The `latex-lab-amsmath` code*

LaTeX Project

October 30, 2026

Abstract

Contents

1 Introduction **1**

2 The Implementation **1**
2.1 File declaration . . . 2
2.2 Measuring . . . 2
2.3 Display environments . . . 2
2.3.1 Tag . . . 2
2.3.2 align & friends . . . 3
2.3.3 `gather` and `gather*` . . . 5
2.3.4 gathered, aligned and alignedat . . . 7
2.3.5 multline and multline* . . . 8
2.4 Cases . . . 11
2.4.1 smallmatrix . . . 12
2.5 The split environment . . . 12
2.6 `\intertext` . . . 17
2.7 `\text` . . . 18
2.8 `\pmb` . . . 20

Index **21**

1 Introduction

This file implements adaptations to the amsmath package needed for the tagging project.

2 The Implementation

Better no @@ expansion here .....

```
1 ⟨@@=⟩
2 ⟨*kernel⟩
```

*

2.1 File declaration

```
\ProvidesFile{latex-lab-amsmath.ltx}
        [2026-04-25 v0.1j amsmath adaptions]

\ExplSyntaxOn
```

2.2 Measuring

When measuring we neither want tagging nor the luamml processing.

\measuring@true

```
\AddToHook{begindocument}
 {
   \def\measuring@true{\let\ifmeasuring@\iftrue\tag_suspend:n{\measuring}\luamml_ignore:}
 }
```

(*End of definition for* `\measuring@true`*. This function is documented on page* **??**.)

2.3 Display environments

2.3.1 Tag

The tag/label must be saved, so that it can be reinserted later.

TODO: `\maketag@@@` is perhaps used in places where tagging/luamml handling is not wanted. This must be checked and handled.

\maketag@@@

```
\def\maketag@@@#1
  {%
    \ifmeasuring@
      \hbox{\m@th\normalfont#1}%
    \else
      \UseTaggingSocket{math/display/tag/begin}
      \hbox{\m@th\normalfont#1
      \UseTaggingSocket{math/luamml/mtable/tag/save}
      }%
      \UseTaggingSocket{math/display/tag/end}
    \fi
   }
```

(*End of definition for* `\maketag@@@`*. This function is documented on page* **??**.)

`\eqref` uses `\tagform@` and so `\maketag@@@` but we do not want this tagging there.

\eqref
\maketag@@@notog

```
\def\maketag@@@notag#1{\hbox{\m@th\normalfont#1}}
\DeclareRobustCommand{\eqref}[1]
  {\textup{\let\maketag@@@\maketag@@@notag\tagform@{\ref{#1}}}}
```

(*End of definition for* `\eqref` *and* `\maketag@@@notog`*. These functions are documented on page* **??**.)

2.3.2 align & friends

Most display environment uses a common command for the end which contains the luamml socket to finalize the mtable.

\common@align@ending

```
\def\common@align@ending {
  \math@cr \black@\totwidth@
  \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {\@currenvir}
  \egroup
  \ifingather@
    \restorealignstate@
    \egroup
    \nonumber
    \ifnum0=`{\fi\iffalse}\fi
  \else
    \dollardollar@end
  \fi
  \ignorespacesafterend
}
```

(*End of definition for* `\common@align@ending`*. This function is documented on page* **??**.)

Now we redefine the display alignments to use these ending.

```
\renewenvironment{align}{%
  \start@align\@ne\st@rredfalse\m@ne
}{%
  \common@align@ending
}
\renewenvironment{align*}{%
  \start@align\@ne\st@rredtrue\m@ne
}{%
  \common@align@ending
}
\renewenvironment{alignat}{%
  \start@align\z@\st@rredfalse
}{%
  \common@align@ending
}
\renewenvironment{alignat*}{%
  \start@align\z@\st@rredtrue
}{%
  \common@align@ending
}
\renewenvironment{flalign}{%
  \start@align\tw@\st@rredfalse\m@ne
}{%
  \common@align@ending
}
\renewenvironment{flalign*}{%
  \start@align\tw@\st@rredtrue\m@ne
}{%
  \common@align@ending
}
\renewenvironment{xalignat}{%
```

```
  \start@align\@ne\st@rredfalse
}{%
  \common@align@ending
}
\renewenvironment{xalignat*}{%
  \start@align\@ne\st@rredtrue
}{%
  \common@align@ending
}
\renewenvironment{xxalignat}{%
  \start@align\tw@\st@rredtrue
}{%
  \common@align@ending
}
```

And register these environments for the math grabbing.

```
 \math_register_env:n {align}
 \math_register_env:n {align*}
 \math_register_env:n {alignat}
 \math_register_env:n {alignat*}
 \math_register_env:n {flalign}
 \math_register_env:n {flalign*}
 \math_register_env:n {xalignat}
 \math_register_env:n {xalignat*}
 \math_register_env:n {xxalignat}
```

The align preamble (used in `\align@`) needs code for luamml to save the cells.

