

L'extension frenchmath*

Antoine Missier
antoine.missier@ac-toulouse.fr

2 novembre 2020

1 Introduction

Cette extension, inspirée de `mafr` de Christian Obrecht [10], permet le respect des règles typographiques mathématiques françaises, en particulier la possibilité d'obtenir automatiquement les majuscules en romain (lettres droites) plutôt qu'en italique (voir [1] et [2]) ainsi que des espacements corrects pour les virgules et point-virgules.

D'autres solutions pour composer les majuscules mathématiques en romain sont proposées dans les extensions `fourier` [12] (avec la famille des polices Adobe Utopia) ou encore `mathdesign` [13] (avec les polices Adobe Utopia, URW Garamond ou Bitstream Charter). Mais `frenchmath` fournit une méthode générique s'adaptant à n'importe quelle police, en particulier Latin Modern (extension `lmodern`) avec laquelle ce document a été composé.

D'autres préconisations, telles que composer en lettre droite et non en italique le symbole différentiel, les nombres i et e [2], sont des règles internationales [6] [7] [9]. 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 anglo-saxons avec le même nom de commande mais de créer de nouvelles commandes ;
- les macros présentées dans la section 2 qui correspondent 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 \mathbf{R} , \mathbf{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 beaucoup de commandes francisées.

*Ce document correspond à `frenchmath` v1.5, dernière modification le 02/11/2020.

1. Nous proposons pour cela l'extension `mismath` [19] qui fournit diverses macros pour les mathématiques internationales.

2. Pour de jolis vecteurs on dispose de l'extension `esvect` d'Eddie Soudrais [18].

2 Utilisation

2.1 Majuscules mathématiques

En France, les lettres majuscules du mode mathématique doivent toujours être composées en romain (A, B, C, ...) et non en italique ([1] p.107, voir aussi [2]). Il faut dire que cette convention n'est pas commode à mettre en œuvre, ni avec L^AT_EX, ni avec les éditeurs de formule des traitements de textes usuels, et peu d'auteurs la respectent. La mise en œuvre automatique de cette recommandation est le principal bénéfice de `frenchmath` (comme de `mafr`).

`capsup`, `capsit`

L'extension `frenchmath` possède deux options : `capsup` (par défaut) et `capsit`. Avec `capsit`, les majuscules du mode mathématique sont composées automatiquement en italique et avec `capsup` en forme droite (dans la famille de fonte par défaut, généralement romain). Quelque soit l'option choisie, il est toujours possible de changer l'aspect d'une lettre particulière, avec les macros L^AT_EX `\mathrm` et `\mathit`. Par défaut `\[P(X)=\sum_{i=0}^n a_i X^i \]` donne

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

2.2 Virgules et point-virgule

Dans le mode mathématique de L^AT_EX, la virgule est toujours, par défaut, un symbole de ponctuation qui 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` [14], 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 bascules³. En mode mathématique :

- avec `icomma` (intelligent comma) de Walter Schmidt [15], 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 [16], 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'extension `numprint` lorsque celle-ci est chargée avec l'option `autolanguage`⁴. Par contre `icomma` fonctionne mais à condition d'être chargé postérieurement. Vu son intérêt cette extension est automatiquement chargée par `frenchmath`. Il faudra donc prendre garde à appeler, dans le préambule, `\usepackage{frenchmath}` après un éventuel `\usepackage[autolanguage]{numprint}`.

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

4. L'option `autolanguage` de `numprint` utilisé 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 grand que l'espace que l'on obtient sans cette option.

Mentionnons enfin l'article *Intelligent commas* de Claudio Beccari [17] qui propose une solution simplifiée par rapport à `ncccomma` mais qui ne fonctionne pas mieux.

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

Le symbole « ; » a été redéfini pour le mode mathématique car l'espace précédant le point-virgule est incorrecte en français `$x \in [0,25 ; 3,75]$` donne $x \in [0,25;3,75]$ sans `frenchmath` et $x \in [0,25 ; 3,75]$ avec `frenchmath`; le comportement de « ; » devient identique à celui de « : »⁵.

