Luciole fonts
User's Guide for LaTeX

Daniel Flipo

daniel.flipo@free.fr

July 3rd 2025

1 What is Luciole ?

Luciole is a set of OpenType text and math fonts developed explicitly for visually impaired people, see https://luciole-vision.com for more information. The text fonts are licensed under Creative-Common CC-by-4.0 and the math font is licensed under Open Font License (OFL), which cover use (including commercial use) and distribution of the fonts for free.

They require LuaTeX or XeTeX as engine and the **unicode-math** package[1], if math fonts are required or just the **fontspec** package[2] otherwise.

2 Usage

The Luciole fonts can be used in normal sizes (10 to 12pt) but visually impaired people will probably need to scale them (option `Scale=2` f.i.), adjusting the baseline skip is recommended, see below.

[1] Please read the documentation **unicode-math.pdf**.
[2] Please read the documentation **fontspec.pdf**.

2.1 Loading text fonts

A file `Luciole.fontspec` is provided to ensure that Italic, Bold and BoldItalic variants are properly loaded. A basic call for Luciole text fonts with scaling could be:

```
\usepackage{fontspec}
\setmainfont{Luciole}[Scale=2]
\renewcommand{\sffamily}{\rmfamily}
\usepackage{realscripts}
\usepackage{setspace}
\setstretch{2.0}
```

This document is composed in `12pt` with `\setstretch{1.2}` and no scaling.

Other options can be added to the `\setmainfont` command, f.i.:
`\setmainfont{Luciole}[Scale=2, Numbers=Lowercase]`

The four text fonts provide proportional, tabular and oldstyle numbers (features `lnum`, `pnum`, `tnum`, `onum`), small caps (features `smcp`, `c2sc`), superscripts and scientific inferiors (digits and lowercase ASCII letters, features `sups`, `sinf`) and the long-s (feature `hist`).

2.2 Loading math fonts with `luciole-math.sty`

Using the `luciole-math` package is the easiest way to load the Luciole math fonts, you can type:

`\usepackage[` *options* [3] `]{luciole-math}`

it loads `unicode-math` with the default options, sets `Luciole-Math.otf` as math font and does a bit more:

1. it checks at `\begin{document}` if packages `amssymb` or `latexsym` are loaded and issues warnings in case they are;
2. it provides aliases for glyphs named differently in Unicode, so that `latexsym` or AMS names are also available;

[3] Possible *options* are `Scale=` or any of the options described in sections 3.1 to 3.4.

3. it defines specific math characters like `\varemptyset` (∅), `\nleqq` (≦̸), `\parallelslant` (⫽), `\shortparallelslant` (⫻), etc.;
4. it redefines the `\TeX`, `\LaTeX` and `\LaTeXe` logos to fit the Luciole fonts unless option `no-logos` has been activated.

The Luciole text fonts have to be loaded as in the previous section.

3 What is provided by Luciole-Math?

Luciole-Math provides a wide range of glyphs including all those available in the `amssymb` and `latexsym` packages. Therefore, the latter two packages *should not* be loaded as they might override Luciole-Math glyphs.

A full list of available glyphs is shown in file `unimath-luciole.pdf`.

3.1 Upright or slanted?

Package `unicode-math` follows TEX conventions for Latin and Greek letters: in math mode, the default option (`math-style=TeX`) prints Latin letters $a…z$ $A…Z$ and lowercase Greek letters $\alpha…\omega$ slanted (italic) while uppercase Greek letters ΑΒΓ…Ω are printed upright. This can be changed by option `math-style` as shown in table 1.

Table 1: Effects of the `math-style` package option.

Package option	Latin	Greek
`math-style=ISO`	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=TeX`	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=french`	$(a, z,$ B, X$)$	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=upright`	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$

Bold letters are printed upright except lowercase Greek letters which are slanted (the default option is `bold-style=TeX`). This can be changed by option `bold-style` as shown in table 2 on the next page.

