The \texttt{lualatex-math} package*

Philipp Stephani
p.stephani2@gmail.com
2022/01/01

Contents

1 Introduction 1

2 Interface 2

3 Implementation of the \LaTeX 2e package 2

\hspace{1em}3.1 Requirements ........................................ 2

\hspace{1em}3.2 Messages ........................................ 3

\hspace{1em}3.3 Initialization ....................................... 3

\hspace{1em}3.4 Patching ........................................ 3

\hspace{1em}3.5 \LaTeX 2e kernel .................................... 5

\hspace{1em}3.6 \texttt{amsmath} ..................................... 5

\hspace{1em}3.7 \texttt{mathtools} .................................. 8

\hspace{1em}3.8 \texttt{icomma} .................................... 9

4 Implementation of the Lua\LaTeX module 10

1 Introduction

\LaTeX brings major improvements to all areas of \TeX typesetting and programming. They are made available through new primitives or the embedded Lua interpreter, and combining them with existing \LaTeX 2e packages is not a task the average \LaTeX user should have to care about. Therefore a multitude of \LaTeX 2e packages have been written to bridge the gap between documents and the new features. The \texttt{lualatex-math} package focuses on the additional possibilities for mathematical typesetting. The most eminent of the new features is the ability to use Unicode and OpenType fonts, as provided by Will Robertson’s \texttt{unicode-math} package. However, there is a smaller group of changes unrelated to Unicode: these are to be dealt with in this package. While in principle most \TeX documents written for traditional engines should work just fine with Lua\LaTeX, there is a small number of breaking changes that require the attention of package authors. The \texttt{lualatex-math} package tries to fix some of the issues encountered while porting traditional macro packages to Lua\LaTeX.

The decision to write patches for existing macro packages should not be made lightly: monkey patching done by somebody different from the original package author ties the patching package to the implementation details of the patched functionality and breaks all rules of encapsulation. However, due to the lack of

*This document corresponds to \texttt{lualatex-math} v1.12, dated 2022/01/01.
alternatives, it has become an accepted way of providing new functionality in \LaTeX. To keep the negative impact as small as possible, the \texttt{lualatex-math} package patches only the \LaTeX\ 2\,\epsilon kernel and a small number of popular packages. In general, this package should be regarded as a temporary kludge that should be removed once the math-related packages are updated to be usable with Lua\TeX. By its very nature, the package is likely to cause problems; in such cases, please refer to the issue tracker\footnote{https://github.com/phst/lualatex-math/issues}.

2 Interface

The \texttt{lualatex-math} package can be loaded with \texttt{\usepackage} or \texttt{\RequirePackage}, as usual. It has no options and no public interface; the patching is always done when the package is loaded and cannot be controlled. As a matter of course, the \texttt{lualatex-math} package needs Lua\TeX\ to function; it will produce error messages and refuse to load under other engines and formats. The package depends on the expl3 bundle, the etoolbox package and the filehook package. The \texttt{lualatex-math} package is independent of the unicode-math package; the fixes provided here are valid for both Unicode and legacy math typesetting.

Currently patches for the \LaTeX\ 2\,\epsilon kernel and the amsmath, mathtools and icomma packages are provided. It is not relevant whether you load these packages before or after \texttt{lualatex-math}. They should work as expected (and ideally you shouldn’t notice anything), but if you load other packages that by themselves overwrite commands patched by this package, bad things may happen, as it is usual with Lua\TeX.

One user-visible change is that the new \texttt{\mathstyle} primitive should work in all cases after the \texttt{lualatex-math} package has been loaded, provided you use the high-level macros \texttt{\frac}, \texttt{\binom}, and \texttt{\genfrac}. The fraction-like \TeX\ primitives like \texttt{\over} or \texttt{\atopwithdelims} and the plain \TeX\ leftovers like \texttt{\brack} or \texttt{\choose} cannot be patched, and you shouldn’t use them.

