The lua-unicode-math package*

Marcel Krüger
tex@2krueger.de
https://github.com/zauguin/lua-unicode-math

January 1, 2026

Modern fonts are usually provided in OpenType format and are designed for Unicode based input. For mathematical fonts this usually means the use of fonts with an OpenType MATH table: Fonts containing special metadata needed to make them usable in a mathematical context.

In LuaTeX such fonts have traditionally been loaded with the unicode-math package. While this works, is very flexible and allows to use the same document in XeTeX and LuaTeX it has performance issues and it sometimes has unexpected interactions with the use of math versions. The lua-unicode-math is a specific LuaLaTeX specific alternative which aims for higher performance and better integration with native LuaTeX features.

1 Usage instructions

1.1 Font packages

For most Opentype the recommended way to load them with lua-unicode-math is to use a dedicated package. Currently the following packages are shipped with lua-unicode-math:

Font	Package
Latin Modern Math	lum-lmodern
New Computer Modern Math	lum-newcomputermodern
STIX2	lum-stix2
XITS	lum-xits
TeX Gyre Pagella Math	lum-pagella
TeX Gyre DejaVu Math	lum-dejavu
TeX Gyre Bonum Math	lum-bonum
TeX Gyre Schola Math	lum-schola
TeX Gyre Termes Math	lum-termes
Fira Math	lum-fira
GFS Neohellenic Math	lum-gfsneohellenic
Erewhon Math	lum-erewhon
XCharter Math	lum-xcharter
Concrete Math	lum-concrete

*This document corresponds to lua-unicode-math v0.3, dated 2026-01-01.

1.2 Loading fonts by name

If you want to use a custom font, you can load fontspec and lua-unicode-math using

```
\usepackage{fontspec,lua-unicode-math}
```

This will load `Latin Modern Math` by default. Another math font can be loaded using `\setmathfont` using the same options as fontspec's `\newfontfamily`. For example, you can use to to configure the current math font using

```
\setmathfont[AutoFakeBold=1]{Latin Modern Math}
```

1.3 Writing maths

There are two ways of entering math: You can directly input Unicode math symbols or use regular LaTeX commands for symbols. All Unicode symbols are supported with the same commands as in unicode-math. For a full list see `texdoc unimath-symbols`.

