The \texttt{lualatex-math} package*

Philipp Stephani
p.stephani2@gmail.com
2020/09/25

Contents

1 Introduction 1

2 Interface 2

3 Implementation of the \LaTeX\ 2e package 2
  3.1 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  3.2 Messages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  3.3 Initialization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  3.4 Patching . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  3.5 \LaTeX\ 2e kernel . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  3.6 \amsmath . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  3.7 \mathtools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
  3.8 \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\TeX, 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\TeX.

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} v1.9, dated 2020/09/25.
alternatives, it has become an accepted way of providing new functionality in \LaTeX.

To keep the negative impact as small as possible, the lualatex-math package patches only the \LaTeX\ 2ε 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 \LaTeXX. By its very nature, the package is likely to cause problems; in such cases, please refer to the issue tracker\(^1\).

## 2 Interface

The 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 lualatex-math package needs \LaTeXX 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 lualatex-math package is independent of the unicode-math package; the fixes provided here are valid for both Unicoide and legacy math typesetting.

Currently patches for the \LaTeX\ 2ε kernel and the amsmath, mathtools and icomma packages are provided. It is not relevant whether you load these packages before or after 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 \LaTeXX.

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

## 3 Implementation of the \LaTeX\ 2ε package

### 3.1 Requirements

```latex
\texttt{\mathstyle}
```

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

```latex
\texttt{\cs_new_nopar:Npn \textbackslash@@_restore_catcode:N \#1 \{ \char_set_catcode:n \{ \int_eval:n \{ \textbackslash \textquoteleft \#1 \textbackslash \textquoteleft \} \} \char_value_catcode:n \{ \textbackslash \textquoteleft \#1 \textbackslash \textquoteleft \} \}}
```

\(^1\)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

luatex-required Issued when not running under \texttt{LuaTeX}.
\msg_new:nnn { lualatex-math } { luatex-required } {
I\- \textup{will}\- \textup{stop}\- \textup{loading}\- \textup{now}.}

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\- \textup{expected}\- that\- \#1\- \text{is}\- a\- macro,\- but\- \text{it}\- \text{isn't}.}

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\- \textup{expected}\- \#1\- \text{to}\- \text{have}\- \text{the}\- \text{meaning} \\%
\#3, \%
but\- \text{it}\- \text{has}\- \text{the}\- \text{meaning} \%
\#2.}

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\- \text{going}\- \text{to}\- \text{patch}\- \text{macro}\- \#1.}

3.3 Initialization

Unless we are running under \texttt{LuaTeX}, 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 \texttt{command} \texttt{factory command} \{\texttt{parameter text}\} \{\texttt{expected replacement text}\} \{\texttt{new replacement text}\} tries to patch \texttt{command}. If \texttt{command} is undefined, do nothing. Otherwise it must be a macro with the given \texttt{parameter text} and \texttt{expected replacement text}, 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.

\begin{verbatim}
\cs_new_protected_nopar:Npn \@_patch:NNnnn #1 #2 #3 #4 #5 { \\
\cs_if_exist:NT #1 { \\
  \token_if_macro:NTF #1 { \\
    \group_begin: \\
    \#2 \@@_temp:w \#3 { \#4 } \\
    \cs_if_eq:NNTF #1 \@@_temp:w { \\
      \msg_info:nnx { lualatex-math } { patch-macro } \\
      { \token_to_str:N #1 } \\
      \group_end: \\
    \#2 \#3 { \#5 } \\
  } { \\
    \msg_warning:nnxxx { lualatex-math } { wrong-meaning } \\
    \{ \token_to_str:N #1 \} \token_to_meaning:N \#1 \token_to_meaning:N \@@_temp:w \\
    \group_end: \\
  } \\
} { \\
  \msg_warning:nnx { lualatex-math } { macro-expected } \\
  \{ \token_to_str:N #1 \} \\
} \\
\cs_generate_variant:Nn \@@_patch:NNnnn { c } \\
\@@_set_mathchar:NN
\end{verbatim}

The macro \texttt{\@_set_mathchar:NN(control sequence)(token)} defines the \texttt{(control sequence)} as an extended mathematical character shorthand whose mathematical code is given by the mathematical code of the character `\texttt{(token)}. We cannot use the \texttt{\Umathcharnumdef} primitive here since we would then rely on the \texttt{\Umathcodenum} primitive which is currently broken.\footnote{http://tug.org/pipermail/luatex/2012-October/003794.html}

\begin{verbatim}
\cs_new_protected_nopar:Npn \@_set_mathchar:NN #1 #2 { \\
\Umathchardef #1 \\
\lua_now:e { \\
  lualatex.math.print_class_fam_slot( \int_eval:n { `#2 } ) \\
} \\
\scan_stop: \\
\}
\end{verbatim}