3 Implementation of the \LaTeX\ 2\,\epsilon package

3.1 Requirements

\begin{verbatim}
\begin{verbatim}
\mathstyle
\frac, \binom, \genfrac
\end{verbatim}
\end{verbatim}

Executing the exhaustive expansion of \texttt{\@\@\_restore\_catcode:N(character token)} restores the category code of the \texttt{(character token)} to its current value.

\begin{verbatim}
\begin{verbatim}
\cs_if_exist:NF \newluabytecode
{ \RequirePackage { lualatexbase } [ 2010/05/27 ] }
directlua{require("lualatex-math")}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\@@\_restore\_catcode:N
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\cs_new\_nopar:\N \@@\_restore\_catcode:N #1 { \char_set\_catcode:nn \{ \int\_eval:n \{ "#1" \} \} \char\_value\_catcode:nn \{ "#1" \} }
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\texttt{\@\@\_restore\_catcode:N(character token)}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\cs\_new\_nopar:\Nm \@@\_restore\_catcode:N #1 { \char\_set\_catcode:nn \{ \int\_eval:n \{ "#1" \} \} \char\_value\_catcode:nn \{ "#1" \} }
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\directlua{require("lualatex-math")}
\end{verbatim}
\end{verbatim}
\end{verbatim}

\footnote{https://github.com/phst/lualatex-math/issues}
We use the macro defined above to restore the category code of the dollar sign. There are packages that make the dollar sign active; hopefully they get loaded after the packages we are trying to patch.

\exp_args:Nx \AtEndOfPackage {
\@@_restore_catcode:N \$
}\char_set_catcode_math_toggle:N \$

3.2 Messages

\texttt{luatex-required} Issued when not running under \LaTeX{}.

\msg_new:nnn { lualatex-math } { luatex-required } {
I will stop loading now.
}\texttt{macro-expected} Issued when trying to patch a non-macro. The first argument must be the detokenized macro name.

\msg_new:nnn { lualatex-math } { macro-expected } {
I've expected that \#1 is a macro, but it isn't.
}\texttt{wrong-meaning} Issued when trying to patch a macro with an unexpected meaning. The first argument must be the detokenized macro name; the second argument must be the actual detokenized meaning; and the third argument must be the expected detokenized meaning.

\msg_new:nnn { lualatex-math } { wrong-meaning } {
I've expected \#1 to have the meaning \langle \#3 \rangle \langle \#2 \rangle.
}\texttt{patch-macro} Issued when a macro is patched. The first argument must be the detokenized macro name.

\msg_new:nnn { lualatex-math } { patch-macro } {
I'm going to patch macro \#1.

3.3 Initialization

Unless we are running under \LaTeX{}, we issue an error and quit immediately.

\sys_if_engine_luatex:F {
\msg_error:nn { lualatex-math } { luatex-required }
} \endinput

3.4 Patching

\@@_temp:w A scratch macro.

\cs_new_eq:NN \@@_temp:w \prg_do_nothing:

\@@_patch:NNnnn \@@_patch:NNnnn The auxiliary macro \@@_patch:NNnnn(command)(factory command)(parameter text)(expected replacement text)(new replacement text) tries to patch \langle command \rangle. If \langle command \rangle is undefined, do nothing. Otherwise it must be a macro with the given \langle parameter text \rangle and \langle expected replacement text \rangle, created by the
given \textit{factory command} or equivalent. In this case it will be overwritten using the 
\textit{parameter text} and the \textit{new replacement text}. Otherwise issue a warning and
don't overwrite.

40 \cs_new_protected_nopar:Npn \@@_patch:NNnnn #1 #2 #3 #4 #5 {
41 \cs_if_exist:NT #1 {
42 \token_if_macro:NTF #1 {
43 \group_begin:
44 #2 \@@_temp:w #3 { #4 }
45 \cs_if_eq:NNTF #1 \@@_temp:w {
46 \msg_info:nxn { lualatex-math } { patch-macro }
47 \{} \token_to_str:N #1 \}
48 \group_end:
49 #2 #1 #3 { #5 }
50 } {
51 \msg_warning:nnxxx { lualatex-math } { wrong-meaning }
52 \{} \token_to_str:N #1 \}
53 \{} \token_to_meaning:N #1 \}
54 \group_end:
55 } {
56 \msg_warning:nnx { lualatex-math } { macro-expected }
57 \{} \token_to_str:N #1 \}
58 \{} \token_to_meaning:N \@@_temp:w \}
59 \group_end:
60 }
61 }
62 \cs_generate_variant:Nn \@@_patch:NNnnn { c }