2 Implementation

```
\ProvidesExplPackage
  {lua-unicode-math}
  {2026-01-01}
  {0.3}
  {Opentype Math support for LuaLaTeX}

⟨@@=l_uni_math⟩
\int_new:N \g__l_uni_math_font_count_int
\tl_new:N \l__l_uni_math_main_family_tl
\tl_new:N \l__l_uni_math_script_family_tl
\tl_new:N \l__l_uni_math_scriptscript_family_tl

\cs_generate_variant:Nn \tl_if_eq:nnT {o}

\msg_new:nnn { l_uni_math } { unicode-math-suppressed } {
  You~tried~to~load~both~lua-unicode-math~and~unicode-math~
  in~the~same~document.~This~is~not~supported,~unicode-math~
  will~be~suppressed.~There~is~a~good~chance~that~this~will~
  break~your~document.~Change~your~document~to~only~use~lua-unicode-math~
  so~solve~this.
}
\msg_new:nnn { l_uni_math } { unicode-math-loaded } {
  You~tried~to~load~lua-unicode-math~while~unicode-math~
  was~already~loaded.~This~does~not~work.~Please~avoid~loading~
  unicode-math.~If~that~is~not~possible~and~you~are~feeling~adventurous~
  you~can~try~loading~the~lua-unicode-math~package~at~the~beginning~
  of~your~document~instead~to~suppress~unicode-math.
}
\disable@package@load{unicode-math} {
  \msg_warning:nn { l_uni_math } {unicode-math-incompatible }
}
\IfPackageLoadedTF {unicode-math} {
  \msg_critical:nn { l_uni_math } {unicode-math-loaded }
} {}
```

```
\cs_if_exist:NF \DeclareMathScriptfontMapping {
  \cs_new:Npn \DeclareMathScriptfontMapping #1 #2 #3 #4 #5 #6 {
    \cs_set:cpx { scriptfont__l_uni_math #1 / #2 } { #3 / #4 }
    \cs_set:cpx { scriptscriptfont__l_uni_math #1 / #2 } { #5 / #6 }
  }
  \cs_set:Npn \__l_uni_math_family_variant:nw #1 #2/#3/ {
    \ifcsname #1__l_uni_math#2/#3 \endcsname
      \lastnamedcs
    \else
      #2/#3
    \fi
    /
  }

% Hook into the kernel to allow different families in scriptfonts
  \cs_set:Npn \getanddefine@fonts #1 #2 {
    \xdef\font@name{\csname \string#2/\tf@size\endcsname}
    \pickup@font \let\textfont@name\font@name
    \xdef\font@name{\csname \exp_last_unbraced:Nno \__l_uni_math_family_variant:nw {scriptfon
    \pickup@font \let\scriptfont@name\font@name
    \xdef\font@name{\csname \exp_last_unbraced:Nno \__l_uni_math_family_variant:nw {scriptscr
    \pickup@font
    \edef\math@fonts{\math@fonts
                  \textfont#1\textfont@name
                  \scriptfont#1\scriptfont@name
                  \scriptscriptfont#1\font@name}
  }
}

\hook_gput_code:nnn { package/fontspec/after } {.} {
  \bool_gset_false:N \g__fontspec_math_bool

  \NewDocumentCommand \setmathfont { O{} m O{} } {
    \int_incr:N \g__l_uni_math_font_count_int
    \exp_args:Nc \newfontfamily
      { g__l_uni_math_font_ \int_use:N \g__l_uni_math_font_count_int _text_font }
      { #2 }
      [ #1, #3, Script = Math, Renderer = Base ]
    \tl_set_eq:NN \l__l_uni_math_main_family_tl \l_fontspec_family_tl

    \exp_args:Nc \newfontfamily
      { g__l_uni_math_font_ \int_use:N \g__l_uni_math_font_count_int  _script_font }
      { #2 }
      [ #1, #3, Script = Math, Renderer = Base, Style = MathScript ]
    \tl_set_eq:NN \l__l_uni_math_script_family_tl \l_fontspec_family_tl

    \exp_args:Nc \newfontfamily
      { g__l_uni_math_font_ \int_use:N \g__l_uni_math_font_count_int  _scriptscript_font }
      { #2 }
      [ #1, #3, Script = Math, Renderer = Base, Style = MathScriptScript ]
    \tl_set_eq:NN \l__l_uni_math_scriptscript_family_tl \l_fontspec_family_tl

