



author Esger Renkema
 contact minim@elrenkema.nl

This package gives a simple and highly-configurable way of using unicode and OpenType mathematics with plain LuaTeX, making use of most of the latter engine's new capabilities in mathematical typesetting. Also included are proper settings and definitions for nearly all unicode mathematical characters, as well as a few shorthands and helper macros that seemed useful additions.

Load the package by saying `\input minim-math.tex`; this will set up luatex for using opentype mathematical fonts and unicode math input. It will not, however, select mathematical fonts for you. That you will have to do for yourself; see below for instructions.

Styles and alphabets

For some (mostly alphabetical) characters, multiple variants are available, e.g. $AAA\mathcal{A}\mathfrak{A}\mathcal{A}$. You can (locally) override the default style of these with `\mathstyle {style}` (equivalent to the old `\bf`, `\rm` etc.) or with one of the shorthands that apply the style to their argument only:

Shorthand	Value of <code>style</code>	Result
<code>\mup</code>	<code>up/rm</code>	ABC
<code>\mit</code>	<code>it</code>	ABC
<code>\mbf</code>	<code>bf</code>	ABC
<code>\mbfit</code>	<code>bfit</code>	ABC
<code>\mbb</code>	<code>bb</code>	$\mathbb{A}\mathbb{B}\mathbb{C}$
<code>\frac</code>	<code>frac</code>	$\mathfrak{A}\mathfrak{B}\mathfrak{C}$
<code>\bfrac</code>	<code>bfrac</code>	$\mathfrak{A}\mathfrak{B}\mathfrak{C}$
<code>\scr</code>	<code>cal/scr</code>	$\mathcal{A}\mathcal{B}\mathcal{C}$
<code>\bfscr</code>	<code>bfscr</code>	$\mathcal{A}\mathcal{B}\mathcal{C}$

Styles without shorthand are `sans/sf`, `sfit`, `sfbf`, `sfbfit`, `tt/mono` and finally the special value `clear` for using the default style. You can use the shorthands directly in sub- and superscripts: $v^{\scr F}$ will result in $v^{\mathcal{F}}$.

While math families are not used anymore for switching between styles (see below), you still can use `\fam` with the values 0, 1, 2, 4, 5, 6 or 7 for doing so. This means that plain tex's `\rm`, `\it`, `\cal`, `\sl`, `\bf` and `\tt` can still be used (at least in math mode).

Please note that `\mup` is not the right choice for upright multiletter symbols or operators: you should use `\mord` or `\mop` instead (see near the end of this chapter). For nonmathematical text, you should use `\text` instead of `\mup`.

The default properties of characters can be set with one of the following three commands:

```
\mathmap {character list} {style}
\mathclass {character list} {class}
\mathfam {character list} nr
```

There, `style` is one of the above and `class` is the name of a class as below. Finally, the `character list` should be a comma-separated list with elements of one of the following forms:

1. a list of characters, like `abc` or `\partial` or \mathbb{R} ;
2. a character range, like ``A-`Z`, `65-90` or `"41-"5A`;
3. one of the alphabets `[Ll]atin`, `[Gg]reek`, or `digits`;
4. one of the style groups `bold`, `boldgreek`, `sans`, `sansgreek`, `mono`, `blackboard`, `fraktur` or `script`;
5. the name of a math class: `ord`, `op`, `bin`, `rel`, `fence`, `open`, `close` or `punct`.

Note that unicode is somewhat irregular in its encoding of mathematical letters; this is taken into account when using ranges as under (2) above. Thus, `\mscra-\mscrz` really gives you all lowercase script characters, despite e.g. *e* being well outside that range.

The default style settings are `\mathmap {latin, greek, Latin}{it}`. Since the math family setting is not used anymore for selecting different styles, the default family of every symbol is zero. Instead, you can use `\mathfam` for mixing fonts (see below). The `class` option to `\mathclass` should be one of the names under 5.

Character variants

You can change the default appearance of several greek characters with `\use-mathvariant {chars}` or `\usemathdefault {chars}`, where `chars` is a list of normal greek characters. As in unicode but against tex's tradition, the variants are $\varepsilon\theta\Theta\kappa\rho\varphi\varepsilon\theta\Theta\kappa\rho\varphi\epsilon\vartheta\Theta\kappa\rho\phi$ and the defaults $\varepsilon\theta\Theta\kappa\rho\varphi$. The macros `\varepsilonpsilon` etc. have been updated to reflect the unicode variants.

The appearance of root symbols can be set with `\closedroots` ($\sqrt{2}$) and `\normalroots` ($\sqrt[2]{2}$, the default).

Say `\unicodedots` to use the unicode dots characters (`...;⋯`) and `\traditionaldots` to construct these characters from periods (`...;⋯`, the default). Both settings affect the meaning of both the actual characters and the `\xdots` macros ($x \in \{l, v, c, a, d\}$ $x \in \{l, v, c, a, d\}$ $x \in \{1, v, c, a, d\}$). Unlike in traditional plain tex, the traditional dots are available in script sizes, too.

Say `\decimalcomma` and have commas appear as 1,2 instead of 1.2 (`\nodedecimalcomma` restores the default). Say `\smartdecimalcomma` for a comma that only acts as punctuation when not immediately followed by a digit. The explicit `\comma`, like `\colon`, will always be punctuation.

The behaviour of limits on integral signs can be set by redefining `\intlimits` (the default is `\let \intlimits = \nolimits`).

If you want to change the meaning (inside math mode) altogether for a single character, you can use the commands `\mathdef` and `\mathlet`. For example, by default, you can use the letter `h` for the reduced planck constant \hbar ; this has been made so with `\mathdef h {\hbar}` (you could also have said `\mathlet h \hbar`).