Other possible customisation: ∇ is printed upright and ∂ is printed slanted by default, but `nabla=italic` and `partial=upright` can change this.

Table 2: Effects of the bold-style package option.

Package option	Latin	Greek
bold-style=ISO	$(\boldsymbol{a},\boldsymbol{z},\boldsymbol{B},\boldsymbol{X})$	$(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Xi})$
bold-style=TeX	$(\mathbf{a},\mathbf{z},\mathbf{B},\mathbf{X})$	$(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{\Gamma},\mathbf{\Xi})$
bold-style=upright	$(\mathbf{a},\mathbf{z},\mathbf{B},\mathbf{X})$	$(\mathbf{\alpha},\mathbf{\beta},\mathbf{\Gamma},\mathbf{\Xi})$

All these options are offered by the unicode-math package, they can be added to the \setmathfont call as well[4], for example:

\setmathfont{Luciole-Math.otf}[math-style=french,partial=upright]
will print for the code

```
\[ \frac{\partial f}{\partial x} = \alpha \symbf{V}
                    + a\nabla\Gamma + \symbf{\beta}\symbf{M} \]
```

$$\frac{\partial f}{\partial x} = \alpha\mathbf{V} + a\nabla\Gamma + \boldsymbol{\beta}\mathbf{M}$$

while the default settings would print

$$\frac{\partial f}{\partial x} = \alpha\mathbf{V} + a\nabla\Gamma + \boldsymbol{\beta}\mathbf{M}$$

Both shapes remain available anytime: $\uppi,\itpi$ prints $\pi,\mathit{\pi}$.

If your text editor is able to handle Greek letters or math symbols, they can be entered in the code instead control sequences (i.e. α, β, Γ,... for \alpha, \beta, \Gamma,...).

3.2 Character variants

Luciole-Math provides eleven "Character Variants" options, listed on table 3 on the following page.

To get 0, \hslash and \emptyset typeset as 0, ℏ and ∅ instead of 0, ℏ and ∅, you can add option CharacterVariant={0,1,2} to the luciole-math call:

\usepackage[CharacterVariant={0,1,2}]{luciole-math}

[4] IMHO it is easier to add *all options* to the \setmathfont command.

Table 3: Character variants.

	Default	Variant	Name
cv01	$\hbar$	$\hbar$	\hslash
cv02	$\emptyset$	$\emptyset$	\emptyset
cv03	ϵ	ε	\epsilon
cv04	κ	$\varkappa$	\kappa
cv05	π	ϖ	\pi
cv06	ϕ	φ	\phi
cv07	ρ	ϱ	\rho
cv08	σ	ς	\sigma
cv09	θ	ϑ	\theta
cv10	Θ	Θ	\Theta
cv11	0	$\emptyset$	0

Please note that curly braces are mandatory whenever more than one "Character Variant" is selected.

Note about \hbar: unicode-math defines \hbar as \hslash (U+210F) while amsmath provides two different glyphs (italic *h* with horizontal or diagonal stroke). luciole-math follows unicode-math; the italic *h* with horizontal stroke can be printed using \hslash or \hbar together with character variant cv01 or with \mithbar (replacement for AMS' command \hbar).

3.3 Stylistic sets

Luciole-Math provides four "Stylistic Sets" options to choose between different glyphs for families of math symbols.

StylisticSet=4, alias[5] Style=leqslant, converts (large) inequalities into their slanted variants, see table 5a on the next page.

StylisticSet=5, alias Style=smaller, converts some symbols into their smaller variants, see table 5b on the following page.

StylisticSet=6, alias Style=subsetneq, converts some inclusion symbols, see table 6a on the next page.

[5]These Style aliases are provided by luciole-math.sty.