`\align@preamble`

```
\def\align@preamble{%
   &\hfil
    \strut@
    \setboxz@h
      {
        \@lign
         $
          \m@th\displaystyle{##}
         \ifmeasuring@
          \luamml_ignore:
         \else
          \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
         \fi
         $
      }%
    \ifmeasuring@
      \savefieldlength@
    \else
      \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
    \fi
    \set@field
    \tabskip\z@skip
   &\setboxz@h
      {
        \@lign
```

```
          $
            \m@th\displaystyle{{}##}
            \ifmeasuring@
            \luamml_ignore:
            \else
            \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
            \fi
          $
        }%
      \ifmeasuring@
        \savefieldlength@
      \else
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
      \fi
      \set@field
      \hfil
      \tabskip\alignsep@
}
```

(*End of definition for* `\align@preamble`. *This function is documented on page* **??**.)

`\math@cr@@@align`

```
\def\math@cr@@@align{%
  \ifst@rred\nonumber\fi
  \if@eqnsw \global\tag@true \fi
  \global\advance\row@\@ne
  \add@amps\maxfields@
  \omit
  \kern-\alignsep@
  \iftag@
    \setboxz@h{\@lign\strut@{\make@display@tag}}%
    \place@tag
  \fi
  \UseTaggingSocket{math/luamml/mtable/tag/set}
  \ifst@rred\else\global\@eqnswtrue\fi
  \global\lineht@\z@
  \cr
}
```

(*End of definition for* `\math@cr@@@align`. *This function is documented on page* **??**.)

2.3.3 gather and gather*

`\gather@`

```
\def\gather@#1{%
    \ingather@true \let\split\insplit@
    \let\tag\tag@in@align \let\label\label@in@display
    \chardef\dspbrk@context\z@
    \intertext@ \displ@y@ \Let@
    \let\math@cr@@@\math@cr@@@gather
    \gmeasure@{#1}%
    \global\shifttag@false
    \tabskip\z@skip
    \global\row@\@ne
    \halign to\displaywidth\bgroup
```

```
          \strut@
          \setboxz@h
            {
             $\m@th\displaystyle{##}
              \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
             $
            }%
          \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
          \calc@shift@gather
          \set@gather@field
          \tabskip\@centering
         &\setboxz@h{\strut@{##}}%
          \place@tag@gather
          \UseTaggingSocket{math/luamml/mtable/tag/set}
           \tabskip \iftagsleft@ \gdisplaywidth@ \else \z@skip \span\fi
           \crcr
           #1%
}
```

(*End of definition for* `\gather@`*. This function is documented on page* **??**.)

`\endgather`

```
\def\endgather{
      \math@cr
      \black@ \totwidth@
      \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
  \egroup
  \dollardollar@end
  \ignorespacesafterend
}
```

(*End of definition for* `\endgather`*. This function is documented on page* **??**.)

The original definition of `gather*` uses `\endgather` but this redirection doesn't work if we alter `gather` so we use the real meaning and add the socket.

`gather*` (*env.*)

```
\renewenvironment{gather*}
 {
   \start@gather\st@rredtrue
 }
 {
    \math@cr
    \black@\totwidth@
    \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
   \egroup
   \dollardollar@end
   \ignorespacesafterend
 }
```

Register both environments

```
\math_register_env:n {gather}
\math_register_env:n {gather*}
```

2.3.4 gathered, aligned and alignedat

These environments are not grabbed as they are inside other display environments but they need various sockets for luamml support.

\start@aligned

```
\renewcommand{\start@aligned}[2]{
  \RIfM@
  \else
    \nonmatherr@ {\begin{\@currenvir}}
  \fi
  \savecolumn@ % Assumption: called inside a group
  \UseTaggingSocket{math/luamml/annotate/false}{}{ \alignedspace@left }
  \ams@start@box {#1} \bgroup
    \maxfields@ #2 \relax
    \ifnum \maxfields@ > \m@ne
      \multiply \maxfields@ \tw@
      \let \math@cr@@@ \math@cr@@@alignedat
      \alignsep@ \z@skip
    \else
      \let \math@cr@@@ \math@cr@@@aligned
      \alignsep@ \minalignsep
    \fi
    \Let@ \chardef \dspbrk@context \@ne
    \default@tag
    \spread@equation % no-op if already called
    \global \column@ \z@
    \ialign \bgroup
      & \column@plus
        \hfil
        \strut@
        $
          \m@th
          \displaystyle
          {##}
          \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
        $
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
        \tabskip \z@skip
      & \column@plus
        $
          \m@th
          \displaystyle
          {
            {}
            ##
          }
          \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
        $
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
        \hfil
        \tabskip\alignsep@
      \crcr
      \ams@return@opt@arg
```

```
}
```

(*End of definition for* `\start@aligned`. *This function is documented on page* ??.)