2.3 Quelques macros et alias utiles

- `\curs` Les lettres cursives ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \dots$) sont composées avec la macro `\curs` et sont différentes de celles obtenues avec `\mathcal`⁶ ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \dots$). L'activation du mode mathématique n'est pas nécessaire.*⁷
- `\infeg` Les relations \leq et \geq s'obtiennent avec les commandes `\infeg` et `\supeg` et diffèrent des versions anglaises de `\leq` (\leq) et `\geq` (\geq). Ce sont des alias des commandes `\leqslant` et `\geqslant` de l'extension `amssymb` chargée par `frenchmath`.*
- `\vide` Le symbole \emptyset ⁸ s'obtient avec `\vide` (alias de `\varnothing` de l'extension `amssymb`); il diffère de la version anglaise obtenue avec `\emptyset` : \emptyset .*
- `\paral` La commande `\paral` fournit la *relation*⁹ du parallélisme : \parallel , plutôt que sa version anglaise `\parallel` : \parallel .*
- `\ssi` La commande `\ssi` produit « si, et seulement si, ».
- `\cmod` Bien que \LaTeX propose par défaut le modulo entre parenthèses, avec `\pmod`, qui est d'usage en français, on peut vouloir composer un modulo entre crochets, ce que permet la commande `\cmod` en respectant le bon espacement propre au modulo : $5 \equiv 53$ [12].

2.4 Identifiants de « fonctions » classiques

- `\pgcd` En arithmétique, nous avons les classiques `\pgcd` et `\ppcm`, qui diffèrent de leur version anglo-saxonne `\gcd` et `\lcm`¹⁰.

5. Un autre problème d'espacement, non spécifique au français, se pose avec les délimiteurs [et], par exemple $x \in]0, \pi[$. Une solution est proposée dans l'extension `mismath`.

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

7. Comme dit dans l'introduction, l'astérisque en fin d'item signale une fonctionnalité similaire dans `mafr`.

8. \LaTeX fournit la commande `\o` qui compose également un O barré, mais trop décalé vers le bas (pour l'ensemble vide) : $S = \emptyset$, alors qu'avec `\vide` on obtient $S = \emptyset$.

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

10. Cette dernière n'est pas implémentée en standard dans \LaTeX (mais dans `mismath`).

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

`\Ker` `\Hom` L^AT_EX fournit les macros `\ker` et `\hom` alors que l'usage français est souvent de commencer ces noms par une majuscule pour obtenir `Ker`¹¹ 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` `\tgh` En principe, les fonctions hyperboliques s'écrivent en français avec les macros L^AT_EX standard `\cosh`, `\sinh`, `\tanh`; les écritures `ch x`, `sh x` et `th x` ne sont la norme qu'avec les langues d'Europe de l'Est [8], néanmoins ces écritures sont aussi utilisées en France [1]. On les obtient avec les commandes `\ch`, `\sh` et `\tgh`¹².

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 (O, \vec{i}, \vec{j}) , `\Oijk` compose $(O, \vec{i}, \vec{j}, \vec{k})$ et `\Ouv` compose (O, \vec{u}, \vec{v}) (utilisé dans le plan complexe). On peut écrire ces commandes en mode texte, sans les délimiteurs du mode mathématique.

`\Oij*` `\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 $(O ; \vec{i}, \vec{j})$, $(O ; \vec{i}, \vec{j}, \vec{k})$, $(O ; \vec{u}, \vec{v})$.

`\ij` `\ijk` Enfin les macros `\ij` et `\ijk` composent la base du plan (\vec{i}, \vec{j}) et de l'espace $(\vec{i}, \vec{j}, \vec{k})$, en homogénéisant la hauteur des flèches. 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 \DeclareOption{capsup}{\setboolean{capsit}{false}} % valeur par défaut
5 \ProcessOptions \relax
6
7 \RequirePackage{mathrsfs} % fournit les majuscules cursives
8 \RequirePackage{amssymb} % fournit \leqslant, \geqslant et \varnothing
9 \RequirePackage{amsopn} % fournit \DeclareMathOperator
10 \RequirePackage{xspace} % utile pour les commandes \curs, \ssi, \Oij
11 \RequirePackage{icomma} % virgule intelligente
12
13 \DeclareSymbolFont{UpSh}{\encodingdefault}{\familydefault}{m}{n}

```

L'option `capsup` redéfinit toutes les lettres majuscules du mode mathématique; `\AtBeginDocument` est nécessaire pour que ces définitions soient prises en compte avec l'extension `beamer`.

11. 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.

12. La commande `\th` existe déjà et produit \mathfrak{b} .