Table 4: Stylistic Sets 4 and 5

(a) `Style=leqslant (+ss04)`

Command	Default	Variant
`\leq`	≤	⩽
`\geq`	≥	⩾
`\nleq`	≰	⩽̸
`\ngeq`	≱	⩾̸
`\leqq`	≦	⩽
`\geqq`	≧	⩾
`\eqless`	≤	⩽
`\eqgtr`	≥	⩾
`\lesseqgtr`	⋚	⋚
`\gtreqless`	⋛	⋛
`\lesseqqgtr`	⪋	⪋
`\gtreqqless`	⪌	⪌

(b) `Style=smaller (+ss05)`

Command	Default	Variant
`\in`	∈	∈
`\ni`	∋	∋
`\mid`	∣	∣
`\nmid`	∤	∤
`\parallel`	∥	∥
`\nparallel`	∦	∦
`\parallelslant`	∥	∥
`\nparallelslant`	∦	∦

`StylisticSet=7`, alias `Style=parallelslant`, converts "parallel" symbols into their slanted variants, see table 6b.

Table 5: Stylistic Sets 6 and 7

(a) `Style=subsetneq (+ss06)`

Command	Default	Variant
`\subsetneq`	⊊	⊊
`\supsetneq`	⊋	⊋
`\subsetneqq`	⫋	⫋
`\supsetneqq`	⫌	⫌

(b) `Style=parallelslant (+ss07)`

Command	Default	Variant
`\parallel`	∥	∥
`\nparallel`	∦	∦
`\shortparallel`	∥	∥
`\nshortparallel`	∦	∦

To enable Stylistic Sets 4, 6 and 7 for Luciole-Math, you should enter

```
\setmathfont{Luciole-Math.otf}[StylisticSet={4,6,7}]  or
\usepackage[Style={leqslant,subsetneq,parallelslant}]{luciole-math}
```

then, `\[x\leq y \quad A \subsetneq B\quad D \parallel D'\]` will print

$$x \leqslant y \quad A \subsetneq B \quad D \parallel D'$$

instead of

$$x \leq y \quad A \subsetneq B \quad D \parallel D'$$

3.4 Other font features

3.4.1 Oldstyle numbers

To get oldstyle numbers in math, the feature **+onum** is available:

```
\usepackage[Numbers=OldStyle]{luciole-math}
```

$0123456789, \mathbf{0123456789}$

3.5 Standard LaTeX math commands

All standard LaTeX math commands, all **amssymb** commands and all **latexsym** commands are supported by Luciole-Math, loading the **luciole-math** package is required for some of them.

Various wide accents are also supported:

- ☞ **\wideoverbar** and **\mathunderbar**[6]

$$\overline{x} \quad \overline{xy} \quad \overline{xyz} \quad \overline{A \cup B} \quad \overline{A \cup (B \cap C) \cup D} \quad \underline{m+n+p}$$

- ☞ **\widehat** and **\widetilde**

$$\hat{x}\ \widehat{xx}\ \widehat{xxx}\ \widehat{xxxx}\ \widehat{xxxxx}\ \widehat{xxxxxx} \quad \tilde{x}\ \widetilde{xx}\ \widetilde{xxx}\ \widetilde{xxxx}\ \widetilde{xxxxx}\ \widetilde{xxxxxx}$$

- ☞ **\widecheck** and **\widebreve**

$$\check{x}\ \widecheck{xx}\ \widecheck{xxx}\ \widecheck{xxxx}\ \widecheck{xxxxx}\ \widecheck{xxxxxx} \quad \breve{x}\ \widebreve{xx}\ \widebreve{xxx}\ \widebreve{xxxx}\ \widebreve{xxxxx}\ \widebreve{xxxxxx}$$

- ☞ **\overparen** and **\underparen**

$$\overparen{x} \quad \overparen{xy} \quad \overparen{xyz} \quad \overparen{\overset{\circ}{A \cup B}} \quad \overparen{\overset{\circ}{A \cup (B \cap C) \cup D}} \quad \overparen{\overset{2}{x+y}} \quad \overparen{\overset{26}{a+b+\ldots+z}}$$