`gathered` (*env.*)

```
\renewenvironment{gathered}[1][c]{%
    \RIfM@\else
        \nonmatherr@{\begin{gathered}}%
    \fi
```

This annotates the space

```
    \UseTaggingSocket{ math/luamml/annotate/false } {}{ \alignedspace@left }
    \ams@start@box{#1}\bgroup
        \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
        \spread@equation
        \ialign\bgroup
            \hfil\strut@$\m@th\displaystyle##
```

This save the cell and then finalize it.

```
            \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
            $
            \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
            \hfil
            \crcr
    \ams@return@opt@arg
}{%
  \endaligned
}
```

`\endaligned`

```
\def\endaligned
 {
    \relax
    \ifnum\column@>\z@
       \math@cr@@@
    \else
       \crcr
    \fi
      \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}
    \egroup
    \restorecolumn@
  \egroup
  \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
 }
```

(*End of definition for* `\endaligned`. *This function is documented on page* ??.)

2.3.5 multline and multline*

`multline*` needs a redefinition before it is registered as it uses `\endmultline`, this must be replaced by the true code.

multline* (*env.*)

```
\renewenvironment{multline*}{\start@multline\st@rredtrue}
 {%
  \iftagsleft@ \@xp\lendmultline@ \else \@xp\rendmultline@ \fi
  \ignorespacesafterend
 }
```

And now we register both

```
\math_register_env:n {multline}
\math_register_env:n {multline*}
```

In the internal commands we have to add sockets for alignment attributes

\multline@

```
\def\multline@#1{%
    \Let@
    \@display@init{\global\advance\row@\@ne \global\dspbrk@lvl\m@ne}%
    \chardef\dspbrk@context\z@
    \restore@math@cr
    \let\tag\tag@in@align
    \global\tag@false \global\let\raise@tag\@empty
    \mmeasure@{#1}%
    \let\tag\gobble@tag \let\label\@gobble
    \tabskip \if@fleqn \@mathmargin \else \z@skip \fi
    \totwidth@\displaywidth
    \if@fleqn
        \advance\totwidth@-\@mathmargin
    \fi
    \halign\bgroup
        \hbox to\totwidth@{%
            \if@fleqn
                \hskip \@centering \relax
            \else
                \hfil
            \fi
            \strut@
            $\m@th\displaystyle{}##\endmultline@math
            \hfil
        }%
        \crcr
        \if@fleqn
            \hskip-\@mathmargin
            \def\multline@indent{\hskip\@mathmargin}% put it back
        \else
            \hfilneg
            \def\multline@indent{\hskip\multlinegap}%
        \fi
        \iftagsleft@
            \iftag@
                \begingroup
                    \ifshifttag@
                        \rlap{\vbox{%
                                \normalbaselines
                                \hbox{%
```

```
                                    \strut@
                                    \make@display@tag
                                }%
                                \vbox to\lineht@{}%
                                \raise@tag
                            }}%
                            \multline@indent
                        \else
                            \setbox\z@\hbox{\make@display@tag}%
                            \dimen@\@mathmargin \advance\dimen@-\wd\z@
                            \ifdim\dimen@<\multlinetaggap
                              \dimen@\multlinetaggap
                            \fi
                            \box\z@ \hskip\dimen@\relax
                        \fi
                    \endgroup
                \else
                    \multline@indent
                \fi
            \else
                \multline@indent
            \fi
        \ifmeasuring@ \else
          \UseTaggingSocket{math/luamml/mtable/aligncol} {left}
        \fi
        #1%
        \ifmeasuring@ \else
          \UseTaggingSocket{math/luamml/mtable/aligncol} {right}
        \fi
}
```

(*End of definition for* `\multline@`. *This function is documented on page* **??**.)

Luckily, `multline` uses `\endmultline@math` in exactly the spot where we have to set the flag. Less luckily, `\endmultline@math` sometimes get overwritten for the last line. But that isn't much of a problem since we want special behavior there anyway.

`\endmultline@math`

```
\def\endmultline@math
 {
   \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
   $
   \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
 }
```

(*End of definition for* `\endmultline@math`. *This function is documented on page* **??**.)

`\rendmultline@`

```
\def\rendmultline@{%
    \iftag@
```

we need to surround the math token with tagging sockets.

```
        \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
        $
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
```

```
        \let\endmultline@math\relax
            \ifshifttag@
                \hskip\multlinegap
                \llap{\vtop{%
                    \raise@tag
                    \normalbaselines
                    \setbox\@ne\null
                    \dp\@ne\lineht@
                    \box\@ne
                    \hbox{\strut@\make@display@tag}%
                }}%
            \else
                \hskip\multlinetaggap
                \make@display@tag
            \fi
```

Here we set the tag TODO: is that sensible with multline? Where is the tag saved?

```
    \UseTaggingSocket{math/luamml/mtable/tag/set}
    \else
        \hskip\multlinegap
    \fi
    \hfilneg
        \math@cr
```

Now we finalize the mtable.

```
        \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multline}
  \egroup
  \dollardollar@end
}
```

(*End of definition for* `\rendmultline@`*. This function is documented on page* **??**.)