\@@_set_mathchar:NN
The macro \texttt{\@@_set_mathchar:NN(\textit{control sequence})(\textit{token})} defines the \textit{control sequence} as an extended mathematical character shorthand whose mathematical
code is given by the mathematical code of the character `(\textit{token}). We cannot use the
\texttt{\normalshape\textbackslash Umathcharnumdef} primitive here since we would then rely on the \texttt{\normalshape\textbackslash Umathcodenum}
primitive which is currently broken.\footnote{\url{http://tug.org/pipermail/luatex/2012-October/003794.html}}
63 \cs_new_protected_nopar:Npn \@@_set_mathchar:NN #1 #2 {
64 \Umathchardef #1
65 \lua_now:e { 
66 lualatex.math.print_class_fam_slot( \int_eval:n { `#2 } )
67 }
68 \scan_stop:
69 }

\@@ before_package:nn
\@@ after_package:nn
The macro \texttt{\@@ before_package:nn(\textit{package})(\textit{code})} executes the \textit{code} before
the \textit{package} is loaded. Accordingly, \texttt{\@@ after_package:nn(\textit{package})(\textit{code})} executes the \textit{code} after the \textit{package} is loaded. If the \textit{package} is already loaded,
nothing happens. We prefer using native \texttt{\normalshape\textbackslash LT2E} hooks if possible.
70 \@ifl@t@r \fmtversion { 2021/11/15 } { 
71 \cs_new_protected_nopar:Npn \@@ before_package:nn #1 #2 {
72 \AddToHook { package/#1/before } { #2 }
73 }
74 \cs_new_protected_nopar:Npn \@@ after_package:nn #1 #2 {
75 \AddToHook { package/#1/after } { #2 }
76 }
77 }
78 \@ifl@t@r \fmtversion { 2020/10/01 } { 
79 \cs_new_protected_nopar:Npn \@@ before_package:nn #1 #2 {
80 \AddToHook { package/before/#1 } { #2 }
81 }
The macro \@@_after_package_or_now:nn \langle package \rangle \{ \langle code \rangle \} executes the \langle code \rangle after the \langle package \rangle is loaded. If the \langle package \rangle is already loaded, the \langle code \rangle is executed immediately.

\frac Here we assume that nobody except amsmath redefines \frac. This is obviously not the case, but we ignore other packages (e.g., nath) for the moment. We only patch the \LaTeX \epsilon kernel definition if the amsmath package is not loaded; the corresponding patch for amsmath follows below. Since \frac is declared by \DeclareRobustCommand, we must patch the macro \frac for the \LaTeX \epsilon kernel.

To do: do we need the additional set of braces around \Ustack?

The popular amsmath package is subject to three \LaTeX \epsilon-related problems:

- The \mathcode primitive is used several times, which fails for Unicode math characters. \Umathcode should be used instead.
• Legacy font dimensions are used for constructing stacks in the `substack`
command and the `subarray` environment. This doesn’t work if a Unicode
math font is selected.

• The fraction commands `\frac` and `\genfrac` don’t use the `\Ustack`
primitive.

These problems have been fixed in version 2.17i of `amsmath`, so we don’t attempt to
patch it if that version is loaded.

\[\c_{\@\@ std minus mathcode_int}\]
\[\c_{\@\@ std equal mathcode_int}\]
These constants contain the standard \TeX mathematical codes for the minus and
the equal signs. We temporarily set the math codes to these constants before loading
the `amsmath` package so that it can request the legacy math code without error.

\[\int_{\@\@ std minus mathcode_int} { "2200 }\]
\[\int_{\@\@ std equal mathcode_int} { "303D }\]
\[\l_{\@\@ minus mathchar}\]
\[\l_{\@\@ equal mathchar}\]
These mathematical characters are saved before `amsmath` is loaded so that we
can temporarily assign the \TeX values to the mathematical codes of the minus
and equals signs. The `amsmath` package queries these codes, and if they represent
Unicode characters, the package loading will fail. If `amsmath` has already been
loaded, there is nothing we can do, therefore we use the non-starred version of
`\AtBeginOfPackageFile`.