The macro \texttt{\@before_package:n}{\texttt{(package)}\texttt{\{code\}}} executes the \texttt{(code)} before the \texttt{(package)} is loaded. Accordingly, \texttt{\@after_package:n}{\texttt{(package)}\texttt{\{code\}}} executes the \texttt{(code)} after the \texttt{(package)} is loaded. If the \texttt{(package)} is already loaded, nothing happens. We prefer using native \TeX{} hooks if possible.

\begin{verbatim}
@ifl@t@r \fmtversion { 2020/10/01 } { \\
\cs_new_protected_nopar:Npn \@before_package:nn #1 #2 { \\
\AddToHook { package/before/#1 } { \#2 } \\
} \\
\cs_new_protected_nopar:Npn \@after_package:nn #1 #2 { \\
\AddToHook { package/after/#1 } { \#2 } \\
} \\
\} \\
\RequirePackage { filehook } [ 2011/03/09 ] \\
\cs_new_protected_nopar:Npn \@before_package:nn #1 #2 { \\
\AtBeginOfPackageFile { \#1 } { \#2 } \\
} \\
\end{verbatim}
The macro \@after_package_or_now:nn{\package}{\code} executes the \code after the \package is loaded. If the \package is already loaded, the \code 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 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 \fracL.

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

\amsmath
The popular amsmath package is subject to three Lua\TeX-related problems:

\begin{itemize}
\item The \mathcode primitive is used several times, which fails for Unicode math characters. \Umathcode should be used instead.
\item 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.
\item The fraction commands \frac and \genfrac don’t use the \Ustack primitive.
\end{itemize}

These problems have been fixed in version 2.17i of amsmath, so we don’t attempt to patch it if that version is loaded.
These constants contain the standard \texttt{\LaTeX} mathematical codes for the minus and the equal signs. We temporarily set the math codes to these constants before loading the \texttt{amsmath} package so that it can request the legacy math code without error.

\begin{verbatim}
102 \int_const:Nn \c_@@_std_minus_mathcode_int { "2200 }
103 \int_const:Nn \c_@@_std_equal_mathcode_int { "303D }
\end{verbatim}

These mathematical characters are saved before \texttt{amsmath} is loaded so that we can temporarily assign the \texttt{\LaTeX} values to the mathematical codes of the minus and equals signs. The \texttt{amsmath} package queries these codes, and if they represent Unicode characters, the package loading will fail. If \texttt{amsmath} has already been loaded, there is nothing we can do, therefore we use the non-starred version of \texttt{\AtBeginOfPackageFile}.

\begin{verbatim}
104 \tl_new:N \l_@@_minus_mathchar
105 \tl_new:N \l_@@_equal_mathchar
106 \@@_before_package:nn { amsmath } {
107 \@ifpackagelater { amsmath } { 2020/08/24 } { } {
108 \@@_set_mathchar:NN \l_@@_minus_mathchar \-
109 \@@_set_mathchar:NN \l_@@_equal_mathchar =
110 }{ }
111}
\end{verbatim}

Now we temporarily reset the mathematical codes.

\begin{verbatim}
112 \char_set_mathcode:nn { `\- } { \c_@@_std_minus_mathcode_int }
113 \char_set_mathcode:nn { `\= } { \c_@@_std_equal_mathcode_int }
114 \@@_after_package:nn { amsmath } { }
\end{verbatim}

The \texttt{amsmath} package defines the control sequences \texttt{\std@minus} and \texttt{\std@equal} as mathematical character shorthands while loading, but uses our restored mathematical codes, which must be fixed.

\begin{verbatim}
115 \cs_set_eq:NN \std@minus \l_@@_minus_mathchar
116 \cs_set_eq:NN \std@equal \l_@@_equal_mathchar
\end{verbatim}

Finally, we restore the original mathematical codes of the two signs.

\begin{verbatim}
117 \Umathcodenum `\- \l_@@_minus_mathchar
118 \Umathcodenum `\= \l_@@_equal_mathchar
\end{verbatim}

All of the following fixes work even if \texttt{amsmath} is already loaded.

\begin{verbatim}
119\end{verbatim}

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

\begin{verbatim}
120 \@@_after_package_or_now:nn { amsmath } { }
121 \@ifpackagelater { amsmath } { 2020/08/24 } { } { }
122 \tl_replace_once:Nnn \begindocumenthook { \mathchardef \std@minus \mathcode `\- \relax
123 \mathchardef \std@equal \mathcode `\= \relax
124 }{ }
125 \@@_set_mathchar:NN \std@minus \-
126 \@@_set_mathchar:NN \std@equal =
127 }{ }
\end{verbatim}

\texttt{subarray} The \texttt{subarray} environment uses legacy font dimensions. We simply patch it to use \texttt{Lua\TeX} font parameters (and \texttt{\LaTeX3} expressions instead of \texttt{\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).