And something similar for the left version

`\lendmultline@`

```
\def\lendmultline@{%
        \hfilneg
        \hskip\multlinegap
        \math@cr
        \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multline}
    \egroup
    \dollardollar@end
}
```

(*End of definition for* `\lendmultline@`*. This function is documented on page* **??**.)

2.4 Cases

`env@cases`

```
\def\env@cases{%
  \let\@ifnextchar\new@ifnextchar
  \left\lbrace
  \def\arraystretch{1.2}%
  \array{@{}l@{\quad}l@{\luamml_ignore:}}%
}
```

(*End of definition for* `env@cases`*. This function is documented on page* **??**.)

2.4.1 smallmatrix

`smallmatrix` (*env.*)

```
\renewenvironment {smallmatrix} {
  \UseTaggingSocket{ math/luamml/annotate/false } {} { \null\, }
  \vcenter \bgroup
    \Let@
    \restore@math@cr
    \default@tag
    \baselineskip 6 \ex@
    \lineskip 1.5 \ex@
    \lineskiplimit \lineskip
    \ialign \bgroup
        \hfil
        $
        \m@th
        \scriptstyle
        ##
        % No \scriptsize here since we want to add the mstyle nodes
        \UseTaggingSocket{math/luamml/save/nn}{ {} {mtd} }
        $
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
        \hfil
      &&
        \thickspace
        \hfil
        $
        \m@th
        \scriptstyle
        ##
        % No \scriptsize here since we want to add the mstyle nodes
        \UseTaggingSocket{math/luamml/save/nn}{ {} {mtd} }
        $
        \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
        \hfil
      \crcr
}{%
      \crcr
      \UseExpandableTaggingSocket{math/luamml/mtable/smallmatrix/save}
    \egroup
  \egroup
  \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
  \UseTaggingSocket{ math/luamml/annotate/false } {}{ \, }
}
```

2.5 The split environment

The split environment is not trivial to handle as it has a rather convoluted implementation in amsmath: depending on in which display environment it is embedded it takes different branches, which makes it difficult to finalize the mtable.

The following patches work with leqno and reqno if the (default) `centertags` are used. The currently fail (the structure is wrong) if the option `tbtags` is used. The

alignment of the cells is currently not handled. A simple debug command until everything is sorted.

__math_split_debug_typeout:n

```
\cs_new:Npn\__math_split_debug_typeout:n#1{\use_none:n{#1}}
```

(*End of definition for* `\__math_split_debug_typeout:n`.)

We need to detect if `\gather@split` has been used or not

\l__math_gathersplit_bool

```
\bool_new:N\l__math_gathersplit_bool
```

(*End of definition for* `\l__math_gathersplit_bool`.)

At first a redefinition of the main environment. Here we only have to add the saving command for the inner table:

split (*env.*)

```
\renewenvironment{split}{%
  \__math_split_debug_typeout:n {begin~split}
  \if@display
    \ifinner
       \@xp\@xp\@xp\split@aligned
    \else
      \ifst@rred \else \global\@eqnswtrue \fi
    \fi
  \else
    \let\endsplit\@empty \@xp\collect@body\@xp\split@err
  \fi
  \collect@body\gather@split
}{\__math_split_debug_typeout:n{end~split}%
     \crcr
    \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}%
    \egroup
  \egroup
  \iftagsleft@ \@xp\lendsplit@ \else \@xp\rendsplit@ \fi
}
```

In `\gather@split` we have to add the finalization socket. We also set the boolean to true so that we can detect if the finalization has already happened. Perhaps this could be done in the luamml code instead?

\gather@split

```
\def\gather@split#1#2#3{
  \__math_split_debug_typeout:n{gather@split}%
  \@xp\endgroup \reset@equation % math@cr will handle equation numbering
  \iftag@
     \toks@\@xp{\df@tag}%
     \edef\split@tag{%
       \gdef\@nx\df@tag{\the\toks@}%
       \global\@nx\tag@true \@nx\nonumber
     }%
  \else \let\split@tag\@empty
  \fi
  \bool_set_true:N\l__math_gathersplit_bool
```

```
  \spread@equation
  \vcenter\bgroup
    \gather@{\split@tag
    \begin{split}#1\end{split}}%
    \def\endmathdisplay@a{%
      \__math_split_debug_typeout:n{endmathdisplay@a}
      \__math_split_debug_typeout:n{finalize~innertable~endmathdisplay@a}
      \math@cr
      \black@ \totwidth@
      \egroup
      \egroup
      \UseExpandableTaggingSocket{math/luamml/mtable/innertable/finalize}%
    }%
}
```

(*End of definition for* `\gather@split`*. This function is documented on page* **??**.)

