, \upbeta...).

\upgreekUndefined

Comme pour lgrmath, si on veut que les formes italiques d'origine restent toujours disponibles avec les commandes \italpha, \itbeta, ..., \itpi, etc., il faut précéder le chargement de l'extension upgreek de la commande \SaveGreekItalics ; et si l'on utilise unicode-math, il faut en outre ajouter la commande \upgreekUndefined avant le \usepackage[...]{upgreek} et les trois dans un \AtBeginDocument placé entre le chargement de unicode-math et celui de frenchmath [20].

[Upgreek]

Avec LaTeX, 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 : $\Gamma, \ldots, \Omega$. Afin de conserver majuscules et minuscules dans le même style, frenchmath fournit l'option Upgreek qui redéfinit les majuscules \Gamma, ..., \Omega pour correspondre à ces variantes. Par contre l'on n'a alors plus accès aux caractères d'origine : $\Gamma, \ldots, \Omega$. L'option Upgreek reprend aussi les minuscules grecques de l'option upgreek, qu'il est donc inutile d'invoquer simultanément.

Signalons aussi l'extension textalpha de Günter Milde [30] qui donne accès aux lettres en forme droite α, β, ..., π, etc., mais en mode texte avec \textalpha, \textbeta, etc. Ces glyphes, voisines de la fonte fcm accessible avec lgrmath, se marient également bien avec la police Latin Modern, par contre le thêta produit, ϑ, n'est pas vraiment celui qui est d'usage en mathématiques.

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 [21], disponible en mode texte dans beaucoup de fontes (ou avec textcomp) ; il diffère du μ droit mathématique.

3 Le code

```
\newif\ifcapsit
\DeclareOption{capsit}{\capsittrue}
\newif\iflgrmath
\DeclareOption{lgrmath}{\lgrmathtrue}
\newif\ifupgreek
\DeclareOption{upgreek}{\upgreektrue}
\newif\ifUpgreek
\DeclareOption{Upgreek}{\Upgreektrue\upgreektrue}
\newif\ifnoibrackets
\DeclareOption{noibrackets}{\noibracketstrue}
\ProcessOptions \relax
```

20. Tout cela est bien contraignant mais c'est lié au fonctionnement capricieux de unicode-math qui génère de nombreuses incompatibilités. Ceci dit, on n'a en principe besoin de nos options de lettres grecques droites que si l'on n'est pas satisfait de celles fournies par unicode-math.

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

```
\@ifpackageloaded{unicode-math}{
    \@ifpackageloaded{mathdesign}{
        \PackageInfo{frenchmath}{Package mathdesign
            is loaded, \MessageBreak
            I don't load mathrsfs and amssymb}
    }{
        %\let\circledS\relax % utilisé uniquement par mathdesign
        \let\eth\relax   % 3 incompatibilités unicode-math / amssymb
        \let\digamma\relax
        \let\backepsilon\relax
        \RequirePackage{amssymb}
        \PackageInfo{frenchmath}{Package unicode-math
            is loaded, \MessageBreak
            I don't load mathrsfs}
    }
    \let\vide\varnothing % \varnothing sera écrasé par unicode-math
}{
  \AtBeginDocument{
    \@ifpackageloaded{mathdesign}{
        \PackageInfo{frenchmath}{Package mathdesign
            is loaded, \MessageBreak
            I don't load mathrsfs and amssymb}
    }{
        \RequirePackage{amssymb} % \leqslant, \geqslant, \varnothing
        \@ifundefined{mathscr}{\RequirePackage{mathrsfs}}{
            \PackageInfo{frenchmath}{Command \string\mathscr\space
                already defined, \MessageBreak
                I don't load mathrsfs}
        }
    }
  }
}
\RequirePackage{amsopn} % fournit \DeclareMathOperator
\@ifpackageloaded{mathptmx}{\RequirePackage{dotlessj}}{}
\RequirePackage{xspace} % utile pour les commandes \ssi, \Oij
\ifnoibrackets\else\RequirePackage{ibrackets}\fi % intelligent brackets
\RequirePackage{decimalcomma} %  depuis frenchmath 2.7

```

`\DeclareMathUp` Sauf si l'option `capsit` est activée, 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 de manière identique à un simple ca-

\apply ractère. Mais la macro \apply ci-dessous va faire le travail [22]. \AtBeginDocument est nécessaire pour que ces définitions soient prises en compte avec la classe beamer par exemple, ou avec unicode-math.

```
\newcommand*\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}

\ifcapsit\else\AtBeginDocument{
  \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

\AtBeginDocument{\@ifpackageloaded{pstricks-add}{\psset{comma=true}}{}}
\DeclareMathSymbol{;}{\mathbin}{operators}{'73} % \mathpunct à l'origine

```

Passons aux alias et identifiants de fonctions classiques.

```
\newcommand\curs{\mathscr}
\newcommand\infeg{\leqslant}
\newcommand\supeg{\geqslant}
\@ifundefined{vide}{\newcommand\vide{\varnothing}}{}
```

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.

```
\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]}

\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}
```

22. 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 été définie de manière identique dans mismath, mais cela ne génère pas d'incompatibilité.

```
\AtBeginDocument{\let\oldth\th %\th existe déjà (mode texte)
    \renewcommand\th{\TextOrMath{\oldth}{\operatorname{th}}}}
\DeclareMathOperator{\cosec}{cosec}
\DeclareMathOperator{\cosech}{cosech}