\@ifpackagelater { amsmath } { 2020/09/23 } { } {  \\
\@0_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@@  \\
⟨@@⟩  \\
\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  \\
\skip_set:Nn \baselineskip {  \\
\Umathstacknumup \scriptstyle  \\
+ \Umathstackdenomdown \scriptstyle  \\
}  \\
\lineskip \Umathstackvgap \scriptstyle  \\
\lineskiplimit \lineskip  \\
\ialign  \\
\c_group_begin_token  \\
\token_if_eq_meaning:NNT c #1 { \hfil }  \\
\Ustartmath  \\
\m@th  \\
\scriptstyle  \\
\alignmark \alignmark  \\
\Ustopmath  \\
\hfil  \\
\crcr  \\
}  \\
\frac  \\
\frac is declared by \DeclareRobustCommand, we must patch the macro \frac.  \\
\@0_patch:cNnnn \ cs_set:Npn \ #1 \ #2 \ } {  \\
\begingroup \Ustack { \ group_begin: \ #1 \ group_end: \ #2 \ }  \\
\endgroup \@@over \ #2  \\
\}  \\
}  \\
\frac
Generalized fractions are typeset by the \texttt{\textbackslash genfrac} command. Since \texttt{\textbackslash genfrac} is declared by \texttt{\textbackslash DeclareRobustCommand}, we have to patch the macro \texttt{\textbackslash genfrac}.

\begin{verbatim}
\@\patch:cNnnn { genfrac- } \cs_set:Npn { #1 #2 #3 #4 #5 #6 }
\{
 \@mathstyle { #4 }
 \genfrac@choice o { #1 }
 \begin{group}
 \langle \@@
 \ifx @ #3 @ \@@over \else \@@above \fi #3 \relax
 \else
 \genfrac@choice c { #2 }
 \}
 \end{group}
 \}
}
\@mathstyle { #4 }
\genfrac@choice o { #1 }

\MT_cramped_internal:Nn
\texttt{\textbackslash MT\_cramped\_internal:Nn(\textit{style})(\textit{expression})} typesets the \textit{expression} in the cramped style corresponding to the given \textit{style} (\texttt{\textbackslash displaystyle} etc.); all we have to do in Lua\TeX\ is to select the correct primitive. Rewriting the user-level \texttt{\textbackslash cramped} command and employing \texttt{\textbackslash mathstyle} 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 } { 
\@\patch:NNnnn \MT_cramped_internal:Nn
\cs_set_nopar:Npn { #1 #2 } { 
\sbox \z@ { $ \m@th #1 $ } 
\m@th
#1
\end\verbatim

\textbf{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. Lua\TeX\ has primitives for setting material in cramped mode, so we make use of them.

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 \ensuremath here.

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

3.8 icomma

The icomma package uses \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-AJS5I/discussion}

\begin{verbatim}
\mathchardef \mathcomma \mathcode `,\n\end{verbatim}
icomma defines the mathemathical character shorthand \icomma at the beginning
of the document, therefore we again patch \@begindocumenthook.

\begin{verbatim}
\@@_after_package_or_now:nn { icomma } { \tl_replace_once:Nnn \@begindocumenthook { \mathchardef \mathcomma \mathcode `\n\@@_set_mathchar:NN \mathcomma \,\n} }
\end{verbatim}
4 Implementation of the LuaLaTeX module

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

```latex
(*lua*)
lualatex = lualatex or {}
lualatex.math = lualatex.math or {}
luatexbase.provides_module({
  name = "lualatex-math",
date = "2013/08/03",
version = 1.3,
description = "Patches for mathematics typesetting with LuaLaTeX",
author = "Philipp Stephani",
licence = "LPPL v1.3+"
})
```

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

```latex
local unpack = unpack or table.unpack
```

