

Concrete Math font, OTF version

Daniel Flipo
daniel.flipo@free.fr

20th July 2023

1 What is concmath-otf?

The concmath-otf package offers an OpenType version of the Concrete Math font created by Ulrik Vieth in MetaFont. concmath-otf.sty is a replacement for the original concmath.sty package.

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

Please note that the current version (0.50) 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 concmath-otf would be:

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

this loads concmath-otf as maths font ² with the default options, see subsections [3.1 on the following page](#), [3.2 on page 3](#) and [3.3 on page 4](#) for customisation.

Please note that the three sets of text fonts have to be chosen separately, f.i. if you want the Concrete text fonts³ as Roman font:

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*.

³They are part of the `cm-unicode` package.

```
\setmainfont{cmunorm.otf}
  [BoldFont =      cmunobx.otf ,
   ItalicFont =    cmunoti.otf ,
   BoldItalicFont = cmunobi.otf ]
```

otherwise you would get Latin Modern for text fonts (rm, sf and tt).

2.2 Calling concmath-otf.sty

A (recommended) alternative is:

```
\usepackage[ options 4 ]{concmath-otf}
```

it loads unicode-math with the default options, sets Concrete-Math as maths font and Concrete text fonts as Roman fonts (families *sf* and *tt* left unchanged) but 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. it reduces spacing in maths mode: `\thinmuskip`, `\medmuskip` and `\thickmuskip` are reduced as in `fourier.sty`. The option `loose` disables these settings.

Apart from the `loose` option mentioned above, `concmath-otf.sty` provides an option `no-text` to be used for loading the `concmath-otf` font together with roman text fonts other than Concrete.

3 What is provided?

`concmath-otf` provides all glyphs available in the `concmath`, `amssymb` and `latexsym` packages and more. Therefore, these two packages *should not* be loaded as they might override `concmath-otf` glyphs.

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

See in section 3.5 on page 7 how to choose from other maths fonts for these styles.

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 $a\dots z$ $A\dots Z$ 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 on the following page.

⁴Possible *options* are `loose`, `no-text`, `Scale=` or any of the options described in sections 3.1, 3.2 and 3.3.

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>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>bold-style=TeX</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>bold-style=upright</code>	(a, z, B, 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` call⁵, for example:

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

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

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

while the default settings would print

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

Both shapes remain available anytime: `\muppi`, `\mitpi` 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. $\alpha, \beta, \Gamma, \dots$ for `\alpha`, `\beta`, `\Gamma`, `\dots`).

3.2 Character variants

`concmath-otf` provides ten “Character Variants” options, listed on table 3 on the following page, to choose between different glyphs for Greek characters and some others.

⁵IMHO it is easier to add *all options* to the `\setmathfont` command.

Table 3: Character variants.

	Default	Variant	Name
cv01	\hbar	\hbar	<code>\hslash</code>
cv02	\emptyset	\emptyset	<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>

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{Concrete-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 (upright).

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

Note: `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `concmath-otf` 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

`concmath-otf` provides four “Stylistic Sets” options to choose between different glyphs for families of maths symbols.

`StylisticSet=4`, alias⁶ `Style=leqslant`, converts (large) inequalities into their slanted variants as shown by table 5a on the next page.

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

`StylisticSet=6`, alias `Style=subsetneq`, converts some inclusion symbols as shown by table 5 on the next page.