```

```
  \DeclareMathScriptfontMapping {TU} {\l__l_uni_math_main_family_tl} {TU} {\l__l_uni_math_s

  \exp_args:NnnV \DeclareSymbolFont {lummain} {TU} \l__l_uni_math_main_family_tl {m} {n}
  \exp_args:NnnnV \SetSymbolFont {lummain} {bold} {TU} \l__l_uni_math_main_family_tl {b} {n
}

\cs_set:Nn \__fontspec_setmainfont_hook:nn
  {
    \tl_if_eq:onT {\g__fontspec_mathrm_tl} {\rmdefault}
      {
        \fontspec_gset_family:Nnn \g__fontspec_mathrm_tl {Renderer=Basic,#1} {#2}
        \__fontspec_setmathrm_hook:nn {#1} {#2}
      }
  }
\cs_set:Nn \__fontspec_setsansfont_hook:nn
  {
    \tl_if_eq:onT {\g__fontspec_mathsf_tl} {\sfdefault}
      {
        \fontspec_gset_family:Nnn \g__fontspec_mathsf_tl {Renderer=Basic,#1} {#2}
        \__fontspec_setmathsf_hook:nn {#1} {#2}
      }
  }
\cs_set:Nn \__fontspec_setmonofont_hook:nn
  {
    \tl_if_eq:onT {\g__fontspec_mathtt_tl} {\ttdefault}
      {
        \fontspec_gset_family:Nnn \g__fontspec_mathtt_tl {Renderer=Basic,#1} {#2}
        \__fontspec_setmathtt_hook:nn {#1} {#2}
      }
  }
\cs_set:Nn \__fontspec_setmathrm_hook:nn
  {
    \SetMathAlphabet \mathrm { normal } \g_fontspec_encoding_tl \g__fontspec_mathrm_tl { \m
    \SetMathAlphabet \mathit { normal } \g_fontspec_encoding_tl \g__fontspec_mathrm_tl { \m
    \SetMathAlphabet \mathbf { normal } \g_fontspec_encoding_tl \g__fontspec_mathrm_tl { \b
  }
\cs_set:Nn \__fontspec_setboldmathrm_hook:nn
  {
    \SetMathAlphabet \mathrm { bold } \g_fontspec_encoding_tl \g__fontspec_bfmathrm_tl { \m
    \SetMathAlphabet \mathit { bold } \g_fontspec_encoding_tl \g__fontspec_bfmathrm_tl { \m
    \SetMathAlphabet \mathbf { bold } \g_fontspec_encoding_tl \g__fontspec_bfmathrm_tl { \b
  }
\cs_set:Nn \__fontspec_setmathsf_hook:nn
  {
    \SetMathAlphabet \mathsf { normal } \g_fontspec_encoding_tl \g__fontspec_mathsf_tl { \m
    \SetMathAlphabet \mathsf { bold } \g_fontspec_encoding_tl \g__fontspec_mathsf_tl { \bfs
  }
\cs_set:Nn \__fontspec_setmathtt_hook:nn
  {
    \SetMathAlphabet \mathtt { normal } \g_fontspec_encoding_tl \g__fontspec_mathtt_tl { \m
    \SetMathAlphabet \mathtt { bold } \g_fontspec_encoding_tl \g__fontspec_mathtt_tl { \bfs
  }
%
\__fontspec_setmathrm_hook:nn {} {}
```

```
  \__fontspec_setmathsf_hook:nn {} {}
  \__fontspec_setmathtt_hook:nn {} {}
}

\cs_set_protected:Npn \operator@font {
  \@fontswitch { \font@warning{Math~mode~required~for~\string\operator@font.} } { \mathtextrm
}

\DeclareSymbolFont {lummain} {TU} {lmm} {m} {n}
\SetSymbolFont {lummain} {bold} {TU} {lmm} {b} {n}

\newattribute \mathfamattr

\lua_load_module:n { lua-unicode-math }

%   up, it, bfup, bfit, sfup, sfit, bfsfup, bfsfit, bfsf, % 9
%   tt, bb, bbit, scr, bfscr, cal, bfcal, frak, bffrak, % 9
%   normal, literal, sf, bf, % 4

%   bf, % 9
%   bbit, % TODO: sigh. I really don't want this one. The only affected characters are    .
%   scr, bfscr, % 9 % TODO: Think about these

\prop_set_from_keyval:Nn \l_tmpa_prop {
  up = 0, bfup = 1, it = 2, bfit = 3,
  sfup = 4, bfsfup = 5, sfit = 6, bfsfit = 7,
  cal = 8, bfcal = 9,
  frak = 12, bffrak = 13,
  tt = 16,
  bb = 20,
  bf = 1024,
  normal = -"7FFFFFFF,
  literal = -1,
}
\cs_set_eq:NN \mathup \mathrm
\prop_map_inline:Nn \l_tmpa_prop {
  \cs_new_protected:cpn { sym #1 } ##1 {
    {
      \mathfamattr = #2 \scan_stop:
      ##1
    }
  }
  \cs_if_exist:cTF { math #1 } {
    \cs_set_eq:cc { mathtext #1 } { math #1 }
  } {
    \cs_set_eq:cc { math #1 } { sym #1 }
  }
}
\cs_set_eq:NN \mathtextrm \mathrm
\cs_set_eq:NN \symrm \symup
```