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 \(\text{\textlangle class\rangle}\text{\textlangle family\rangle}\text{\textlangle slot\rangle}\), suitable for the right-hand side of \texttt{\textbackslash Umathchardef}.

```latex
function lualatex.math.print_class_fam_slot(char)
  local code = tex.getmathcode(char)
  local class, family, slot = unpack(code)
  local result = string.format("%i %i %i ", class, family, slot)
  tex.sprint(cctb.string, result)
end
```

return lualatex.math
```

(/lua)

Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>General: Initial version</td>
</tr>
<tr>
<td>v0.2</td>
<td>General: Added patch for the icomma package</td>
</tr>
<tr>
<td>v0.3</td>
<td>General: Patched math group allocation to gain access to all families</td>
</tr>
<tr>
<td>v0.3a</td>
<td>General: Updated for changes in l3kernel</td>
</tr>
<tr>
<td>v0.3b</td>
<td><code>\begindocumenthook</code>: Another update for a change in l3kernel</td>
</tr>
<tr>
<td>v0.3c</td>
<td>\texttt{\textbackslash@@_set_mathchar:NN}: l3kernel renamed \texttt{\textbackslash lua_now:x} to \texttt{\textbackslash lua_now:x:n}</td>
</tr>
<tr>
<td>v1.0</td>
<td>General: Switched to l3docstrip</td>
</tr>
<tr>
<td>v1.1</td>
<td>\texttt{\textbackslash@@_set_mathchar:NN}: Update reasoning why \texttt{\textbackslash Umathcharnumdef} is not used here</td>
</tr>
<tr>
<td></td>
<td>General: Add fix and unit test for amsopn</td>
</tr>
</tbody>
</table>
v1.2
\_\_\_equal_mathchar: Replace removed macro \chk_if_free_cs:N ............. 6
v1.3
General: Stop using the deprecated module function .................... 10
v1.3a
\@\_set_mathchar:NN: \texttt{l3kernel} has (currently) dropped \lua_now:x:n .... 4
v1.4
\MT_cramped_internal:Nn: Added \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: \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 Lua\TeX{} math primitives .......................... 2
v1.5
General: Removed patch for \texttt{\newmcodes@}; amsmath now has a definition
usable in Lua\TeX{} ........................................................... 6
Removed unused helper macro \@\_char_dim:NN ............................... 6
Removed unused Lua function print_fam_slot ................................ 10
v1.6
General: Removed patch for \texttt{\newmcodes@}; amsmath now has a definition usable in
Lua\TeX{} ............................................................................. 8
v1.7
\texttt{\genfrac}: Adapt patch to changes in amsmath ......................... 8
v1.8
\@\_set_mathchar:NN: \lua_now:x:n is now called \lua_now:e ................ 4
\texttt{\frac}: Stop using \texttt{\_\_\_D} control sequences .................. 4
\texttt{\frac}: Stop using \texttt{\_\_\_D} control sequences .................. 5, 7
\texttt{\genfrac}: Stop using \texttt{\_\_\_D} control sequences ................ 8
\texttt{\subarray}: Stop using \texttt{\_\_\_D} control sequences ................. 6
\texttt{\subarray}: Stop using \texttt{\_\_\_D} control sequences ................. 7
v1.9
\texttt{\texttt\begindocumenthook}: Don’t patch newer versions of amsmath ........ 6
\MT_cramped_internal:Nn: Stop using \texttt{\_\_\_D} control sequences .......... 9
\texttt{\frac}: Adapt to changes in \texttt{\LaTeX{}2e} kernel .................. 5
\_\_\_equal_mathchar: Don’t patch newer versions of amsmath .............. 6
General: Require 2020 version of \texttt{\LaTeX{}2e} ......................... 2
Use \texttt{\LaTeX{}2e} hooks if available .................................. 2
\texttt{\subarray}: Don’t patch newer versions of amsmath ................. 7
Stop using \texttt{\_\_\_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
\_ ............................................................... 258, 260
