

∞ XCharter-Math ∞

Daniel Flipo
daniel.flipo@free.fr

26th October 2023

1 What is XCharter-Math?

XCharter-Math is an OpenType maths font based on Bitstream Charter meant to be used with XCharter text fonts.

Latin and Greek letters and many maths symbols are borrowed or derived from Michael Sharpe's XCharter fonts. Other sources for maths glyphs were found in MathDesign (by Paul Pichaureau) and Fourier-GUTenberg (by Michel Bovani).

It requires LuaTeX or XeTeX as engine and the `unicode-math` package¹.

Please note that the current version (0.51) is *experimental*, *do expect metrics and glyphs to change* until version 1.0 is reached. Comments, suggestions and bug reports are welcome!

2 Usage

2.1 Calling `\setmathfont`

A basic call for XCharter-Math would be:

```
\usepackage{unicode-math}  
\setmathfont{XCharter-Math.otf} % Call by file name or  
\setmathfont{XCharter Math}    % Call by font name
```

this loads XCharter-Math as maths font² with the default options, see subsections [3.1 on page 3](#), [3.2 on page 4](#) and [3.3 on page 5](#) for customisation.

Please note that the three sets of text fonts have to be chosen separately, f.i.:

¹Please read the documentation `unicode-math.pdf`.

²Both calls work equally well with LuaTeX; with XeTeX a call by font name will fail unless the font is declared as a *system font*.

```

\setmainfont{XCharter} % rm
\setsansfont{Cabin}[Scale=MatchLowercase] % sf
\setmonofont{Inconsolatazi4}[Scale=MatchLowercase] % tt

```

otherwise you would get Latin Modern for text fonts.

2.2 Calling `xcharter-otf.sty` (recommended)

As an alternative to load XCharter-Math you can type:

```
\usepackage[ options 3 ]{xcharter-otf}
```

it loads `unicode-math` with the default options, sets XCharter-Math as maths font and XCharter Text fonts as Roman fonts (families *sf* and *tt* left unchanged) but does a bit more:

1. it loads `realscripts` for better superscripts;
2. it checks at `\begin{document}` if packages `amssymb` or `latexsym` are loaded and issues warnings in case they are;
3. it provides aliases for glyphs named differently in Unicode, so that `latexsym` or AMS names are also available;
4. it defines specific maths characters like `\varempyset` (\emptyset), `\parallelslant` ($//$), `\shortparallelslant` ($//$), etc.;
5. it reduces spacing in maths mode: `\thinmuskip`, `\medmuskip` and `\thickmuskip` unless the `loose` option is activated.

Apart from the `loose` option mentioned above, `xcharter-otf.sty` provides two options `no-text` and `Scale=<decimal>` meant to be used to load the XCharter-Math font together with roman text fonts other than XCharter, while keeping the advantages 1. to 5. pointed in the preceding list, f.i.

```
\usepackage[no-text, Scale=0.98]{xcharter-otf}
```

Option `no-text` can also be useful if XCharter is to be loaded with specific options, f.i.

```
\usepackage[no-text]{xcharter-otf}
```

```
\setmainfont{XCharter}[RawFeature=+onum;+ss01]
```

3 What is provided?

XCharter-Math provides all glyphs supplied by `Fourier-GUTenberg` plus all glyphs available in the `amssymb` and `latexsym` packages and many more. Therefore, these two packages *should not* be loaded as they might override XCharter-Math glyphs.

³Possible *options* are `loose`, `no-text`, `Scale=` or any of the options described in sections 3.1, 3.2 and 3.3.

Sans-serif, typewriter and fraktur alphabets are borrowed from Latin Modern fonts. See in section 3.6 on page 8 how to choose from other maths fonts for these styles. A full list of available glyphs is shown in file `unimath-xcharter.pdf`.

3.1 Upright or slanted?

Package `unicode-math` follows \TeX conventions for Latin and Greek letters: in maths mode, the default option (`math-style=TeX`) prints Latin letters $a\dots z$ $A\dots Z$ and lowercase Greek letters $\alpha\dots\omega$ slanted (italic) while uppercase Greek letters $\text{A}\Gamma\dots\Omega$ 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
<code>math-style=ISO</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=TeX</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=french</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=upright</code>	(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.

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

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

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

All these options are offered by the `unicode-math` package but they can be added to the `\setmathfont` call⁴, for example:

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

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

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

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

while the default settings would print

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

Both shapes remain available anytime: `\uppi`, `\itpi` prints π , π .

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

3.2 Character variants

XCharter-Math provides fourteen “Character Variants” options, listed on table 3, to choose between different glyphs for Greek characters and some others. Alternative calligraphic capitals for E, Q and T have been added (version 0.50) and an alternative italic v , easier to distinguish from ν (version 0.51).

Table 3: Character variants.

