

∞ Erewhon-Math ∞

Daniel Flipo
daniel.flipo@free.fr

15th November 2025

1 What is Erewhon-Math?

Erewhon-Math is an Utopia based OpenType maths font. The maths symbols and Greek letters are borrowed or derived from Michel Bovani's *Fourier-GUTenberg*, Latin letters and digits are borrowed from Michael Shape's Erewhon font.

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

It is meant to be used with Utopia based OpenType text fonts like Erewhon. For *Fourier-GUTenberg* users who want to switch to LuaLaTeX or XeLaTeX, the file `fourier-otf.sty` can be used as a replacement of `fourier.sty`.

Please note that the current version (0.72) 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 Erewhon-Math would be:

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

this loads Erewhon-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{erewhon}3 % rm
\setsansfont{Cabin}[Scale=MatchLowercase] % sf
\setmonofont{Inconsolatazi4}[Scale=MatchLowercase] % tt
```

otherwise you would get Latin Modern for text fonts.

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

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

```
\usepackage[ options4 ]{fourier-otf}
```

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

1. it loads `realscripts.sty` for better superscripts and footnote calls unless option `fakedscripts` has been activated;
2. it redefines the `\TeX`, `\LaTeX` and `\LaTeXe` logos to fit the Erewhon fonts unless option `no-logos` has been activated;
3. it loads `fourier-orns.sty`, providing many text ornaments;
4. it checks at `\begin{document}` if packages `amssymb` or `latexsym` are loaded and issues warnings in case they are;
5. it provides aliases for glyphs named differently in Unicode, so that `latexsym` or AMS names are also available;
6. it defines specific maths characters like `\Bbbbackslash` (\backslash), `\varempyset` (\emptyset), `\parallelslant` ($//$), `\shortparallelslant` ($\#$), etc.;
7. 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, `fourier-otf.sty` provides two options `no-text` and `Scale=<decimal>` meant to be used to load the Erewhon-Math font together with roman text fonts other than Erewhon, while keeping the advantages 1. to 5. pointed in the preceding list, f.i. `\usepackage[no-text,Scale=0.98]{fourier-otf}`

Please note that the fonts loaded by `fourier-otf.sty` are about 2% larger than those loaded by `fourier.sty`: the Erewhon text fonts are based on Utopia reduced à 94%, while the `fourier` fonts are based on Utopia reduced à 92%. Option `Scale=0.98` can be used to mimic the output produced by the `fourier.sty` package.

Option `no-text` can also be useful if Erewhon is to be loaded with specific options, f.i. `\usepackage[no-text]{fourier-otf}`
`\setmainfont{erewhon}[RawFeature=+onum;+ss01]`

³Erewhon (with capital E) is fine with LuaTeX but would fail with XeTeX (`erewhon.fontspec` not found).

⁴Possible *options* are `loose`, `no-text`, `fakedscripts`, `Scale=` or any of the options described in sections 3.1 to 3.4.

3 What is provided?

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

Sans-serif, typewriter and fraktur styles are borrowed from Latin Modern fonts. See in section 3.6 on page 9 how to choose from other maths fonts for these styles.

A full list of available glyphs is shown in file unimath-erewhon.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>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=TeX</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=upright</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$

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, they can be added to the `\setmathfont` call as well⁵, for example:

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

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

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

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

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

Erewhon-Math provides fourteen “Character Variants” options, listed on table 3, to choose between different glyphs for Greek characters and some others. Alternative calligraphic capitals have been added for E, Q and T in version 0.50.

Table 3: Character variants.

	Default	Variant	Name
cv00	0	0	0
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>
cv11	∂	∂	<code>\partial</code>
cv20	\mathcal{E}	\mathcal{E}	<code>\symcal{E}</code>
cv21	\mathcal{Q}	\mathcal{Q}	<code>\symcal{Q}</code>
cv22	\mathcal{T}	\mathcal{T}	<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{Erewhon-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 about `\hbar` (v0.43): `unicode-math` defines `\hbar` as `\slash` (U+210F) while `amsmath` provides two different glyphs (italic *h* with horizontal or diagonal stroke). `erewhon-math` now follows `unicode-math`; the italic *h* with horizontal stroke can be printed using `\slash` or `\hbar` together with character variant `cv01` or with `\mitbar` (replacement for AMS' command `\hbar`).

3.3 Stylistic sets

Erewhon-Math provides six “Stylistic Sets” options to choose between different glyphs for families of maths symbols.

StylisticSet=1, alias⁶ **Style=mathcal**, forces commands `\mathcal{}` and `\mathscr{}` to print *ABC* instead of *Aℬℳ* (the default). Please note that `unicode-math` recommends to use `\symcal` and `\symscr` instead of `\mathcal` and `\mathscr`.

If you want to use both commands `\mathcal{ABC}` and `\mathscr{ABC}` to print *ABC* and *Aℬℳ* respectively, you can use `unicode-math`'s option `range` this way:

```
\setmathfont{Erewhon-Math}[your options]
```

```
\setmathfont{Erewhon-Math}[range={cal,bfcal},StylisticSet=1]
```

Both lines are mandatory: the first one loads Erewhon-Math while the second one modifies `\mathcal{}` command's output.

StylisticSet=3, alias⁶ **Style=upint**, converts integrals signs into their upright variants, see table 4.

Table 4: `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>\ointctrlockwise</code>
Default	\int	\int	\oint	\oint
Upright	\int	\int	\oint	\oint

StylisticSet=4, alias⁶ **Style=leqslant**, converts (large) inequalities into their slanted variants, see table 5a on the following page.

StylisticSet=5, alias⁶ **Style=smaller**, converts some symbols into their smaller variants, see table 5b on the next page.

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

Table 5: 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>\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>\gtreqqless</code>	\gtreqqless	\gtreqqless			
<code>\lesssim</code>	\lesssim	\lesssim			
<code>\gtrsim</code>	\gtrsim	\gtrsim			

StylisticSet=6, alias⁷ **Style=subsetneq**, converts some inclusion symbols, see table 6a.

StylisticSet=7, alias⁷ **Style=parallelslant**, converts “parallel” symbols into their slanted variants, see table 6b.

Table 6: 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>	\parallel	\parallel
<code>\supsetneqq</code>	\supsetneqq	\supsetneqq	<code>\nshortparallel</code>	\nparallel	\nparallel

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

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

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

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

instead of

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

⁷These Style aliases are provided by `fourier-otf.sty`.

3.4 Other font features

3.4.1 Oldstyle numbers

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

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

0123456789, **0123456789**

3.4.2 Delimiters' size

When switching from Type 1 to OpenType, Adobe has significantly increased⁸ the sizes of Utopia's delimiters '()', '[]' and '{}'. Erewhon-Math has been built from Fourier (Utopia's Type 1 type faces⁹), while the Erewhon text fonts have been built from the OpenType version; the difference was visible on examples like ($P(n)$) which has been typeset as $(P(n))$ by versions up to 0.53.

This glitch is now fixed, you get now $(P(n))$, $[P[n]]$, $\{P\{n\}\}$. An option `Style=smalldelim` (+ss09), which can be passed either to the `fourier-otf` package or to the `\setmathfont` command, has been added in version 0.54 for backward compatibility: it reverts to the former behaviour.

3.5 Standard LaTeX math commands

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

Various wide accents are also supported:

☞ `\wideoverbar` and `\mathunderbar`¹⁰

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

☞ `\widehat` and `\widetilde`

\hat{x} \hat{xx} \hat{xxx} \hat{xxxx} \hat{xxxxx} \hat{xxxxxx} \tilde{x} \tilde{xx} \tilde{xxx} \tilde{xxxx} \tilde{xxxxx} \tilde{xxxxxx}

☞ `\widecheck` and `\widebreve`

\check{x} \check{xxx} \check{xxxx} \check{x} \breve{xxx} \breve{xxxx}

⁸Actually +21% both in height and thickness!

⁹Scaled at 94% instead of 92% to match Erewhon text fonts.

¹⁰`\overline` and `\underline` are not font related, they are based on `\rule`.

☞ `\overparen` and `\underparen`

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

$$\underparen{x} \quad \underparen{xz} \quad \underparen{xyz} \quad \underparen{x+z}^2 \quad \underparen{a+b+\dots+z}^{26}$$

☞ `\overbrace` and `\underbrace`

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

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

☞ `\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`

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

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

☞ `\overrightarrowtharpoonup` and `\overleftarrowtharpoonup`

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

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

☞ `\underrightarrow` and `\underleftarrow`

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

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

☞ `\underrightharpoonup` and `\underleftarrowtharpoonupdown`

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

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

☞ Finally `\widearc` and `\overrightarrowarc` (loading `fourier-otf.sty` is required)

$$\widearc{AMB} \quad \overrightarrowarc{AMB}$$

☞ Typewriter alphabet is borrowed from Latin Modern (`\symtt` command):