In unicode-math there is a package option to use italic or upright letters for `\symsf` and `\symbfsf`. Depending on the setting these become aliases for `\symsfup` and `\symbfsfup` or `\symsfit` and `\symbfsfit`. Since we do not want package option, we always use

upright letters and you ca redefine \symsf and \symbfsf if you want to change it.

```
\cs_set_eq:NN \mathtextsf \mathsf
\cs_set_eq:NN \symsf \symsfup
\cs_set_eq:NN \symbfsf \symbfsfup

\clist_map_inline:nn { cal, calbf, frak, frakbf, bb } {
  \cs_set_eq:cc { math #1 } { sym #1 }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathord :nn } #1 #2 {
  \cs_set:Npx #1 {
    \char_generate:nn {#2} {12}
  }
}
\tl_map_inline:nn {\mathbin \mathclose \mathpunct \mathrel} {
  \cs_new_eq:cc
    { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N #1 :nn }
    { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathord :nn }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathop :nn } #1 #2 {
  \exp_args:Nc \Umathchardef { \cs_to_str:N #1 op } 1~\symlummain #2~
  \cs_set_eq:cN { \cs_to_str:N #1 oplimits } \scan_stop:
  \cs_set:Npx #1 {
    \char_generate:nn {#2} {12}
  }
  \mathcode #2 = "8000~
  \cs_set:cpx { \char_generate:nn {"FFFF} {12} \char_generate:nn {#2} {12} } {
    \use:c { \cs_to_str:N #1 op }
    \use:c { \cs_to_str:N #1 oplimits }
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathopen :nn } #1 #2 {
  \token_if_eq_meaning:NNTF #1 \sqrt {
    \cs_set:Npx \sqrtsign {
      \Uradical \symlummain #2~
    }
    \cs_set:Npx \root ##1 \of {
      \Uroot \symlummain #2~ { ##1 }
    }
  }{
    \cs_set:Npx #1 {
      \char_generate:nn {#2} {12}
    }
  }
}

\group_begin:
\cs_set:Npn \l_tmp_cs:n #1 {
  \group_end:

  \cs_new_protected:Npn \__l_uni_math__check_mup_helper:w ##1 #1 ##2 \q_mark ##3 ##4 \q_stop
    ##3 {##2}
```

```
  }

  \cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathalpha :nn } ##1 ##2 {
    \cs_set:Npx ##1 {
      \char_generate:nn {##2} {12}
    }
    \exp_after:wN \__l_uni_math__check_mup_helper:w \token_to_str:N ##1 \q_mark \cs_set_eq:cN
  }
}

\exp_args:No \l_tmp_cs:n {
  \token_to_str:N \mup
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathfence :nn } #1 #2 {
  \cs_set:Npx #1 {
    \char_generate:nn {#2} {12}
  }
  \cs_set:cpx {l \cs_to_str:N #1} {
    \Udelimiter 4 ~ \symlummain #2 ~
  }
  \cs_set:cpx {r \cs_to_str:N #1} {
    \Udelimiter 5 ~ \symlummain #2 ~
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathaccent :nn } #1 #2 {
  \cs_set_protected:Npx #1 {
    \Umathaccent fixed 0 ~ \symlummain #2 ~
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathbotaccent :nn } #1 #2 {
  \cs_set:Npx #1 {
    \exp_not:N \PackageError{lua-unicode-math}{Unsupported~type~\token_to_str:N \mathbotaccen
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathaccentwide :nn } #1 #2 {
  \cs_set:Npx #1 {
    \Umathaccent 0 ~ \symlummain #2 ~
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathbotaccentwide :nn } #1 #2
  \cs_set:Npx #1 {
    \Umathaccent bottom 0 ~ \symlummain #2 ~
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathaccentoverlay :nn } #1 #2
  \cs_set:Npx #1 {
    \Umathaccent overlay 0 ~ \symlummain #2 ~
  }
```

```
  % \cs_set:Npx #1 {
  %   \exp_not:N \PackageError{lua-unicode-math}{Unsupported~type~\token_to_str:N \mathaccent
  % }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathover :nn } #1 #2 {
  \cs_set:Npx #1 {
    \exp_not:N \PackageError{lua-unicode-math}{Unsupported~type~\token_to_str:N \mathover}{}
  }
}

\cs_new:cpn { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N \mathunder :nn } #1 #2 {
  \cs_set:Npx #1 {
    \exp_not:N \PackageError{lua-unicode-math}{Unsupported~type~\token_to_str:N \mathunder}{}
  }
}

\cs_generate_variant:Nn \exp_args:Ne {c}
\cs_new:Npn \UnicodeMathSymbol #1 #2 #3 #4 {
  \use:c { __l_uni_math_UnicodeMathSymbol_ \token_to_str:N #3 :nn }
    {#2} {#1}
}
\input {unicode-math-table}
\cs_undefine:N \UnicodeMathSymbol

\cs_set_protected:Npn \triangle { \mathord { \bigtriangleup } }
\cs_set_protected:Npn \mathellipsis { \mathinner { \unicodeellipsis } }
\cs_set_protected:Npn \cdots { \mathinner { \unicodecdots } }

\clist_map_inline:nn {
  \to \rightarrow,
  \le \leq,
  \ge \geq,
  \neq \ne,
  \bigcirc \mdlgwhtcircle,
  \circ \vysmwhtcircle,
  \bullet \smblkcircle,
  \mathyen \yen,
  \mathsterling \sterling,
  \diamond \smwhtdiamond,
  \emptyset \varnothing,
  \hbar \hslash,
  \land \wedge,
  \lor \vee,
  \owns \ni,
  \gets \leftarrow,
  \mathring \ocirc,
  \lnot \neg,
  \longdivision \longdivisionsign,
  \backepsilon \upbackepsilon,
  \eth \matheth,
  \dotsb@ \cdots,
  \@cdots \cdots,
} {
```

```
  \cs_set_eq:NN #1
}

\cs_set_eq:NN \intoplimits \nolimits

\cs_set_protected:cpx { \char_generate:nn {"FFFF} {12} ' } {
  \prime_helper:w "2032~
}

\cs_set_protected:Npn \uproot #1 {
  \__l_uni_math_uproot:w #1 \scan_stop:
}

\cs_set_protected:Npn \leftroot #1 {
  \__l_uni_math_leftroot:w #1 \scan_stop:
}
```