[6] **\overline** and **\underline** are not font related, they are based on **\rule**.

$$\underbrace{x} \quad \underbrace{xz} \quad \underbrace{xyz} \quad \underbrace{x+z}_{2} \quad \underbrace{a+b+\ldots+z}_{26}$$

☞ **\overbrace** and **\underbrace**

$$\overbrace{a} \quad \overbrace{ab} \quad \overbrace{abc} \quad \overbrace{abcd} \quad \overbrace{abcde} \quad \overbrace{a+b+c}^{3} \quad \overbrace{a+b+\ldots+z}^{26}$$

$$\underbrace{a} \quad \underbrace{ab} \quad \underbrace{abc} \quad \underbrace{abcd} \quad \underbrace{abcde} \quad \underbrace{a+b+c}_{3} \quad \underbrace{a+b+\ldots+z}_{26}$$

☞ **\overbracket** and **\underbracket**

$$\overbracket{a} \quad \overbracket{ab} \quad \overbracket{abc} \quad \overbracket{abcd} \quad \overbracket{abcde} \quad \overbracket{a+b+c}^{3} \quad \overbracket{a+b+\ldots+z}^{26}$$

$$\underbracket{a} \quad \underbracket{ab} \quad \underbracket{abc} \quad \underbracket{abcd} \quad \underbracket{abcde} \quad \underbracket{a+b+c}_{3} \quad \underbrace{a+b+\ldots+z}_{26}$$

☞ **\overrightarrow** and **\overleftarrow**

$$\vec{v} \quad \vec{M} \quad \vec{vv} \quad \vec{AB} \quad \vec{ABC} \quad \vec{ABCD} \quad \vec{ABCDEFGH}.$$

$$\overleftarrow{v} \quad \overleftarrow{M} \quad \overleftarrow{vv} \quad \overleftarrow{AB} \quad \overleftarrow{ABC} \quad \overleftarrow{ABCD} \quad \overleftarrow{ABCDEFGH}$$

☞ **\overrightharpoon** and **\overleftharpoon**

$$\vec{v} \quad \vec{M} \quad \vec{vv} \quad \vec{AB} \quad \vec{ABC} \quad \vec{ABCD} \quad \vec{ABCDEFGH}.$$

$$\overleftarrow{v} \quad \overleftarrow{M} \quad \overleftarrow{vv} \quad \overleftarrow{AB} \quad \overleftarrow{ABC} \quad \overleftarrow{ABCD} \quad \overleftarrow{ABCDEFGH}$$

☞ **\underrightarrow** and **\underleftarrow**

$$\underrightarrow{v} \quad \underrightarrow{M} \quad \underrightarrow{vv} \quad \underrightarrow{AB} \quad \underrightarrow{ABC} \quad \underrightarrow{ABCD} \quad \underrightarrow{ABCDEFGH}.$$

$$\underleftarrow{v} \quad \underleftarrow{M} \quad \underleftarrow{vv} \quad \underleftarrow{AB} \quad \underleftarrow{ABC} \quad \underleftarrow{ABCD} \quad \underleftarrow{ABCDEFGH}$$

☞ **\underrightharpoon** and **\underleftharpoondown**

$$\underrightarrow{v} \quad \underrightarrow{M} \quad \underrightarrow{vv} \quad \underrightarrow{AB} \quad \underrightarrow{ABC} \quad \underrightarrow{ABCD} \quad \underrightarrow{ABCDEFGH}.$$

$$\underleftarrow{v} \quad \underleftarrow{M} \quad \underleftarrow{vv} \quad \underleftarrow{AB} \quad \underleftarrow{ABC} \quad \underleftarrow{ABCD} \quad \underleftarrow{ABCDEFGH}.$$