```

Présentons les commandes pour les bases et repères, qui 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{
    \renewcommand\ij{%
        \ensuremath{\left(\vec{\imath}, \vec{\jmath}\,\right)}\xspace}}
\newcommand\ijk{%
    \ensuremath{%
        \left(\vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
        \vec{\vphantom{t}\smash{k}}\,\right)}%
    \xspace}

```

Ce n'est qu'à la fin du préambule, donc avec `\AtBeginDocument`, que l'on examine les options `lgrmath` ou `upgreek`, pour que cela fonctionne avec unicode-math et pour laisser à l'utilisateur la possibilité de charger upgreek après frenchmath et éviter un conflit d'option. On sauvegarde les lettres grecques d'origine dans les macros `\italpha`, …, `\itomega`, grâce à la commande `\SavedGreekItalics`. La macro `\upgreekUndefined` ne sert que si l'on veut utiliser upgreek avec unicode-math : elle permet de « vider » la définition de `\upalpha`, `\upbeta`…, car sinon leur redéfinition par upgreek est impossible.

```
\newcommand\SaveGreekItalics{
    \let\italpha\alpha
    \let\itbeta\beta
    \let\itgamma\gamma
    \let\itdelta\delta
    \let\itepsilon\epsilon
    \let\itzeta\zeta
    \let\iteta\eta
    \let\ittheta\theta
    \let\itiota\iota
    \let\itkappa\kappa
    \let\itlambda\lambda
    \let\itmu\mu
    \let\itnu\nu
    \let\itxi\xi
    \let\itpi\pi
    \let\itrho\rho
    \let\itsigma\sigma
    \let\ittau\tau
    \let\itupsilon\upsilon
    \let\itphi\phi
    \let\itchi\chi
    \let\itpsi\psi
    \let\itomega\omega
    \let\itvarepsilon\varepsilon
    \let\itvartheta\vartheta
    \let\itvarpi\varpi
    \let\itvarsigma\varsigma
    \let\itvarphi\varphi
}

\newcommand\upgreekUndefined{
    \let\upalpha\@undefined
    \let\upbeta\@undefined
    \let\upgamma\@undefined
    \let\updelta\@undefined
    \let\upepsilon\@undefined
    \let\upzeta\@undefined
    \let\upeta\@undefined
    \let\uptheta\@undefined
    \let\upiota\@undefined
    \let\upkappa\@undefined
    \let\uplambda\@undefined
    \let\upmu\@undefined
    \let\upnu\@undefined
    \let\upxi\@undefined
    \let\uppi\@undefined
    \let\uprho\@undefined
    \let\upsigma\@undefined
    \let\uptau\@undefined
```

```
    \let\upupsilon\@undefined
    \let\upphi\@undefined
    \let\upchi\@undefined
    \let\uppsi\@undefined
    \let\upomega\@undefined
    \let\upvarepsilon\@undefined
    \let\upvartheta\@undefined
    \let\upvarpi\@undefined
    \let\upvarrho\@undefined
    \let\upvarsigma\@undefined
    \let\upvarphi\@undefined
}

\AtBeginDocument{
  \iflgrmath
    \SaveGreekItalics
    \RequirePackage[font=fcm,style=french]{lgrmath}
  \fi
  \ifupgreek
    \@ifpackageloaded{upgreek}{}{
      \@ifpackageloaded{unicode-math}{\upgreekUndefined}{}
      \SaveGreekItalics
      \RequirePackage[Symbol]{upgreek}
    }
    \renewcommand\alpha{\upalpha}
    \renewcommand\beta{\upbeta}
    \renewcommand\gamma{\upgamma}
    \renewcommand\delta{\updelta}
    \renewcommand\epsilon{\upepsilon}
    \renewcommand\zeta{\upzeta}
    \renewcommand\eta{\upeta}
    \renewcommand\theta{\uptheta}
    \renewcommand\iota{\upiota}
    \renewcommand\kappa{\upkappa}
    \renewcommand\lambda{\uplambda}
    \renewcommand\mu{\upmu}
    \renewcommand\nu{\upnu}
    \renewcommand\xi{\upxi}
    \renewcommand\pi{\uppi}
    \renewcommand\rho{\uprho}
    \renewcommand\sigma{\upsigma}
    \renewcommand\tau{\uptau}
    \renewcommand\upsilon{\upupsilon}
    \renewcommand\phi{\upphi}
    \renewcommand\chi{\upchi}
    \renewcommand\psi{\uppsi}
    \renewcommand\omega{\upomega}
    \renewcommand\varepsilon{\upvarepsilon}
    \renewcommand\vartheta{\upvartheta}
    \renewcommand\varpi{\upvarpi}
```

```
    \renewcommand\varrho{\upvarrho}
    \renewcommand\varsigma{\upvarsigma}
    \renewcommand\varphi{\upvarphi}
  \fi
  \ifUpgreek
    % unicode-math utilise \upGamma, \upDelta...
    \renewcommand\Gamma{\Upgamma}
    \renewcommand\Delta{\Updelta}
    \renewcommand\Theta{\Uptheta}
    \renewcommand\Lambda{\Uplambda}
    \renewcommand\Xi{\Upxi}
    \renewcommand\Pi{\Uppi}
    \renewcommand\Sigma{\Upsigma}
    \renewcommand\Upsilon{\Upupsilon}
    \renewcommand\Phi{\Upphi}
    \renewcommand\Psi{\Uppsi}
    \renewcommand\Omega{\Upomega}
  \fi
}
```

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

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

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

[14] *mismath – Miscellaneous mathematical macros*, Antoine Missier, CTAN, v2.10 20/02/2024.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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