\[\l_{\@\@ minus mathchar}\]
\[\l_{\@\@ equal mathchar}\]
Now we temporarily reset the mathematical codes.

\[\char_set_mathcode:nn { `\- } { \c_{\@\@ std minus mathcode_int } }\]
\[\char_set_mathcode:nn { `\= } { \c_{\@\@ std equal mathcode_int } }\]
\[\l_{\@\@ minus mathchar}\]
\[\l_{\@\@ equal mathchar}\]
The `amsmath` package defines the control sequences `\std@minus` and `\std@equal`
as mathematical character shorthands while loading, but uses our restored mathemati-
cal codes, which must be fixed.

\[\cs_set_eq:NN \std@minus \l_{\@\@ minus mathchar}\]
\[\cs_set_eq:NN \std@equal \l_{\@\@ equal mathchar}\]
Finally, we restore the original mathematical codes of the two signs.

\[\Umathcodenum `\- \l_{\@\@ minus mathchar}\]
\[\Umathcodenum `\= \l_{\@\@ equal mathchar}\]
\[\l_{\@\@ minus mathchar}\]
\[\l_{\@\@ equal mathchar}\]
All of the following fixes work even if `amsmath` is already loaded.

\[\begin{document}\textbf{amsmath} repeats the definition of `\std@minus` and `\std@equal`
at the beginning of the document, so we also have to patch the internal kernel macro
`\@begindocumenthook` which contains the hook code.

\[\@ifpackagelater { amsmath } { 2020/08/24 } { } { \mathchardef \std@minus \mathcode `\- \relax \textbf{amsmath}\}@after_package_or_now:nn { amsmath } { { \mathchardef \std@equal \mathcode `\= \relax \textbf{amsmath}\}@after_package_or_now:nn { amsmath } { } \textbf{amsmath}\}`
subarray  The `subarray` environment uses legacy font dimensions. We simply patch it to use Lua\TeX\ font parameters (and \LaTeX\3 expressions instead of TeX arithmetic). Since subscript arrays are conceptually vertical stacks, we use the sum of top and bottom shift for the default vertical baseline distance (`\baselineskip`) and the minimum vertical gap for stack for the minimum baseline distance (`\lineskip`).

```latex
\@ifpackagelater { amsmath } { 2020/09/23 } { } {
\@@_patch:NNnnn \subarray \cs_set:Npn { #1 } {
    \vcenter
    \bgroup
    \Let@
    \restore@math@cr
    \default@tag
    \baselineskip \fontdimen 10 \scriptfont \tw@
    \advance \baselineskip \fontdimen 12 \scriptfont \tw@
    ⟨@@=⟩
    \lineskip \thr@@ \fontdimen 8 \scriptfont \thr@@
    ⟨@@=lltxmath⟩
    \lineskiplimit \lineskip
    \ialign
    \bgroup
    \ifx c #1 \hfil \fi $ \m@th \scriptstyle ## $ \hfil
    \crcr
    } {\vcenter \c_group_begin_token \Let@
\restore@math@cr \default@tag \baselineskip \fontdimen 10 \scriptfont \tw@
    + \fontdimen 12 \scriptfont \tw@
    \lineskip \Umathstackvgap \scriptstyle
    \ialign
    \c_group_begin_token \token_if_eq_meaning:NNT c #1 { \hfil }
    \Ustartmath
    \m@th
    \scriptstyle
    \alignmark \alignmark
    \Ustopmath
    \hfil
    \crcr
\}}
\frac  Since `\frac` is declared by `\DeclareRobustCommand`, we must patch the macro `\frac_`.  

```
Generalized fractions are typeset by the \texttt{\textbackslash genfrac} command. Since \texttt{\textbackslash genfrac} is declared by \texttt{\DeclareRobustCommand}, we have to patch the macro \texttt{\textbackslash genfrac\_L}. 

\begin{verbatim}
\begingroup #1 \endgroup \@over #2
\genfrac \GenericOptionalParameters{\arg1}{\arg2}\OptionalArgument{\arg3}{\arg4}{\arg5}{\arg6}
\end{verbatim}

3.7 mathtools

\texttt{mathtools}' \texttt{\textbackslash cramped} command and others that make use of its internal version use a hack involving a null radical. \LaTeX{} has primitives for setting material in cramped mode, so we make use of them.