```

14 \ifthenelse{\boolean{capsit}}{\AtBeginDocument{
15     \DeclareMathSymbol{A}\mathalpha{UpSh}{‘A} %’A codage octal du A
16     \DeclareMathSymbol{B}\mathalpha{UpSh}{‘B}
17     \DeclareMathSymbol{C}\mathalpha{UpSh}{‘C}
18     \DeclareMathSymbol{D}\mathalpha{UpSh}{‘D}
19     \DeclareMathSymbol{E}\mathalpha{UpSh}{‘E}
20     \DeclareMathSymbol{F}\mathalpha{UpSh}{‘F}
21     \DeclareMathSymbol{G}\mathalpha{UpSh}{‘G}
22     \DeclareMathSymbol{H}\mathalpha{UpSh}{‘H}
23     \DeclareMathSymbol{I}\mathalpha{UpSh}{‘I}
24     \DeclareMathSymbol{J}\mathalpha{UpSh}{‘J}
25     \DeclareMathSymbol{K}\mathalpha{UpSh}{‘K}
26     \DeclareMathSymbol{L}\mathalpha{UpSh}{‘L}
27     \DeclareMathSymbol{M}\mathalpha{UpSh}{‘M}
28     \DeclareMathSymbol{N}\mathalpha{UpSh}{‘N}
29     \DeclareMathSymbol{O}\mathalpha{UpSh}{‘O}
30     \DeclareMathSymbol{P}\mathalpha{UpSh}{‘P}
31     \DeclareMathSymbol{Q}\mathalpha{UpSh}{‘Q}
32     \DeclareMathSymbol{R}\mathalpha{UpSh}{‘R}
33     \DeclareMathSymbol{S}\mathalpha{UpSh}{‘S}
34     \DeclareMathSymbol{T}\mathalpha{UpSh}{‘T}
35     \DeclareMathSymbol{U}\mathalpha{UpSh}{‘U}
36     \DeclareMathSymbol{V}\mathalpha{UpSh}{‘V}
37     \DeclareMathSymbol{W}\mathalpha{UpSh}{‘W}
38     \DeclareMathSymbol{X}\mathalpha{UpSh}{‘X}
39     \DeclareMathSymbol{Y}\mathalpha{UpSh}{‘Y}
40     \DeclareMathSymbol{Z}\mathalpha{UpSh}{‘Z}
41 }}
42 \AtBeginDocument{\@ifpackageloaded{pstricks-add}{\psset{comma=true}}{}}
43 \DeclareMathSymbol{;}\mathbin{UpSh}{059} % \mathpunct à l’origine
44
45 \newcommand*\curs[1]{\ensuremath{\mathscr{#1}}\xspace}
46 \newcommand*\infeg{\leqslant}
47 \newcommand*\supeg{\geqslant}
48 \newcommand*\vide{\varnothing}
49 \newcommand*\paral{\mathrel{/\!/}} % \parallel existe déjà : ||
50 \newcommand*\ssi{si, et seulement si,\xspace}
51 \newcommand*\cmod[1]{\quad[#1]}
52
53 \DeclareMathOperator{\pgcd}{pgcd}
54 \DeclareMathOperator{\ppcm}{ppcm}
55 \DeclareMathOperator{\card}{card}
56 \DeclareMathOperator{\Card}{Card}
57 \DeclareMathOperator{\Ker}{Ker}
58 \DeclareMathOperator{\Hom}{Hom}
59 \DeclareMathOperator{\rg}{rg}
60 \DeclareMathOperator{\Vect}{Vect}
61 \DeclareMathOperator{\ch}{ch}
62 \DeclareMathOperator{\sh}{sh}
63 \DeclareMathOperator{\tgh}{th} %\th existe déjà

```