```
ABCDEFGHIJKLMN O PQRSTUVWXYZabcdefghijklmnopqrstu vwxyz
```

but it can be borrowed from another Math font, i.e.

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

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

All others have to be borrowed from another maths font if needed, i.e.

```
\setmathfont{Garamond-Math.otf}[range=\symscr/latin,12
Scale=MatchLowercase]
\symscr{a}\symscr{b}\symscr{c} ... \symscr{z}$
abcdefghijklmnopqrstu vwxyz
```

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

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

3.8 Missing symbols

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

3.9 Fourier ornaments

When loaded by `\usepackage{fourier-otf}`, Erewhon-Math loads `fourier-orns.sty` which provides all logos and ornaments available in Fourier-GUTenberg.

¹²/latin: lowercase (latin) only; /Latin: uppercase (latin) only.

`fourier-orns.sty` as of v2.x automatically fetches its glyphs in a specific OpenType font with LuaTeX or XeTeX engines and from a Type 1 font otherwise (pdfTeX).

- ☞ A variant of the euro symbol: `\eurologo` €, €, €, €.
- ☞ Two “starred” bullets: `\starredbullet` †, `\decosix` †.
- ☞ Decos and logos: `\warning` ⚠, `\noway` ☹, `\caution` ⚠, `\bomb` 💣, `\decoone` ✖, `\decotwo` ☹, `\decothreeleft` ⚡, `\decothreeright` ⚡, `\decofourleft` ⚡, `\decofourright` ⚡, `\floweroneleft` 🌸, `\floweroneright` 🌸, `\lefthand` 🖐, `\righthand` 🖐, `\textxswup` ✂, `\textxswdown` ✂.
- ☞ Smileys: `\grimace` 😬, `\textthing` †.
- ☞ Leaves: `\aldineleft` 🌿, `\aldineright` 🌿, `\aldine` 🌿, `\aldinesmall` 🌿, `\leafleft` 🌿, `\leafright` 🌿, `\leafNE` 🌿, `\leafNW` 🌿, `\leafSE` 🌿, `\leafSW` 🌿.
- ☞ Pilcrows: `\oldpilcrowone` ¶, `\oldpilcrowtwo` ¶, `\oldpilcrowthree` ¶, `\oldpilcrowfour` ¶, `\oldpilcrowfive` ¶, `\oldpilcrowsix` ¶.

All these logos and ornaments are also available as described in the `fournier-orns` documentation: for instance, you could type `{\FourierOrns E 2 F}` to get 🌸 ⚠ 🌸.

Finally, some symbols are also provided in maths mode, with other names:

- ☞ `forbidden` (☹), `beware` (⚠), `boom` (💣),
- ☞ `thething` (†) is a *QED symbol* for a false proof. Of course, you don’t need it!
- ☞ `xswordsup` (✂) and `xswordsdown` (✂) may be used as tags for a debated statement, or for anything else.

4 Acknowledgements

All glyphs in Erewhon-Math are borrowed or derived either from Erewhon fonts or Fourier-GUTenberg package. Many thanks to Michael Sharpe and Michel Bovani for providing these.

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

Thanks to Oliver Natt for providing valuable feedback!