Some fixes for amsmath: Since amsmath is defining `\leftroot`, `\uproot` and `\root` with non Unicode definitions, we need to hide our definitions and restore them afterwards. We define `\varGamma` to stop amsmath from trying to define greek letter variants.

```
\tl_const:Nn \c__l_uni_math_amsmath_cmds_tl {
  \uproot
  \leftroot
  \iint
  \iiint
  \iiiint
  \dddot
  \ddddot
  \overleftrightarrow
  \underrightarrow
  \underleftarrow
  \underleftrightarrow
  \hat
  \check
  \tilde
  \acute
  \grave
  \dot
  \ddot
  \breve
  \bar
  \vec
  \mathring
}
\tl_const:Nn \c__l_uni_math_amsmath_cmds_defined_tl {
  \root
  \int
  \oint
  \overrightarrow
  \overleftarrow
}
\hook_gput_code:nnn { package/amsmath/before } {.} {
  \tl_map_inline:Nn \c__l_uni_math_amsmath_cmds_tl {
    \cs_new_eq:cN { __l_uni_math_saved_ \cs_to_str:N #1 } #1
    \cs_undefine:N #1
```

```
  }
  \tl_map_inline:Nn \c__l_uni_math_amsmath_cmds_defined_tl {
    \cs_new_eq:cN { __l_uni_math_saved_ \cs_to_str:N #1 } #1
  }
  \cs_set:Npn \varGamma { \temporary_definition_do_not_use }
}
\hook_gput_code:nnn { package/amsmath/after } {.} {
  \tl_map_inline:Nn \c__l_uni_math_amsmath_cmds_tl {
    \cs_set_eq:Nc #1 { __l_uni_math_saved_ \cs_to_str:N #1 }
    \cs_undefine:c { __l_uni_math_saved_ \cs_to_str:N #1 }
  }
  \tl_map_inline:Nn \c__l_uni_math_amsmath_cmds_defined_tl {
    \cs_set_eq:Nc #1 { __l_uni_math_saved_ \cs_to_str:N #1 }
    \cs_undefine:c { __l_uni_math_saved_ \cs_to_str:N #1 }
  }
  \cs_undefine:N \varGamma
}
```