

Le gras en maths avec Lua^{La}T_EX et unicode-math

Bascules `\mathbf`/`\symbf` et convention typographique française (guide indicatif)

C. Pierquet

Juillet 2026

Table des matières

1	Le problème	2
2	Les deux familles de bascules	2
2.1	Commandes figées vs commandes sémantiques	2
2.2	Démonstration	2
3	Le cas de la typographie [fr]	3
3.1	Les options <code>bold-style</code> et <code>math-style</code>	3
3.2	La convention typographique française	3
4	La solution pragmatique : bascule manuelle	3
4.1	Idées	3
4.2	Raccourci pratique	3
4.3	Le cas des lettres grecques	4
4.4	Rattraper les anciens réflexes : <code>\mathbf=sym</code>	4
5	Limites à connaître	4
5.1	Les opérateurs en gras?	4
5.2	Les gros opérateurs en <code>displaystyle</code> ?	5
5.3	Polices sans-serif	5
6	Compléments	5
6.1	Mémo	5
6.2	Liens	5

1 Le problème

Avec les polices maths OpenType modernes (unicode-math), mettre une formule en gras n'est **plus aussi simple** qu'avec l'ancien `\mathbf` de $\text{\LaTeX}$ 2_ε. La raison : unicode-math distingue deux familles de commandes qui ne se comportent pas du tout pareil.

À retenir

Ce guide suppose une compilation en Lua $\text{\LaTeX}$ (ou Xe $\text{\LaTeX}$) avec le package unicode-math, et une police maths OpenType complète (ici Erewhon, via fourier-otf).

Code

```
\usepackage[math-style=french,bold-style=ISO]{fourier-otf}
%\unimathsetup{mathbf=sym} %pour forcer mathbf en symbf
```

2 Les deux familles de bascules

2.1 Commandes figées vs commandes sémantiques

- Les commandes **figées** (`\mathrm`, `\mathbf`, `\mathit`, ...) sélectionnent une forme de police **fixe**, héritée du $\text{\LaTeX}$ classique. `\mathbf` reproduit volontairement l'ancien comportement : gras **droit**, uniquement pour le latin, quel que soit le réglage.
- Les commandes **sémantiques** (`\symbf`, `\symbfup`, `\symbfit`, ...) ajoutent le gras **en respectant la convention typographique choisie** (via l'option `bold-style`), et couvrent aussi le grec.

Commande	Comportement	Dépend de bold-style
<code>\mathbf</code>	gras droit, toujours (latin uniquement)	Non
<code>\symbf</code>	gras "sémantique", suit bold-style	Oui
<code>\symbfup</code>	gras droit forcé	Non
<code>\symbfit</code>	gras italique forcé	Non
<code>\symbfsf</code> (up/it)	variantes sans-serif grasses	dépend de sans-style

2.2 Démonstration

Code

```
\texttt{normal} & $f(x)=\sqrt{x+\frac{1}{x}}$ & $AB^2=AC^2+BC^2$\\
\texttt{symbf} & $\symbf{f(x)=\sqrt{x+\frac{1}{x}}}$ & $\symbf{AB^2=AC^2+BC^2}$\\
\texttt{mathbf} & $\mathbf{f(x)=\sqrt{x+\frac{1}{x}}}$ & $\mathbf{AB^2=AC^2+BC^2}$\\
↪ $\mathbf{AB^2=AC^2+BC^2}$\\
```

Rendu

normal	$f(x) = \sqrt{x + \frac{1}{x}}$	$AB^2 = AC^2 + BC^2$
symbf	$\symbf{f(x) = \sqrt{x + \frac{1}{x}}}$	$\symbf{AB^2 = AC^2 + BC^2}$
mathbf	$\mathbf{f(x) = \sqrt{x + \frac{1}{x}}}$	$\mathbf{AB^2 = AC^2 + BC^2}$

Remarque

Sur AB^2 , `symbf` et `mathbf` donnent des résultats différents, indépendamment du mode `math-style=french` déclaré (voir section 3.2).

3 Le cas de la typographie [fr]

3.1 Les options bold-style et math-style

math-style règle la forme des lettres **hors gras** (droit/italique selon la casse). bold-style règle ce que fait `\symbf` **une fois en gras**. Ce sont deux *axes* indépendants.

Valeur de bold-style	Effet sur <code>\symbf</code>
TeX (défaut)	gras droit pour le latin
ISO	gras italique, uniformément (majuscules comprises)
upright	toujours droit
literal (à vérifier)	reprend la forme du caractère <i>tel que tapé</i> dans le source

À retenir

Aucune de ces valeurs ne reproduit le mélange typographique français (minuscule italique / majuscule et grec droits). C'est le *point* que ce guide essaye de combler à la section 4.

3.2 La convention typographique française

En typographie mathématique française, les **minuscules latines** (variables, fonctions) sont en *italique*, tandis que les **majuscules latines** (points, labels) et les **lettres grecques** (minuscules et majuscules) restent **droites** — y compris en gras.

Remarque

L'option `math-style=french` d'`unicode-math` gère déjà cette convention pour le texte **normal** (non gras). Le problème ne se pose que pour le **gras**, où `bold-style` n'a pas d'équivalent "french".

4 La solution pragmatique : bascule manuelle

4.1 Idées

Plutôt que de chercher un réglage global qui n'existe pas, la solution la plus fiable est de **choisir explicitement** selon le contenu :

- `\symbf` pour le cas général (variables, fonctions minuscules);
- `\symbfup` dès qu'il y a des **majuscules** ou des **chiffres** (labels, bornes d'intégrale).

