Concrete Math font, OTF version

Daniel Flipo
daniel.flipo@free.fr

28th March 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 package[1].

Please note that the current version (0.40) 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 4 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 fonts[3] as Roman font:

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

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

[3] They 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 `concmat-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.

Sans-serif, typewriter glyphs are not supplied. 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 $a...z$ $A...Z$ and lowercase Greek letters $\alpha...\omega$ slanted (italic) while uppercase Greek letters $AB\Gamma...\Omega$ are printed upright. This can be changed by option `math-style` as shown in table 1.

[4] Possible *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
`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`	$(a, z, \mathrm{B}, \mathrm{X})$	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=upright`	$(\mathrm{a}, \mathrm{z}, \mathrm{B}, \mathrm{X})$	$(\alpha, \beta, \Gamma, \Xi)$

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

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` call[5], for example:

`\setmathfont{Concrete-Math.otf}[math-style=french,partial=upright]`

will print for the code

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

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

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

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

[5] 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	$\varTheta$	\Theta

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 $\boldsymbol{\varepsilon}$, $\boldsymbol{\varphi}$ instead of $\boldsymbol{\epsilon}$, $\boldsymbol{\phi}$.

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, alias[6] 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.

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

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

Table 4: Stylistic Sets 4 and 5

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

Command	Default	Variant
`\leq`	$\leq$	$\leqslant$
`\geq`	$\geq$	$\geqslant$
`\nleq`	$\not\leq$	$\not\leqslant$
`\ngeq`	$\not\geq$	$\not\geqslant$
`\leqq`	$\leqq$	$\leqslant$
`\geqq`	$\geqq$	$\geqslant$
`\nleqq`	$\not\leqq$	$\not\leqslant$
`\ngeqq`	$\not\geqq$	$\not\geqslant$
`\eqless`	$\eqless$	$\leqslant$
`\eqgtr`	$\eqgtr$	$\geqslant$
`\lesseqgtr`	$\lesseqgtr$	$\lesseqgtr$
`\gtreqless`	$\gtreqless$	$\gtreqless$
`\lesseqqgtr`	$\lesseqqgtr$	$\lesseqqgtr$
`\gtreqqless`	$\gtreqqless$	$\gtreqqless$

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

Command	Default	Variant
`\mid`	$\mid$	$\mid$
`\nmid`	$\nmid$	$\nmid$
`\parallel`	$\parallel$	$\parallel$
`\nparallel`	$\nparallel$	$\nparallel$

Table 5: Stylistic Sets 6

Command	Default	Variant
`\subsetneq`	$\subsetneq$	$\subsetneq$
`\supsetneq`	$\supsetneq$	$\supsetneq$
`\subsetneqq`	$\subsetneqq$	$\subsetneqq$
`\supsetneqq`	$\supsetneqq$	$\supsetneqq$

`\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
$x \leqslant y \quad A \subsetneq B$ instead of $x \leq y \quad A \subsetneq B$

3.4 Standard LaTeX math commands

All standard LaTeX 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:

- `\wideoverbar` and `\mathunderbar`[7]

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

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

- `\widehat` and `\widetilde`

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

- `\widecheck` and `\widebreve`

$$\breve{x}\quad \widecheck{xxxx}\quad \widecheck{xxxxxxx}\quad \breve{x}\quad \widebreve{xxxx}\quad \widebreve{xxxxxxx}$$

- `\overparen` and `\underparen`

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

$$\underparen{x}\quad \underparen{xz}\quad \underparen{xyz}\quad \underparen{\underset{2}{x+z}}\quad \underparen{\underset{26}{a+b+\ldots+z}}$$

- `\overbrace` and `\underbrace`

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

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

- `\overbracket` and `\underbracket`

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

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

- `\overrightarrow` and `\overleftarrow`

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

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

- `\overrightharpoon` and `\overleftharpoon`

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

$$\overleftharpoon{v}\ \overleftharpoon{M}\ \overleftharpoon{vv}\ \overleftharpoon{AB}\ \overleftharpoon{ABC}\ \overleftharpoon{ABCD}\ \overleftharpoon{ABCDEFGH}$$

- `\underrightarrow` and `\underleftarrow`

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

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

- `\underrightharpoon` and `\underleftharpoondown`

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

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

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

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

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 (`\symscr` or `\symcal` or `\mathcal` command), uppercase:
  $\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
- Blackboard-bold alphabet (`\symbb` or `\mathbb` command), uppercase only except lowercase `\Bbbk` (AMS)
  $\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\quad\Bbbk$
- Fraktur alphabet, borrowed from Latin Modern
  $\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}$
  but this can overwritten, i.e.

```
\setmathfont{Asana-Math.otf}[range=frak,Scale=MatchUppercase]
$\symfrak{ABCDEFGHIJKL...XYZ abcdefghijkl...xyz}$
```

  $\mathfrak{ABCDEFGHIJKL...XYZ\ abcdefghijkl...xyz}$

- Sans serif and Typewriter alphabets have to be imported, i.e.

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

  $\mathsf{ABCD...klm}\quad\mathsf{\mathit{NOPQ...xyz}}$

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

  $\mathtt{ABCDE...XYZ\ abcde...xyz}$

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 ⇒ (U+2964) in font ErewhonMath`

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.