\_ ............................................................... 108, 110, 115, 123, 126
\_ ............................................................... 109, 111, 116, 124, 127
\@ after_package:nn ...................................... 70, 87, 112
\@ after_package_or_now:nn ............................... 86, 120, 218, 256
<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>@before_package:mn</td>
<td>70, 106</td>
</tr>
<tr>
<td>@patch:Nm</td>
<td>40, 131, 219</td>
</tr>
<tr>
<td>@patch:cNmn</td>
<td>40, 91, 172, 183</td>
</tr>
<tr>
<td>@restore_catcode:N</td>
<td>11, 16</td>
</tr>
<tr>
<td>@set_mathchar:N</td>
<td>63, 108, 109, 126, 127, 260</td>
</tr>
<tr>
<td>@temp:w</td>
<td>39, 44, 45, 53</td>
</tr>
<tr>
<td>@above</td>
<td>192, 207</td>
</tr>
<tr>
<td>@over</td>
<td>175, 179, 192, 205</td>
</tr>
<tr>
<td>@begindocumenthook</td>
<td>120, 257</td>
</tr>
<tr>
<td>@ifempty</td>
<td>70</td>
</tr>
<tr>
<td>@ifpackage:later</td>
<td>107, 121, 130</td>
</tr>
<tr>
<td>@ifpackage:loaded</td>
<td>87, 90</td>
</tr>
<tr>
<td>@mathstyle</td>
<td>187, 199</td>
</tr>
<tr>
<td>@</td>
<td>20, 27, 28, 29</td>
</tr>
<tr>
<td>\AddToHook</td>
<td>72, 75</td>
</tr>
<tr>
<td>\advance</td>
<td>138, 231, 245</td>
</tr>
<tr>
<td>\alignmark</td>
<td>167</td>
</tr>
<tr>
<td>amsmath (package)</td>
<td>1, 2, 5, 6, 11</td>
</tr>
<tr>
<td>amsopn (package)</td>
<td>10</td>
</tr>
<tr>
<td>\AtBeginDiffPackageFile</td>
<td>80</td>
</tr>
<tr>
<td>\AtEndDiffPackage</td>
<td>15</td>
</tr>
<tr>
<td>\AtEndDiffPackageFile</td>
<td>83</td>
</tr>
<tr>
<td>\AtEndPreamble</td>
<td>89</td>
</tr>
<tr>
<td>\baselineskip</td>
<td>137, 138, 155</td>
</tr>
<tr>
<td>\begin{group}</td>
<td>93, 175, 190</td>
</tr>
<tr>
<td>\group</td>
<td>133, 144</td>
</tr>
<tr>
<td>\binom</td>
<td>2</td>
</tr>
<tr>
<td>\box</td>
<td>247</td>
</tr>
<tr>
<td>Breitfeld, Peter</td>
<td>9</td>
</tr>
<tr>
<td>\c_@@_std_equal_mathcode_int</td>
<td>102, 111</td>
</tr>
<tr>
<td>\c_@@_std_minus_mathcode_int</td>
<td>102, 110</td>
</tr>
<tr>
<td>\c_group_begin_token</td>
<td>151, 162</td>
</tr>
<tr>
<td>\char_set_catcode:nn</td>
<td>12</td>
</tr>
<tr>
<td>\char_set_catcode:math_toggle:N</td>
<td>18</td>
</tr>
<tr>
<td>\char_set_mathcode:nn</td>
<td>110, 111</td>
</tr>
<tr>
<td>\char_value_catcode:n</td>
<td>13</td>
</tr>
<tr>
<td>\crrr</td>
<td>148, 170</td>
</tr>
<tr>
<td>\cs_generate_variant:Nn</td>
<td>62</td>
</tr>
<tr>
<td>\cs_if_eq:NNTF</td>
<td>45</td>
</tr>
<tr>
<td>\cs_if_exist:VF</td>
<td>8</td>
</tr>
<tr>
<td>\cs_if_exist:VT</td>
<td>41</td>
</tr>
<tr>
<td>\cs_new_eq:N</td>
<td>39</td>
</tr>
<tr>
<td>\cs_new:nopar:Mpn</td>
<td>11</td>
</tr>
<tr>
<td>\cs_new_protected_nopar:Mpn</td>
<td>40, 63, 71, 74, 79, 82, 86</td>
</tr>
<tr>
<td>\cs_set:Mpn</td>
<td>91, 131, 172, 183</td>
</tr>
<tr>
<td>\cs_set_eq:NN</td>
<td>113, 114</td>
</tr>
<tr>
<td>\cs_set_nopar:Mpn</td>
<td>220</td>
</tr>
<tr>
<td>\cs_to_str:N</td>
<td>251</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>\defaultTag</td>
<td>136, 154</td>
</tr>
<tr>
<td>\dimen#</td>
<td>230, 231, 233, 245, 246</td>
</tr>
<tr>
<td>\directlua</td>
<td>10</td>
</tr>
</tbody>
</table>
\skip_set:Nn  \std@equal  \std@equals  \std@minus  \subarray (environment)  \sys_if_engine_luatex:F
\textfont  \textstyle  \textstyle  \textstyle  \textstyle
\thr@@  \tl_if_empty:nTF  \tl_new:N  \tl_replace_once:Nnn  \token_if_eq_meaning:NNT
\token_if_macro:NTF  \token_to_meaning:N  \token_to_str:N  \tw@
\Umathchardef  \Umathcodenum  \Umathstackdenomdown  \Umathstacknumup  \Umathstackvgap
\unicode-math (package)  \unpack (function)  \use:c  \Ustack  \Ustartmath  \Ustopmath
\vcenter  \wrong-meaning (message)  \vcenter
\z@