To enable Stylistic Sets 4 and 6 for `concmath-otf`, you should enter

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

Table 4: Stylistic Sets 4 and 5

(a) Style=leqslant (+ss04)			(b) Style=smaller (+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>\nleqq</code>	$\not\leqq$	$\not\leqslant$			
<code>\ngeqq</code>	$\not\geqq$	$\not\geqslant$			
<code>\eqless</code>	\lessdot	\lessdot			
<code>\eqgtr</code>	\gtrdot	\gtrdot			
<code>\lesseqgtr</code>	\lessgtr	\lessgtr			
<code>\gtreqless</code>	\gtrless	\gtrless			
<code>\lesseqqgtr</code>	\lessgtr	\lessgtr			
<code>\gtreqqless</code>	\gtrless	\gtrless			

Table 5: Stylistic Sets 6

Command	Default	Variant
<code>\subsetneq</code>	\subsetneq	\subsetneq
<code>\supsetneq</code>	\supsetneq	\supsetneq
<code>\subsetneqq</code>	\subsetneqq	\subsetneqq
<code>\supsetneqq</code>	\supsetneqq	\supsetneqq

`\setmathfont{Concrete-Math.otf}[StylisticSet={4,6}]` or
`\usepackage[Style={leqslant,subsetneq}]{concmath-otf}`

then, `\[x\leq y \quad A \subsetneq B\]` will print as
 $x \leqslant y \quad A \subsetneqq B$ instead of $x \leq y \quad A \subsetneq B$

StylisticSet=3, alias⁷ Style=upint, converts integrals signs into their upright variants, see table 6 on the following page.

3.4 Standard L^AT_EX math commands

All standard L^AT_EX maths commands, all amssymb commands and all latexsym commands are supported by concmath-otf, for some of them loading concmath-otf.sty is required.

Various wide accents are also supported:

⁷These Style aliases are provided by concmath-otf.sty.

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	
Upright	\int	\iint	\iiint	\iiiiint	\oint	\oiint	\oiiint	

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

- `\wideoverbar` and `\mathunderbar`⁸

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

- `\widehat` and `\widetilde`

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

- `\widecheck` and `\widebreve`

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

- `\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`

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

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

⁸`\overline` and `\underline` are not font related, they are based on `\rule`.

- `\overbracket` and `\underbracket`

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

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

- `\overrightarrow` and `\overleftarrow`

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

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

- `\overrightarrow` and `\overleftarrow`

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

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

- `\underrightarrow` and `\underleftarrow`

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

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

- `\underrightarrow` and `\underleftarrow`

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

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

- Finally `\widearc` and `\overrightarrow` (loading `concmath-otf.sty` is required)

$$\widearc{AMB} \quad \overrightarrow{AMB}$$

3.5 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 uppercase only (commands `\symscr` or `\symcal`), also in Bold (commands `\symbfscr` or `\symbfcal`):

ABCDEFGHIJKLMN OPQRSTUVWXYZ
ABCDEFGHIJKLMN OPQRSTUVWXYZ

- Blackboard-bold alphabet (`\symbb` or `\mathbb` command):

ABCDEFGHIJKLMN OPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz 0123456789

- Fraktur alphabet, borrowed from Latin Modern:
 $\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}$
 any alphabet can be overwritten, i.e.

```
\setmathfont{Asana-Math.otf}[range=frak,Scale=MatchUppercase]
 $\frac{ABCDEFGHIJKLMNOPQRSTUVWXYZ}{abcdefghijklmnopqrstuvwxyz}$ 
```

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}$

- Sans-serif (Latin and Greek) and Typewriter (Latin) alphabets (commands `\symsfup{}`, `\symsfit{}`, `\symbfsfup{}`, `\symbfsfit{}`, `\symttt{}`):

ABCDEFGHIJKLMabcdefghijklm *NOPQRSTUVWXYZnoprstuvwxyz*

ΑΒΓΔΕΖΗΘΙΚΛΜαβγδεζηθικλμ *ΝΞΟΠΡΣΤΥΦΧΨΩνξοπρστυφχψω*

ABCDEFGHIJKLMNPNOPQRSTUVWXYZabcdefghijklmnpqrstuvwxyz

3.6 Missing symbols

`concmath-otf` 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 `ConcreteMath`

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

The original Metafont glyphs have been converted first to Type 1 (pfa) using `mftrace` and `fontforge`. The `cm-unicode` package has also helped a lot while cleaning the glyphs.

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 `OpenType Math`.