	Default	Variant	Name
cv00	0	0	0
cv01	ħ	ħ	<code>\hslash</code>
cv02	∅	∅	<code>\emptyset</code>
cv03	ε	ε	<code>\epsilon</code>
cv04	κ	κ	<code>\kappa</code>
cv05	π	π	<code>\pi</code>
cv06	φ	φ	<code>\phi</code>
cv07	ρ	ρ	<code>\rho</code>
cv08	σ	ς	<code>\sigma</code>
cv09	θ	θ	<code>\theta</code>
cv10	Θ	Θ	<code>\Theta</code>
cv11	<i>v</i>	<i>v</i>	<i>v</i>
cv20	ℰ	ℰ	<code>\symcal{E}</code>
cv21	ℚ	ℚ	<code>\symcal{Q}</code>
cv22	ℒ	ℒ	<code>\symcal{T}</code>

For instance, to get `\epsilon` and `\phi` typeset as ϵ and ϕ instead of ϵ and ϕ , you can add option `CharacterVariant={3,6}` to the `\setmathfont` call:

```
\setmathfont{XCharter-Math.otf}[CharacterVariant={3,6}]
```

This works for all shapes and weights of these characters: f.i. `\symbf{\epsilon}`, `\symbf{\phi}` are output as ϵ , ϕ instead of ϵ , ϕ .

Similarly with `math-style=french`, `\epsilon` and `\phi` are output as ϵ and ϕ (up-right).

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

Note about `\hbar` (v0.43): `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `XCharter-Math` now follows `unicode-math`; the italic h with horizontal stroke can be printed using `\hslash` or `\hbar` together with character variant `cv01` or with `\mi thbar` (replacement for AMS' command `\hbar`).

3.3 Stylistic sets

`XCharter-Math` provides five “Stylistic Sets” options to choose between different glyphs for families of maths symbols.

`StylisticSet=4`, alias⁵ `Style=leqslant`, converts large inequalities into their slanted variants, see table 5a.

`StylisticSet=5`, alias `Style=smaller`, converts some symbols into their smaller variants, see table 5b.

Table 4: Stylistic Sets 4 and 5

(a) <code>Style=leqslant</code> (+ss04)			(b) <code>Style=smaller</code> (+ss05)		
Command	Default	Variant	Command	Default	Variant
<code>\leq</code>	\leq	\leqslant	<code>\in</code>	\in	\in
<code>\geq</code>	\geq	\geqslant	<code>\ni</code>	\ni	\ni
<code>\nleq</code>	$\not\leq$	$\not\leqslant$	<code>\mid</code>	\mid	\mid
<code>\ngeq</code>	$\not\geq$	$\not\geqslant$	<code>\nmid</code>	\nmid	\nmid
<code>\leqq</code>	\leqq	\leqslant	<code>\parallel</code>	\parallel	\parallel
<code>\geqq</code>	\geqq	\geqslant	<code>\nparallel</code>	\nparallel	\nparallel
<code>\leqless</code>	\leqless	\leqless	<code>\parallelslant</code>	\parallel	\parallel
<code>\eqgtr</code>	\eqgtr	\eqgtr	<code>\nparallelslant</code>	\nparallel	\nparallel
<code>\lesseqgtr</code>	\lesseqgtr	\lesseqgtr			
<code>\gtreqless</code>	\gtreqless	\gtreqless			
<code>\lesseqqgtr</code>	\lesseqqgtr	\lesseqqgtr			
<code>\gtreqqlless</code>	\gtreqqlless	\gtreqqlless			

`StylisticSet=6`, alias `Style=subsetneq`, converts some inclusion symbols, as shown in table 6a on the next page.

`StylisticSet=7`, alias `Style=parallelslant`, converts “parallel” symbols into their slanted variants, see table 6b on the following page.

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

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

⁵These `Style` aliases are provided by `xcharter-otf.sty`.

Table 5: Stylistic Sets 6 and 7

(a) Style=subsetneq (+ss06)			(b) Style=parallelslant (+ss07)		
Command	Default	Variant	Command	Default	Variant
<code>\subsetneq</code>	\subsetneq	\subsetneq	<code>\parallel</code>	\parallel	\parallel
<code>\supsetneq</code>	\supsetneq	\supsetneq	<code>\nparallel</code>	\nparallel	\nparallel
<code>\subsetneqq</code>	\subsetneqq	\subsetneqq	<code>\shortparallel</code>	\shortparallel	\shortparallel
<code>\supsetneqq</code>	\supsetneqq	\supsetneqq	<code>\nshortparallel</code>	\nshortparallel	\nshortparallel

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

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

instead of

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

StylisticSet=3, alias⁶ Style=upint, converts integrals signs into their upright variants, see table 6.