In newer versions of \texttt{mathtools}, this issue is fixed, in which case we skip the patch.

\begin{verbatim}
\MT_cramped_internal:Nn
\end{verbatim}

The macro \texttt{\MT\_cramped\_internal:Nn\{\textit{style}\}\{\textit{expression}\}} typesets the \textit{expression} in the cramped style corresponding to the given \textit{style} (\texttt{displaystyle} etc.);
all we have to do in LuaTeX is to select the correct primitive. Rewriting the user-level ·cramped· command and employing \textstyle would be possible as well, but we avoid this way since we want to patch only a single command.

\begin{verbatim}
\@after_package_or_now:nn { mathtools } {
\@ifpackagelater { mathtools } { 2021/03/28 } { } {
\@@_patch:NNnnn \MT_cramped_internal:Nn
\cs_set_nopar:Npn { #1 #2 } {
\setbox \z@ \hbox {$\m@th #1$\kern -\nulldelimiterspace \radical \z@ { #2 }$
\ifx #1 \displaystyle
\dimen@ = \fontdimen 8 \textfont 3
\advance \dimen@ .25 \fontdimen 5 \textfont 2
\else
\dimen@ = 1.25 \fontdimen 8 \textfont
\ifx #1 \textstyle
\textfont
\else
\ifx #1 \scriptstyle
\scriptfont
\else
\scriptscriptfont
\fi
\fi
3
\fi
\fi
\advance \dimen@ -\ht\z@
\ht\z@ = -\dimen@
\ifvmode \leavevmode \fi
{ }
\box\z@
}
\ifx \dimen@ = \fontdimen 8 \textfont 3
\else
\dimen@ = \fontdimen 8 \textfont
\ifx \dimen@ = \fontdimen 8 \textfont
\else
\ifx \dimen@ = \fontdimen 8 \textfont
\else
\fi
\fi
\else
\advance \dimen@ -\ht\z@
\ht\z@ = -\dimen@
\ifvmode \leavevmode \fi
{ }
\box\z@
}
\end{verbatim}

Here the additional set of braces is absolutely necessary, otherwise the changed mathematical style would be applied to the material after the \mathchoice construct. As the original command works in both text and math mode, we use \texttt{\textbackslash ensuremath} here.

\begin{verbatim}
{ \texttt{\textbackslash ensuremath} {
\use:c { cramped \cs_to_str:N #1 } \#2
}
}
\end{verbatim}

3.8 icomma

The icomma package uses \texttt{\textbackslash mathchardef} to save the mathematical code of the comma character. This breaks for Unicode fonts. The incompatibility was noticed by Peter Breitfeld.\footnote{https://groups.google.com/forum/#!topic/de.comp.text.tex/Cputk-AJS51/discussion}
\mathcomma \icomma defines the mathematical character shorthand \icomma at the beginning of the document, therefore we again patch @{begindocumenthook@}.}

269 \@@_after_package_or_now:nn { \icomma } {
270 \@iflt\r \fmtversion { 2020/10/01 } {]
271 \hook_gput_code:nnn { \begindocument } { \lualatex-math } {
272 \@@_set_mathchar:NN \mathcomma ,
273 \mathcode `\, = "8000 -
274 }
275 \hook_gset_rule:nnnn
276 { \begindocument } { \lualatex-math } { voids } { \icomma }
277 }
278 \tl_replace_once:Nnn \@begindocumenthook {
279 \mathchardef \mathcomma \mathcode `\,]
280 }
281 \@@_set_mathchar:NN \mathcomma ,
282 }
283 }
284 }
285 (/>package)

4 Implementation of the LuaLaTeX module

For the Lua module, we use the standard luatexbase-modutils template.

286 ("lua")
287 lualatex = lualatex or {}
288 lualatex.math = lualatex.math or {}
289 luatexbase.provides_module({
290 name = "lualatex-math",
291 date = "2013/08/03",
292 version = 1.3,
293 description = "Patches for mathematics typesetting with LuaLaTeX",
294 author = "Philipp Stephani",
295 licence = "LPPL v1.3+"
296 })

unpack The function unpack needs to be treated specially as it got moved around in Lua 5.2.