Setting up fonts

The minimum you need do to set up a mathematical font is this:

```
\font\tenmath
  {Latin Modern Math:mode=base;script=math;ssty=0;} at 10pt
\font\tenmaths
  {Latin Modern Math:mode=base;script=math;ssty=1;} at 7pt
```

```

\font\tenmathss
      {Latin Modern Math:mode=base;script=math;ssty=2;} at 5pt
\textfont      0 = \tenmath
\scriptfont    0 = \tenmaths
\scriptscriptfont 0 = \tenmathss

```

Note that you only have to set up the font for a single family: opentype mathematical fonts typically contain all necessary variants of all mathematical characters. Therefore, the `\fam` setting has been made a no-op (use `\setfam` if you really need the old primitive) and the default family of all symbols has been set to zero.

As mentioned above, you can still change the family number of specific characters and this allows you to mix mathematical fonts. For instance, if you dislike the current blackboard bold characters, just assign a second font to family 1 and say `\mathfam {blackboard} 1`. Less useful are the parameters `\accentfam`, `\radicalfam` and `\extensiblefam` that control the family of all accents, radicals and extensibles.

Do note that various spacing constants are set according to the *last* math family that is assigned to. Therefore, you should assign your main math font to a family after all others.

Shorthands and additions

You can use `\text` for adding nonmathematical text to your equations. It will behave well in sub- and superscripts: `\text{word}^{\text{word}}` gives $\text{word}^{\text{word}}$. By default, the font used is the normal mathematical font. You can change this by setting the `\textfam` parameter to some nonzero value and assigning a different font to that family (see above). You probably want to do this, since most commonly-used mathematical fonts do not include a normal kerning table.

All the usual arrows can be made extensible by prefacing them with an `x`, including `\xmapsto` and `\xmapsfrom`. Alternatively, you can use `\rightarrow` etc. as shorthands. Additionally, you can use the following:

Shorthand	Result
<code>\bra x, \ket y</code>	$\langle x , y\rangle$
<code>\braket x y</code>	$\langle x y\rangle$
<code>\norm x, \Norm x</code>	$ x , \ x\ $
<code>x \stackrel{?}{=} y</code>	$x \stackrel{?}{=} y$
<code>x \stackbin a+ y</code>	$x \stackbin a+ y$
<code>f\inv</code>	f^{-1} (cf. f^{-1})
<code>a \xrightarrow[down]{up} b</code>	$b \xrightarrow[down]{up} b$
<code>a \xrightarrow[down]{up} b</code>	$b \xrightarrow[down]{up} b$
<code>\frac12, \tfrac12, \dfrac12</code>	$\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$

Also new are the operators `\Tr`, `\tr`, `\Span`, `\GL`, `\SL`, `\SU`, `\U`, `\SO`, `\O`, `\Sp`, `\im`, `\End`, `\Aut`, `\Dom` and `\Codom`. You can define new operators with `\newmathop` and `\newlargemathop`: `\newmathop{op}` will define the new operator `\op`. For

single use of an upright symbol, operator or large operator you can use `\mord`, `\mop` and `\mlop`. The difference between `\mord` and `\mup` is that `\mord` also applies the correct symbol spacing.

The accents `\overbrace`, `\underbracket` etc. allow a label between square brackets: `$$\underbrace{[=1]{(x^2+y^2)}}{=1}$$` gives

$$\underbrace{(x^2 + y^2)}_{=1}$$

Finally, the following (entirely optional) alternative to using dollar signs is provided, which also offers slight improvements in the spacing of displayed equations:

	<code>\[... \]</code>	inline mathematics
<code>\display</code>	<code>\[... \]</code>	display mathematics
<code>\displaynr {nr}</code>	<code>\[... \]</code>	numbered display mathematics
<code>\displaynr</code>	<code>\[... \]</code>	automatically numbered display mathematics

The automatic display numbering uses the count `\equationnumber` and the token list `\setequationnumber` internally. All displays created this way can be made left-aligned by saying `\leftdisplaystrue`.

Best practices

The following remarks on mathematical typesetting have no relation to the contents of this package; I have included them because I find them hard to remember.

1. `\eqalign` gives a vertically centered box and can occur many times in an equation, while `\eqalignno` and `\leqalignno` span whole lines (put the equation numbers in the third column). All assume the relation (or operator) appears at the right hand side of the ampersand.
2. The command `\displaylines` can only have one column that spans the whole line (and you will have to add the equation number by hand).
3. Further alignment commands are `\cases`, `\matrix`, `\pmatrix` (with parentheses) and `\bordermatrix` (includes labels for lines and columns).
4. Finetuning alignments can be done with `\smash`, `\phantom`, `\hphantom` and `\vphantom`.
5. Small matrices like $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ can be made by misusing `\choose` or `\atop`.
6. If you start a line with a binary operator, put a `{}` before it: this way, tex recognises it as such.
7. Thin spaces (`\,`) should be inserted: before dx , before units, after factorials and after `\dots` if those are followed by a closing parenthesis.
8. Whether the differential operator should be set upright or not is as of yet an open question in mathematics.
9. You should prefer `\bigr` and `\bigl` etc. over `\big`, `\Big`, `\bigg` and `\Bigg`.
10. An overview of mathematical symbols, with control sequences and their availability in different fonts, can be found in `unimath-symbols.pdf`, which is part of the unicode-math package.

Licence

This package may be distributed under the terms of the European Union Public Licence (EURL) version 1.2 or later. An english version of this licence has been included as an attachment to this file; copies in other languages can be obtained at

<https://joinup.ec.europa.eu/collection/eupl/eupl-text-eupl-12>