☞ Finally **\widearc** and **\overrightarc** (loading **luciole-math.sty** is required)

$$\widearc{AMB} \quad \overrightarc{AMB}$$

All the extensible arrows provided by the **mathtools** package are available in the Luciole-Math font (loading **luciole-math.sty** is required), f.i.:

$$X \xLeftrightarrow{\text{above}} Y \xhookrightarrow[\text{under}]{} Z \xmapsto{\text{above}} W$$

A wide range of extensible vertical delimiters is provided:

$$\left/ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \end{vmatrix} \begin{Vmatrix} a_1 \\ a_2 \\ a_3 \end{Vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \end{vmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{vmatrix} a_1 \\ a_2 \\ a_3 \end{vmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \left\langle \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} \right\rangle \left\langle\!\left\langle \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} \right\rangle\!\right\rangle \backslash$$

3.6 Mathematical alphabets

☞ All Latin and Greek characters are available in italic, upright, bold and bold italic via the **\symit{}**, **\symup{}**, **\symbf{}** and **\symbfit{}** commands.

☞ Calligraphic alphabet (**\symscr** or **\symcal** command), uppercase:
$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
also in boldface (**\symbfscr**,**\symbfcal** or **\mathbfcal** command):
$\boldsymbol{\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}}$

☞ Blackboard-bold alphabet uppercase (**\symbb** command) and **\Bbbk**:

$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZk}$

☞ Fraktur alphabet :

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}$

☞ Typewriter alphabet:

$\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}$

Like Latin Modern, Luciole-Math provides only four lowercase Latin letters in script (or calligraphic) shape: $\mathscr{e}$, $\mathscr{g}$, $\mathscr{l}$, $\mathscr{o}$ (**\mscre**, **\mscrg**, **\ell**, **\mscro**).

All others (range "1D4B6 to "1D4CF) have to be borrowed from another math font if needed, i.e.

```
\setmathfont{NotoSansMath-Regular.otf}%
            [range="1D4B6-"1D4CF, Scale=MatchLowercase]
```

Please remember that the *last loaded* font sets the MATH TABLE, so it is recommended to reload the base font with an empty range argument:

```
\setmathfont{Luciole-Math.otf}[range={}]
```

3.7 Bold variant

In case short math formulas have to be printed in section titles, a *limited* bold variant is provided.

Example of usage: **Einstein's equation $E = mc^2$**

```
\setmathfont{Luciole-Math-Bold.otf}[version=bold, options]
\section{\mathversion{bold} Einstein's equation $E=mc^2$}
```

It is also possible to use the \boldmath command, this way:

```
\setmathfont{Luciole-Math-Regular.otf}%
            [BoldFont = Luciole-Math-Bold.otf]
\section{\boldmath Einstein's equation $E=mc^2$}
```

3.8 Missing symbols

Luciole-Math does not aim at being as complete as NotoSansMath-Regular or Cambria, the current glyph coverage compares with Latin Modern or TeXGyre math fonts. In case some symbols do not show up in the output file, you will see warnings in the .log file, for instance:

```
Missing character: There is no ⇒ (U+2964) in font LucioleMath
```

Borrowing them from a more complete font, say NotoSansMath, is a possible workaround:

```
\setmathfont{NotoSansMath-Regular.otf}[range={"2964},Scale=2]
```

scaling is possible, multiple character ranges are separated with commas:

```
\setmathfont{NotoSansMath-Regular.otf}[range={"294A-"2951,"2964}]
```

Let's mention **albatross**, a useful tool to find out the list of fonts providing a given glyph: f.i. type in a terminal "**albatross -t U+2964**", see the manpage or **albatross-manual.pdf**.

4 Acknowledgements

I am grateful to George Williams and his co-workers for providing and maintaining FontForge, to Ulrik Vieth for his illuminating paper published in TUGboat 2009 Volume 30 about OpenType Math and to Khaled Hosny for his essential survey about OpenType Math Fonts https://github.com/notofonts/math/blob/main/documentation/building-math-fonts/index.md.