297 local unpack = unpack or table.unpack

298 local cctb = luatexbase.catcodetables or
299 {string = luatexbase.registernumber("catcodetable@string")}

print_class_fam_slot The function print_class_fam_slot takes one argument which must be a number. It interprets the argument as a Unicode code point whose mathematical code is printed in the form \texttt{(class)}\texttt{(family)}\texttt{(slot)}, suitable for the right-hand side of \texttt{\mathchardef}.}

300 function lualatex.math.print_class_fam_slot(char)
301 local code = tex.getmathcode(char)
302 local class, family, slot = unpack(code)
303 local result = string.format("%i %i %i ", class, family, slot)
304 tex.sprint(cctb.string, result)
305 end
306 return lualatex.math
307 (/>lua)
Change History

v0.1
General: Initial version .............................................. 1

v0.2
General: Added patch for the icomma package .......................... 9

v0.3
General: Patched math group allocation to gain access to all families .... 5
v0.3a
General: Updated for changes in \texttt{l3kernel} ...................... 1
v0.3b
\begindocumenthook: Another update for a change in \texttt{l3kernel} .... 6
v0.3c
@@_set_mathchar:NN: \texttt{l3kernel} renamed \texttt{lua_now:x} to \texttt{lua_now:x:n} .... 4

v1.0
General: Switched to \texttt{l3docstrip} ................................ 1

v1.1
@@_set_mathchar:NN: Update reasoning why \texttt{Umathcharnumdef} is not used here 4
General: Add fix and unit test for \texttt{amsopn} ...................... 8

v1.10
General: Skip patch if \texttt{mathtools} is recent enough ............... 8
Use new \LaTeX\ 2e hook management if available ..................... 9

v1.11
General: Adapt to March 2021 changes to \texttt{mathtools} ............ 8

v1.12
\after_package:nn: Use the new generic hook names if available ..... 4

v1.2
\after_equal_mathchar: Replace removed macro \texttt{chk_if_free_cs:N} .... 6

v1.3
General: Stop using the deprecated \texttt{module} function .......... 10
v1.3a
@@_set_mathchar:NN: \texttt{l3kernel} has (currently) dropped \texttt{lua_now:x:n} .... 4

v1.4
\MT_cramped_internal:NN: Added \texttt{ensuremath} to work around issue 11 .... 9
General: Removed patch for math group allocation; the kernel itself now supports all available math families .................. 5
v1.4a
@@_set_mathchar:NN: \texttt{lua_now:x:n} is back ........................ 4
General: Avoid \texttt{RequireLuaModule} ................................ 2
Load \texttt{luatexbase} only if required ............................. 2
Load all of \texttt{luatexbase} ........................................ 10
Pick up new name for string catcode table where available ........ 10
Use expl3 versions of \LaTeX\ math primitives ........................ 2

v1.5
General: Removed patch for \texttt{Mathstrutbox}; \texttt{amsmath} now has a definition usable in \LaTeX\ 2eX .............................................. 7
Removed unused helper macro \texttt{@@_char_dim:NN} ................... 6
Removed unused Lua function \texttt{print_fam_slot} ................. 10

v1.6
General: Removed patch for \texttt{newcodes}; \texttt{amsmath} now has a definition usable in \LaTeX\ 2eX .............................................. 8

v1.7
\genfrac: Adapt patch to changes in \texttt{amsmath} .................. 8

v1.8
@@_set_mathchar:NN: \texttt{lua_now:x:n} is now called \texttt{lua_now:e} .... 4
Stop using \texttt{\ldots:D} control sequences ........................ 4
\frac: Stop using \texttt{\ldots:D} control sequences ................. 5, 7
\genfrac: Stop using \ldots:D control sequences ........................................ 8
General: Stop using \ldots:D control sequences .................................... 6
\subarray: Stop using \ldots:D control sequences ................................... 7
v1.9
\begindocumenthook: Don’t patch newer versions of amsmath .................. 6
\MT_cramped_internal:Nn: Stop using \ldots:D control sequences ............... 9
\frac: Adapt to changes in \LaTeX\ kernel .......................................... 5
\_\_equal_mathchar: Don’t patch newer versions of amsmath ................... 6
General: Require 2020 version of \LaTeX\  ......................................... 2
Use builtin \LaTeX\ hooks if available ............................................. 2
\subarray: Don’t patch newer versions of amsmath .................................. 7
Stop using \ldots:D control sequences ............................................ 7