`\insplit@` In `\insplit@` we have to add the sockets which store the cell content.

```
\def\insplit@{\__math_split_debug_typeout:n{insplit@}%
  \global\setbox\z@\vbox\bgroup
    \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
    \default@tag % disallow use of \tag here
    \ialign\bgroup
      \hfil
      \strut@
      $\m@th\displaystyle {##}%
      \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
      $%
      \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
      &$\m@th\displaystyle {{}##}%
      \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
      $%
      \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
      \hfill % Why not \hfil?---dmj, 1994/12/28
      \crcr
}
```

(*End of definition for* `\insplit@`*. This function is documented on page* **??**.)

And now the difficult part. Depending on the options leqno/reqno `\lendsplit@` or `\rendsplit@` are used for the typesetting and the inner table must be finalized here in case this hasn't happen yet. This must be tested with the boolean from `\gather@split`

```
\def\lendsplit@{%
    \global\setbox9\vtop{\unvcopy\z@}%
    \ifinalign@
        \setbox\@ne\vbox{%
            \unvcopy\z@
            \global\setbox8\lastbox
        }%
        \setbox\@ne\hbox{%
            \unhcopy8%
            \unskip
            \setbox\tw@\lastbox
            \unskip
            \global\setbox\thr@@\lastbox
```

```
      }%
      \__math_split_debug_typeout:n{lendsplit@/aligncase}
      \ifctagsplit@
       \__math_split_debug_typeout:n{lendsplit@/aligncase/centertags}
         \gdef\split@{%
             \hbox to\wd\thr@@{}%
            &\vcenter{\vbox{\moveleft\wd\thr@@\box9}}%
            \__math_split_debug_typeout:n{finalize~innertable~aligncase}
            \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
         }%
      \else
      %TODO, not correct yet.
       \__math_split_debug_typeout:n{lendsplit@/aligncase/tbtags}
         \gdef\split@{%
             \hbox to\wd\thr@@{}%
            &\vbox{\moveleft\wd\thr@@\box9}%
            \__math_split_debug_typeout:n{finalize~innertable~aligncase}
            \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
         }%
      \fi
   \else
```

If not in align we need to test for `\gather@split`

```
      \ifctagsplit@
         \bool_if:NTF \l__math_gathersplit_bool
         {
           \__math_split_debug_typeout:n{lendsplit/equationcase/centertags}
           \gdef\split@%
            {\UseTaggingSocket{math/luamml/annotate/false}{}{\vcenter{\box9}}}
         }
         {
           \__math_split_debug_typeout:n {lendsplit/gathercase/centertags}
           \gdef\split@{\vcenter{\box9}%
           \__math_split_debug_typeout:n {finalize~innertable~gathercase}
           \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
         }
      \else
        % TODO tbtags not correct yet.
         \bool_if:NTF \l__math_gathersplit_bool
         {
           \__math_split_debug_typeout:n {lendsplit/equationcase/tbtags}
           \gdef\split@%
            {\UseTaggingSocket{math/luamml/annotate/false}{}{\box9}}
         }
         {
           \__math_split_debug_typeout:n {lendsplit/gathercase/tbtags}
           \gdef\split@{
             \box9%
             \__math_split_debug_typeout:n {finalize~innertable~gathercase}
             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
         }
      \fi
   \fi
   \aftergroup\split@
```

```
}
```

\rendsplit@ And more or less the same for the \rendsplit@ environment.

```
\def\rendsplit@{%
    \ifinalign@
        \global\setbox9 \vtop{%
            \unvcopy\z@
            \global\setbox8 \lastbox
            \unskip
        }%
        \setbox\@ne\hbox{%
            \unhcopy8
            \unskip
            \global\setbox\tw@\lastbox
            \unskip
            \global\setbox\thr@@\lastbox
        }%
        \ifctagsplit@
            \gdef\split@{%
                \hbox to\wd\thr@@{}
                &\vcenter{\vbox{\moveleft\wd\thr@@\boxz@}}%
            \__math_split_debug_typeout:n {rendsplit/aligncase/centertags}
            \__math_split_debug_typeout:n {finalize~innertable~aligncase}
            \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
            }%
        \else
           \__math_split_debug_typeout:n{rendsplit@/aligncase/tbtags}
           %TODO tbtags is not correct yet
            \global\setbox7 \hbox{\unhbox\tw@\unskip}%
            \gdef\split@{%
                \global\@tempcnta\column@
               &\setboxz@h{}%
                \savetaglength@
                \global\advance\row@\@ne
                \vbox{\moveleft\wd\thr@@\box9}%
               \crcr
                \noalign{\global\lineht@\z@}%
                \add@amps\@tempcnta
                \UseTaggingSocket{math/luamml/annotate/false}{}{\box\thr@@}
               &\box7
                 \__math_split_debug_typeout:n {finalize~innertable~aligncase}
                 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
            }%
        \fi
```