Table 6: Style=upint (+ss03)

Command	<code>\int</code>	<code>\iint</code>	<code>\iiint</code>	<code>\iiiiint</code>	<code>\oint</code>	<code>\oiint</code>	<code>\oiiint</code>	
Default	\int	\iint	\iiint	\iiiiint	\oint	\oiint	\oiiint	\oiiiiint
Upright	\int	\iint	\iiint	\iiiiint	\oint	\oiint	\oiiint	\oiiiiint

Command	<code>\intclockwise</code>	<code>\awint</code>	<code>\varointclockwise</code>	<code>\ointctrckwise</code>
Default	\int	\int	\oint	\oint
Upright	\int	\int	\oint	\oint

3.4 Other font features

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

`\setmathfont{XCharter-Math.otf}[Numbers=OldStyle]` or
`\usepackage[Style=fulloldstyle]{xcharter-otf}`

0123456789, **0123456789**

⁶These Style aliases are provided by `xcharter-otf.sty`.

3.5 Standard \LaTeX math commands

All standard \LaTeX maths commands, all `amssymb` commands and all `latexsym` commands are supported by XCharter-Math, for some of them loading `xcharter-otf.sty` is required.

Various wide accents are also supported:

☞ `\wideoverbar` and `\mathunderbar`⁷

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

☞ `\widehat` and `\widetilde`

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

☞ `\widecheck` and `\widebreve`

$$\check{x} \quad \widecheck{xxx} \quad \widecheck{xxxxx} \quad \breve{x} \quad \breve{xxx} \quad \breve{xxxxx}$$

☞ `\overparen` and `\underparen`

$$\overparen{x} \quad \overparen{xy} \quad \overparen{xyz} \quad \overparen{A \cup B} \quad \overparen{A \cup (B \cap C) \cup D} \quad \overparen{x+y} \quad \overparen{a+b+\dots+z}$$

$$\underparen{x} \quad \underparen{xz} \quad \underparen{xyz} \quad \underparen{x+z} \quad \underparen{a+b+\dots+z}$$

☞ `\overbrace` and `\underbrace`

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

$$\underbrace{a} \quad \underbrace{ab} \quad \underbrace{abc} \quad \underbrace{abcd} \quad \underbrace{abcde} \quad \underbrace{a+b+c}_3 \quad \underbrace{a+b+\dots+z}_{26}$$

☞ `\overbracket` and `\underbracket`

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

$$\underbracket{a} \quad \underbracket{ab} \quad \underbracket{abc} \quad \underbracket{abcd} \quad \underbracket{abcde} \quad \underbracket{a+b+c}_3 \quad \underbracket{a+b+\dots+z}_{26}$$

☞ `\overrightarrow` and `\overleftarrow`

$$\overrightarrow{v} \quad \overrightarrow{M} \quad \overrightarrow{v} \quad \overrightarrow{AB} \quad \overrightarrow{ABC} \quad \overrightarrow{ABCD} \quad \overrightarrow{ABCDEFGH}$$

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

⁷`\overline` and `\underline` are not font related, they are based on `\rule`.

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

- ☞ Sans serif alphabet is borrowed from Latin Modern,
ABCDEFGHIJKLMNOPQRSTUVWXYZmnopqrstuvwxyz
but it can be borrowed from another maths font, i.e.

```
\setmathfont{STIXTwoMath-Regular.otf}[range={sfup,sfit},  
                                         Scale=MatchUppercase]  
\symsfup{ABCD...klm}\quad\symsfit{NOPQ...xyz}$
```

ABCDEFGHIJKLMNOPQRSTUVWXYZmnopqrstuvwxyz

- ☞ Typewriter alphabet is borrowed from Latin Modern,
ABCDEFGHIJKLMNOPQRSTUVWXYZmnopqrstuvwxyz
but it can be borrowed from another maths font, i.e.

```
\setmathfont{STIXTwoMath-Regular.otf}[range=tt,  
                                         Scale=MatchUppercase]  
\symtt{ABCDE...XYZ abcde...xyz}$
```

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

Like Latin Modern, XCharter-Math provides only four lowercase latin letters in script (or calligraphic) shape: *e*, *g*, *l*, *o* (`\mscre`, `\mscrg`, `\ell`, `\mscro`).

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

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

3.7 Bold variant

In case short maths formulas have to be printed in section titles, a *limited* bold variant has been added in version 0.35. Example of usage:

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

It is also possible to use the `\boldmath` command:

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

3.8 Missing symbols

XCharter-Math does not aim at being as complete as STIXTwoMath-Regular or Cambria, the current glyph coverage compares with TeXGyre maths 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 \Rightarrow (U+2964) in font XCharterMath

Borrowing them from a more complete font, say Asana-Math, is a possible workaround:

```
\setmathfont{Asana-Math.otf}[range={"2964"},Scale=1.02]
```

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

```
\setmathfont{Asana-Math.otf}[range={"294A-"2951","2964","2ABB-"2ABE"}]
```

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 U+2964", see the manpage or albatross-manual.pdf.

4 Acknowledgements

Many thanks to Michael Sharpe for providing the XCharter text fonts in OpenType format.

I am grateful to George Williams and his co-workers for providing and maintaining FontForge and to Ulrik Vieth for his illuminating paper published in TUGboat 2009 Volume 30 about Open Type Math.