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\$: .................................................................................. 16, 18
\,: .................................................................................. 272, 273, 279, 281
\, : .................................................................................. 117, 119, 124, 132, 135
\, = .............................................................................. 118, 120, 125, 133, 136
\_\_after_package.nn ......................................................... 70, 96, 121
\_\_after_package_or_now.nn ................................................. 95, 129, 227, 269
\_\_before_package:nn ......................................................... 70, 115
\_\_patch.Hmnn ................................................................. 40, 140, 229
\_\_patch.chmnn ................................................................. 40, 100, 181, 192
\_\_restore_catcode:N ......................................................... 11, 16
\_\_set_mathchar:NN ........................................................ 63, 117, 118, 135, 136, 272, 281
\_\_temp:w ......................................................................... 39, 44, 45, 53
\_\_above ................................................................. 201, 216
\_\_before ................................................................. 184, 188, 201, 214
\_\_begindocumenthook ...................................................... 129, 278
\_\_dimen ................................................................. 70, 78, 270
\_\_ifpackage .......................................................... 116, 130, 139, 228
\_\_ifpackageloaded .......................................................... 96, 99
\_\_mathstyle ............................................................... 196, 208
\_ 

A

\addtolength .......................................................... 72, 75, 80, 83
\advance ............................................................................ 147, 241, 255
\alignmark ........................................................................ 176
\amsmath \(\text{package}\) ...................................................... 1, 2, 5, 6, 11, 12
\amsopn \(\text{package}\) ...................................................... 11
\AtBeginIfPackageFile ...................................................... 88
\AtEndIfPackage .......................................................... 15
\AtEndIfPackageFile .......................................................... 91
\AtEndPreamble ........................................................... 98

B

\baselineskip ............................................................... 146, 147, 164
\begin{group} .......................................................... 102, 184, 199
\begin{group} ............................................................. 142, 153
\bison ................................................................. 2
\begin{itemize}
\item \texttt{hbox} ~~~ 231
\item \texttt{hfil} ~~~ 154, 156, 172, 178
\item \texttt{hook_gput_code:nn} ~~~ 271
\item \texttt{hook_gset_rule:nnn} ~~~ 275
\item \texttt{ht} ~~~ 255, 256
\item \texttt{\&align} ~~~ 152, 170
\item \texttt{icomma} \textit{(package)} ~~~ 1, 2, 9-11
\item \texttt{ifmode} ~~~ 257
\item \texttt{ifx} ~~~ 154, 201, 239, 244, 247
\item \texttt{int_const:Nn} ~~~ 111, 112
\item \texttt{int_eval:n} ~~~ 12, 66
\item \texttt{kern} ~~~ 235
\item \texttt{\textbackslash{}3docstrip} \textit{(package)} ~~~ 11
\item \texttt{\textbackslash{}3kernel} \textit{(package)} ~~~ 11
\item \texttt{\textbackslash{}_\textbackslash{}_equal_mathchar} ~~~ 113, 123, 125
\item \texttt{\textbackslash{}_\textbackslash{}_equal_mathchar} ~~~ 113, 122, 124
\item \texttt{leavevmode} ~~~ 257
\item \texttt{\textbackslash{}let\textbackslash{}0} ~~~ 143, 161
\item \texttt{\textbackslash{}lineskip} ~~~ 149, 151, 168, 169
\item \texttt{\textbackslash{}linekip} ~~~ 151, 169
\item \texttt{\textbackslash{}lua_now:e} ~~~ 65
\item \texttt{luatex-base} \textit{(message)} ~~~ 19
\item \texttt{luatex-base-modutils} \textit{(package)} ~~~ 11
\item \texttt{luatex-base-modutils} \textit{(package)} ~~~ 10
\item \texttt{\textbackslash{}with} ~~~ 155, 174, 233
\item \texttt{macro-expected} \textit{(message)} ~~~ 23
\item \texttt{mathchardef} ~~~ 132, 133, 279
\item \texttt{mathcode} ~~~ 132, 133, 273, 279
\item \texttt{mathcomma} ~~~ 269
\item \texttt{mathstyle} ~~~ 2
\item \texttt{\texttt{mathtools}} \textit{(package)} ~~~ 1, 2, 8, 11
\item messages:
  \begin{itemize}
  \item \texttt{luatex-required} ~~~ 19
  \item \texttt{macro-expected} ~~~ 23
  \item \texttt{patch-macro} ~~~ 32
  \item \texttt{wrong-meaning} ~~~ 26
  \end{itemize}
\item \texttt{module (function)} ~~~ 11
\item \texttt{\textbackslash{}msg_error:nn} ~~~ 36
\item \texttt{\textbackslash{}msg_info:nn} ~~~ 46
\item \texttt{\textbackslash{}msg_new:nn} ~~~ 19, 23, 26, 32
\item \texttt{\textbackslash{}msg_warning:nn} ~~~ 57
\item \texttt{\textbackslash{}msg_warning:nxxx} ~~~ 51
\item \texttt{\textbackslash{}MT\_cramped_internal:Nn} ~~~ 227
\end{itemize}
### Packages