Code

```
\symbf{f(x)=\sqrt{x+\frac{1}{x}}+\displaystyle\int_1^4 f(x) \, \, \symbfup{d}x}$  
  
\symbfup{AB^2=AC^2+BC^2}$  
  
\symbf{\symbfup{AB}^2=\symbfup{AC}^2+\symbfup{BC}^2}$
```

Rendu

$$f(x) = \sqrt{x + \frac{1}{x}} + \int_1^4 f(x) \, dx$$

$$AB^2 = AC^2 + BC^2$$

$$AB^2 = AC^2 + BC^2$$

4.2 Raccourci pratique

Si la formule ne contient **que** des majuscules latines (ex. des labels de points), `\mathbf` donne le même résultat que `\symbfup` et est plus court à taper :

Code

```
\mathbf{AB^{2}=AC^{2}+BC^{2}}$ % équivalent à \sybfup ici
```

4.3 Le cas des lettres grecques

`\mathbf` ne couvre pas le grec, c'est historique : les cinq “text math alphabets” de $\text{\LaTeX}$ (`\mathrm`, `\mathbf`, `\mathit`, `\mathsf`, `\mathtt`) n'ont jamais eu vocation à composer des lettres grecques.

Code

```
\mathbf{\lambda + \mu + \sigma^2}$ % ne produit RIEN de correct
\sybf{\lambda + \mu + \sigma^2}$ % fonctionne
```

Rendu

normal	$\lambda + \mu + \sigma^2$	$\Pi + \Omega$
sybf	$\boldsymbol{\lambda} + \boldsymbol{\mu} + \boldsymbol{\sigma}^2$	$\boldsymbol{\Pi} + \boldsymbol{\Omega}$
mathbf	(rien — bascule ignorée)	(rien)
sybfup	$\boldsymbol{\lambda} + \boldsymbol{\mu} + \boldsymbol{\sigma}^2$	$\boldsymbol{\Pi} + \boldsymbol{\Omega}$

À retenir

Dès qu'une formule contient du grec, il vaut mieux utiliser `\sybf`/`\sybfup`.

4.4 Rattraper les anciens réflexes : `mathbf=sym`

Si du code existant (ou des habitudes bien ancrées) utilise encore `\mathbf`, l'option `mathbf=sym` du package `unicode-math` le transforme en pur synonyme de `\sybf` : il suit alors `bold-style` et couvre le grec.

Code

```
\unimathsetup{mathbf=sym}
```

Remarque

Avec certains packages, comme `fourier-otf`, seule une liste fermée d'options est proposée en *passage* à `unicode-math` (`math-style`, `bold-style`, etc.). Une option comme `mathbf=sym` n'en fait pas partie et sera ignorée s'il est passé en option de `\usepackage`. Il faut l'appliquer **après coup** via `\unimathsetup`, comme ci-dessus.

5 Limites à connaître

5.1 Les opérateurs en gras?

Unicode ne définit **pas de point de code** “+ gras” ou “= gras”. `\sybf`/`\mathbf` n'ont donc **pas de prise** sur les opérateurs (+, =, <, ...) : seules les lettres et les chiffres changent...

À retenir

Pour une formule **entièrement** grasse (opérateurs compris), il faut charger une vraie police Bold et basculer dessus, afin de ne pas mettre chaque lettre en gras individuellement :

```
\setmathfont{Police.otf}[BoldFont=Police-Bold.otf] puis {\boldmath ... }
```

5.2 Les gros opérateurs en displaystyle?

Même avec `\boldmath`, `\boldsymbol` ou `\mathversion{bold}`, un `\int` ou un `\sum` en `displaystyle` peut **rester à sa taille "texte"** en gras. C'est une limitation de la table `MathVariants` de la police `Bold` (les variantes de taille agrandie n'y sont pas toujours dessinées), et non pas un bug réglable côté $\text{\LaTeX}$.

5.3 Polices sans-serif

Une police maths sans-serif (ex. `Lete Sans Math`) n'a en général **pas de cut italique distinct** : `\mathbf` et `\symbf` peuvent alors rendre identique. Ce n'est pas un bug, c'est la tradition typographique du sans-serif en maths (toujours droit).

6 Compléments

6.1 Mémo

Situation	Bascule à utiliser
Variable, fonction (minuscule)	<code>\symbf</code>
Label, point, majuscule, chiffre, borne	<code>\symbfup</code>
Que des majuscules latines	<code>\mathbf</code> (raccourci, si <code>\mathbf=sym</code> non actif)
Formule avec du grec	jamais <code>\mathbf</code> → <code>\symbf</code> / <code>\symbfup</code>
Formule 100 % grasse (opérateurs compris)	<code>\boldmath + BoldFont=...</code>

Remarque

Astuce : ces deux réflexes peuvent être raccourcis en macros perso, par exemple :

```
\newcommand{\grasi}[1]{\symbf{#1}} \newcommand{\grasd}[1]{\symbfup{#1}}
```

Attention toutefois aux collisions de noms avec tes autres packages (`\providecommand` ou un préfixe type `\gras...` évite les surprises).

6.2 Liens

- `unicode-math` : <https://mirrors.ircam.fr/pub/CTAN/macros/unicodetex/latex/unicode-math/unicode-math.pdf>
- `pdf2lua` : <https://danielflipo.fr/doc/luatex/pdf2lua.pdf>