and again the if we are not in align we need to test for \gathersplit

```
    \else
        \ifctagsplit@
          \bool_if:NTF \l__math_gathersplit_bool
            {
              \__math_split_debug_typeout:n {rendsplit/equationcase/centertags}
              \gdef\split@%
               {\UseTaggingSocket{math/luamml/annotate/false}{}{\vcenter{\boxz@}}}
            }
```

```
            {
              \__math_split_debug_typeout:n  {rendsplit/gathercase/centertags}
              \gdef\split@{\vcenter{\boxz@}%
               \__math_split_debug_typeout:n {finalize~innertable~gathercase}
               \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
            }
        \else
           \bool_if:NTF \l__math_gathersplit_bool
            {
              \__math_split_debug_typeout:n {rendsplit/equationcase/tbtags}
              \gdef\split@{%
                \UseTaggingSocket{math/luamml/annotate/false}{}{\boxz@}}
            }
            {
              \__math_split_debug_typeout:n {rendsplit/gathercase/tbtags}
              \gdef\split@{%
                \boxz@
                \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
            }
        \fi
    \fi
    \aftergroup\split@
}
```

(*End of definition for* `\rendsplit@`. *This function is documented on page* ??.)

2.6 \intertext

The `\intertext` command errors with active tagging as it is processed twice which leads to duplicated structures.

`\intertext@`

```
\NewTaggingSocket  {math/intertext}{2}
\NewTaggingSocketPlug {math/intertext}{kernel}
 {
  \tag_mc_end_push:
  \tagpdfparaOn
  #2
  \tagpdfparaOff
  \tag_mc_begin_pop:n{}
 }
\AssignTaggingSocketPlug{math/intertext}{kernel}
```

This can go once luamml defines the plug directly

```
\IfSocketPlugExistsF {math/intertext}{luamml}
 {
   \int_new:N \l__luamml_struct_tmpa_int
   \NewTaggingSocketPlug{math/intertext}{luamml}
     {
       \tag_mc_end_push:
        \tag_struct_begin:n{
          tag = mtext,
          stash,
        }
```

```
        \int_set:Nn \l__luamml_struct_tmpa_int { \tag_get:n{struct_num} }
        \tag_mc_begin:n{}
        #2
        \tag_mc_end:
        \tag_struct_end:
        \__luamml_amsmath_save_intertext_with_struct_elem:N \l__luamml_struct_tmpa_int
        \tag_mc_begin_pop:n{}
     }
  \AddToHook{package/luamml/after}{\AssignTaggingSocketPlug{math/intertext}{luamml}}
 }

\def\intertext@{%
  \def\intertext##1{%
    \ifvmode\else\\\@empty\fi
    \noalign{%
      \penalty\postdisplaypenalty\vskip\belowdisplayskip
      \vbox{
```

Stop tagging when measuring:

```
       \ifmeasuring@\tag_suspend:n{\measuring}\fi
       \normalbaselines
        \ifdim\linewidth=\columnwidth
        \else \parshape\@ne \@totalleftmargin \linewidth
        \fi
```

Use the tagging socket

```
       \UseTaggingSocket{math/intertext}{}
        {
        \noindent\ignorespaces##1\par
        }
      }% end vbox
      \penalty\predisplaypenalty\vskip\abovedisplayskip%
    }%
}}
```

(*End of definition for* `\intertext@`. *This function is documented on page* ??.)

2.7 \text

The `\text` command uses `\mathchoice` which "typesets" the argument four times. This makes it quite problematic for tagging. Without precautions structure objects would be created four times and would get MC-chunks as kids that don't really exist. amsmath contains a switch that allows to execute code only in the first (displaymath) branch, but that isn't usable here. At first because we don't know if the first branch creates the same structure as the one that is actually used. At second because the engines executes some commands like `\label` and `\pdfannot` only at shipout from the branch that really was used. So we would get structure data from one `\mathchoice`-branch and MC-labels and links from another one and that gets very messy.

We therefore have to avoid that tagging is active in unused branches. In pdflatex it is not possible to detect the mathstyle before, so we use a label. With lualatex is is possible to redefine `\text` not to use `\mathchoice`

`\text@`

```
\AddToHook{package/amstext/after}
```

```
{
```

currently amsmath is loaded in a begindocument hook, so this test is fine. If amstext is loaded earlier (in the kernel), this needs perhaps a change.

```
  \sys_if_engine_luatex:TF
   {
     \def\text@#1{{
       \tag_socket_use:nnn {math/luamml/hbox}{}
      {{%
       \ifcase\mathstyle
       \hbox{{#1}}\or
       \hbox{{#1}}\or
       \hbox{{#1}}\or
       \hbox{{#1}}\or
       \hbox{{\let\f@size\sf@size\selectfont#1}}\or
       \hbox{{\let\f@size\sf@size\selectfont#1}}\or
       \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
       \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
       \ERROR
       \fi
       \check@mathfonts
     }}}}
   }
   {
     \tag_if_active:T
      {
        \def\text@#1
         {{
          \int_gincr:N\g__math_mathchoice_int
          \tag_suspend:n{\text@}
          \mathchoice
           {
            \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{0}
            \textdef@\displaystyle\f@size{#1}
           }
           {
            \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{2}
            \textdef@\textstyle\f@size{\firstchoice@false #1}
           }
           {
            \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{4}
            \textdef@\textstyle\sf@size{\firstchoice@false #1}
           }
           {
             \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{6}
             \textdef@\textstyle \ssf@size{\firstchoice@false #1}
           }
           \check@mathfonts
         }}
     }
   }
}
```

(*End of definition for* `\text@`. *This function is documented on page* **??**.)