- **amsmath**: 1, 2, 5, 6, 11, 12
- **amssymb**: 11
- **etoolbox**: 2
- **expl3**: 2, 11
- **filehook**: 2
- **l3docstrip**: 11
- **l3kernel**: 11
- **luatexbase**: 11
- **luatexbase-modutils**: 10
- **mathtools**: 1, 2, 8, 11
- **nath**: 5
- **unicode-math**: 1, 2
- **patch-macro**: 32
- **prg_do_nothing**: 39
- **print_class_fam_slot**: 10, 300
- **print_fam_slot**: 11
- **ProvidedExplPackage**: 5

### Requirements

- **\\over**: 102, 106
- **\relax**: 236
- **\RequirePackage**: 132, 133, 201
- **\RequirePackage\relax**: 4, 7, 9, 86
- **\restore@math@cr**: 144, 162
- **\scriptstyle**: 5
- **\scriptscriptfont**: 146, 147, 149, 248
- **\textfont**: 155, 165, 166, 168, 175, 247
- **\textstyle**: 231
- **\textsize**: 164
- **\textsize@equals**: 123, 133, 136
- **\textsize@equal**: 122
- **\textsize@minus**: 122, 132, 135
- **\textstyle**: 140
- **\scriptstyle**: 139
- **\scriptscriptstyle**: 35

### Commands

- **\textfont**: 240, 241, 245
- **\textstyle**: 244
- **\textwidth**: 149
- **\textwidth@empty**: 213
- **\textwidth@new**: 113, 114
- **\textwidth@replaceonce**: 131, 278
- **\textwidth_if_eq_meaning**: 172
- **\textwidth_if_macro**: 42
- **\textwidth_if_meaning**: 52, 53
- **\textwidth_if_starts**: 47, 52, 58
- **\textwidth@new**: 146, 147

---

** Robertson, Will **
<table>
<thead>
<tr>
<th>Category</th>
<th>Command</th>
<th>Reference Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>\backslashmathchardef</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>\backslashmathcodenom</td>
<td>124, 125</td>
</tr>
<tr>
<td></td>
<td>\backslashmathstackdenomdown</td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>\backslashmathstacknumup</td>
<td>165</td>
</tr>
<tr>
<td></td>
<td>\backslashmathstackvgap</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>\pkg{unicode-math}</td>
<td>1, 2</td>
</tr>
<tr>
<td></td>
<td>unstack (function)</td>
<td>10, 297</td>
</tr>
<tr>
<td></td>
<td>\use:\c</td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>\vcenter</td>
<td>106, 188, 211</td>
</tr>
<tr>
<td></td>
<td>\startmath</td>
<td>173</td>
</tr>
<tr>
<td></td>
<td>\stopmath</td>
<td>177</td>
</tr>
<tr>
<td>V</td>
<td>\vcenter</td>
<td>141, 159</td>
</tr>
<tr>
<td>W</td>
<td>wrong-meaning (message)</td>
<td>26</td>
</tr>
<tr>
<td>Z</td>
<td>\v@</td>
<td>231, 236, 255, 256, 259</td>
</tr>
</tbody>
</table>