```

64
65 \newcommand\@Oij{\ensuremath{
66   \left(0, \vec{\imath}, \vec{\jmath}\backslash, \right)
67   }\xspace
68 }
69 \newcommand\@@Oij{\ensuremath{
70   \left(0 ; \vec{\imath}, \vec{\jmath}\backslash, \right)
71   }\xspace
72 }
73 \newcommand\Oij{\@ifstar{\@Oij}{\@Oij}}
74
75 \newcommand\@Oijk{\ensuremath{
76   \left(0, \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
77   \vec{\vphantom{t}\smash{k}}\backslash, \right)
78   }\xspace
79 }
80 \newcommand\@@Oijk{\ensuremath{
81   \left(0 ; \vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
82   \vec{\vphantom{t}\smash{k}}\backslash, \right)
83   }\xspace
84 }
85 \newcommand\Oijk{\@ifstar{\@Oijk}{\@Oijk}}
86
87 \newcommand\@Ouv{\ensuremath{
88   \left(0, \vec{u}, \vec{v}\backslash, \right)}\xspace
89 }
90 \newcommand\@@Ouv{\ensuremath{
91   \left(0 ; \vec{u}, \vec{v}\backslash, \right)}\xspace
92 }
93 \newcommand\Ouv{\@ifstar{\@Ouv}{\@Ouv}}
94
95 \AtBeginDocument{\renewcommand\ij{\ensuremath{
96   \left(\vec{\imath}, \vec{\jmath}\backslash, \right)
97   }\xspace
98 }}
99 \newcommand\ijk{\ensuremath{
100   \left(\vec{\vphantom{t}\imath}, \vec{\vphantom{t}\jmath},
101   \vec{\vphantom{t}\smash{k}}\backslash, \right)
102   }\xspace
103 }
104

```

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.

- [3] *Règles françaises de typographie mathématique*. Alexandre André, 02/09/2015.
- [4] *Le petit typographe rationnel*. Eddie Saudrais, 20/03/2000.
- [5] *Norme ISO 31-11 : 1992 et sa révision ISO 80000-2 : 2009 (extraits)*. <http://aalem.free.fr/maths/mathematiques.pdf>.
- [6] *Typesetting mathematics for science and technology according to ISO 31/XI*, Claudio Beccari, TUGboat Volume 18 (1997), N° 1.
- [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 Nomenclature and Symbols), dec 1999.
- [8] *L^AT_EX Companion*. Frank Mittelbach, Michel Goossens, 2^e édition, Pearson Education France, 2005.
- [9] *The Not So Short Introduction to L^AT_EX 2_ε*. Manuel L^AT_EX de Tobias Oetiker, Hubert Partl, Irene Hyna et Elisabeth Schlegl, CTAN, v6.2 28/02/2018.
- [10] *La distribution mafr*. Extension L^AT_EX de Christian Obrecht, CTAN, v1.0 17/09/2006.
- [11] *L'extension tdsfrmath*. Extension L^AT_EX de Yvon Henel, CTAN, v1.3 22/06/2009.
- [12] *Fourier-GUTenberg*. Extension L^AT_EX de Michel Bovani, CTAN, v1.3 30/01/2005.
- [13] *The mathdesign package*. Extension L^AT_EX de Paul Pichaureau, CTAN, 29/08/2013.
- [14] *A Babel language definition file for French*. Extension L^AT_EX babel-french de Daniel Flipo, CTAN, v3.5c 14/09/2018.
- [15] *The icomma package for L^AT_EX 2_ε*. Extension L^AT_EX de Walter Schmidt, CTAN, v2.0 10/03/2002.
- [16] *The nccomma package*. Alexander I. Rozhenko, CTAN, v1.0 10/02/2005.
- [17] *Intelligent commas*. Claudio Beccari, The PracT_EX Journal, 2011, No.1
- [18] *Typesetting vectors with beautiful arrow with L^AT_EX 2_ε*. Extension L^AT_EX esvect d'Eddie Saudrais, CTAN, v1.3 11/07/2013.
- [19] *mismath – Miscellaneous mathematical macros*. Extension L^AT_EX d'Antoine Missier, CTAN, v1.4 22/05/2019.