2.8 \pmb

\pmb prints its argument three times. For tagging we must mark two of occurrences as artifact. For luatex the attributes in the box must be reset, for this we switch to expl3-boxes.

\pbm@@ \pmb@

TODO: this is wrong with mathml-SE tagging (see also the related bm code in latex-lab-mathpkg).

```
\cs_new:Npn\__math_patch_amsbsy_pmb:
 {
   \NewTaggingSocket{amsbsy/pmb}{2}
   \NewTaggingSocketPlug{amsbsy/pmb}{default}
    {
      \tag_mc_end:\tag_mc_begin:n{artifact}
      \tag_mc_reset_box:N##1
      ##2
      \tag_mc_end:
      \tag_mc_begin:n{}
    }
   \AssignTaggingSocketPlug{amsbsy/pmb}{default}
   \def\pmb@@##1##2##3{\leavevmode\hbox_set:Nn\l__math_tmpa_box{##3}
      \dimen@-\box_wd:N\l__math_tmpa_box
      \kern-.5\ex@\box_use:N\l__math_tmpa_box
      \UseTaggingSocket{amsbsy/pmb}{\l__math_tmpa_box}
        {
          \kern\dimen@\kern.25\ex@\raise.4\ex@\box_use:N\l__math_tmpa_box
          \kern\dimen@\kern.25\ex@\box_use_drop:N\l__math_tmpa_box
        }
    }
   \def\pmb@##1##2{\hbox_set:Nn\l__math_tmpa_box{$\m@th##1{##2}$}
     \setboxz@h{$\m@th##1\mkern.5mu$}\pmbraise@\wdz@
     \binrel@{##2}
     \dimen@-\box_wd:N\l__math_tmpa_box
     \binrel@@{
       \mkern-.8mu\box_use:N\l__math_tmpa_box
       \UseTaggingSocket{amsbsy/pmb}{\l__math_tmpa_box}
        {
          \kern\dimen@\mkern.4mu\raise\pmbraise@\box_use:N\l__math_tmpa_box
          \kern\dimen@\mkern.4mu\box_use_drop:N\l__math_tmpa_box
        }
       }
    }
  }
\AddToHook{package/amsbsy/after}
  {
    \__math_patch_amsbsy_pmb:
  }
```

(*End of definition for* \pbm@@ *and* \pmb@. *These functions are documented on page* **??**.)

```
\ExplSyntaxOff
```

⟨/kernel⟩

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols

\, ... 413, 451
\\ 688

A

\abovedisplayskip 702
\AddToHook 6, 684, 705, 790
\advance 139, 294, 304, 341, 614
\aftergroup 582, 654
\array 410
\arraystretch 409
\AssignTaggingSocketPlug . . 665, 684, 766

B

\baselineskip 418
\begin 206, 254, 489
\begingroup 327
\belowdisplayskip 690
\bgroup 162, 210, 224, 257, 260, 306, 414, 421, 487, 501, 504
bool commands:
\bool_if:NTF 553, 567, 627, 640
\bool_new:N 454
\bool_set_true:N 485
bool internal commands:
\l__math_gathersplit_bool 454, 485, 553, 567, 627, 640
\box 345, 381, 537, 546, 557, 561, 571, 576, 615, 619, 620
box commands:
\box_use:N 769, 772, 781, 784
\box_use_drop:N 773, 785
\box_wd:N 768, 779
box internal commands:
\l__math_tmpa_box 767, 768, 769, 770, 772, 773, 776, 779, 781, 782, 784, 785

C

\chardef 155, 220, 258, 295, 502
\columnwidth 694
\cr 150
\crcr 178, 249, 266, 277, 317, 444, 446, 468, 516, 616
cs commands:
\cs_new:Npn 453, 755

D

\DeclareRobustCommand 23
\def 8, 10, 22, 25, 93, 136, 152, 181, 271, 292, 320, 323, 362, 368, 398, 406, 409, 474, 490, 500, 518, 584, 686, 687, 709, 729, 767, 776
\displaystyle . 100, 104, 119, 123, 166, 167, 230, 232, 239, 244, 261, 262, 314, 364, 370, 507, 508, 511, 512, 736
\displaywidth 162, 302
\dp 380

E

\edef 479
\egroup 28, 31, 185, 197, 280, 282, 395, 403, 448, 449, 470, 471, 495, 496
\else . . . 14, 34, 103, 110, 122, 129, 148, 177, 205, 216, 253, 276, 287, 301, 310, 321, 339, 348, 351, 354, 358, 384, 389, 460, 461, 463, 472, 483, 541, 551, 565, 606, 625, 639, 688, 695
\end 489
\endaligned 269, 271
\endgather *6*, 181
\endgroup 347, 476
\endmultline *8*
\endsplit 464
environments:
gather* 189
gathered 252
multline* 285
smallmatrix 412
split 455
\eqref *2*, 22
\ERROR 721
\ExplSyntaxOff 794
\ExplSyntaxOn 5

F

\fi . 20, 33, 36, 105, 112, 124, 131, 137, 138, 146, 148, 177, 207, 219, 255, 278, 287, 301, 305, 312, 324, 344, 346, 350, 353, 356, 360, 387, 391, 461, 462, 465, 472, 484, 550, 580, 581, 624, 652, 653, 688, 692, 696, 722

G

gather* (env.) 189
gathered (env.) 252
\gathersplit *16*

\gdef 480, 535, 544, 556, 561, 570, 575, 599, 610, 630, 635, 643, 648
\global 138, 139, 148, 149, 159, 161, 223, 294, 298, 461, 481, 501, 519, 523, 530, 586, 588, 594, 596, 609, 611, 614, 617

H

\halign 162, 306
\hbox 13, 16, 22, 307, 331, 340, 382, 525, 536, 545, 591, 600, 609, 713, 714, 715, 716, 717, 718, 719, 720
hbox commands:
\hbox_set:Nn 767, 776
\hfil 94, 133, 226, 247, 261, 265, 311, 315, 422, 431, 434, 443, 505, 515
\hfill 515
\hfilneg 322, 392, 399
\hskip 309, 319, 320, 323, 345, 375, 385, 390, 400

I

\ialign 224, 260, 421, 504
\ifcase 712
\ifdim 342, 694
\iffalse 33
\ifinner 458
\ifnum 33, 212, 274
\IfSocketPlugExistsF 666
\iftrue 8
\ifvmode 688
\ignorespaces 699
\ignorespacesafterend . . 37, 187, 199, 288
int commands:
\int_gincr:N 731
\int_new:N 668
\int_set:Nn 676
\int_use:N 735, 739, 743, 747
int internal commands:
\g__math_mathchoice_int 731, 735, 739, 743, 747
\intertext *17*, 687

K

\kern 142, 769, 772, 773, 784, 785

L

\label *18*, 154, 300
\lastbox 523, 528, 530, 588, 594, 596
\lbrace 408
\leavevmode 767
\left 408
\let 8, 24, 153, 154, 157, 214, 217, 297, 298, 300, 373, 407, 464, 483, 717, 718, 719, 720
\lineskip 419, 420
\lineskiplimit 420
\linewidth 694, 695
\llap 376
luamml commands:
\luamml_ignore: 8, 102, 121, 410
luamml internal commands:
__luamml_amsmath_save_intertext_-with_struct_elem:N 681
\l__luamml_struct_tmpa_int 668, 676, 681

M

math commands:
\math_register_env:n . . 84, 85, 86, 87, 88, 89, 90, 91, 92, 201, 202, 290, 291
math internal commands:
__math_patch_amsbsy_pmb: . 755, 792
__math_split_debug_typeout:n 453, 453, 456, 467, 475, 491, 492, 500, 532, 534, 538, 543, 547, 555, 560, 562, 569, 574, 577, 602, 603, 607, 621, 629, 634, 636, 642, 647
__math_tag_if_mathstyle:nn 735, 739, 743, 747
\mathchoice *18*, 733
\mathstyle 712
\measuring 8, 692
\minalignsep 218
\mkern 777, 781, 784, 785
\moveleft 537, 546, 601, 615
\multiply 213
multline* (env.) 285
\multlinegap 323, 375, 390, 400
\multlinetaggap 342, 343, 385

N

\NewTaggingSocket 656, 757
\NewTaggingSocketPlug 657, 669, 758
\noalign 617, 689
\noindent 699
\nonumber 32, 137, 481
\normalbaselines 330, 378, 693
\normalfont 13, 16, 22
\null 379, 413

O

\omit 141
\or . . 713, 714, 715, 716, 717, 718, 719, 720

P

\par 699
\parshape 695
\pdfannot *18*
\penalty 690, 702

\pmb *20*
\postdisplaypenalty 690
\predisplaypenalty 702
\ProvidesFile 3

Q

\quad 410

R

\raise 772, 784
\ref 24
\relax 211, 273, 309, 345, 373
\renewcommand 203
\renewenvironment . . 39, 44, 49, 54, 59, 64, 69, 74, 79, 189, 252, 285, 412, 455
\rlap 329

S

\scriptsize 427, 439
\scriptstyle 425, 437
\selectfont 717, 718, 719, 720
\setbox 340, 379, 501, 519, 521, 523, 525, 528, 530, 586, 588, 591, 594, 596, 609
smallmatrix (env.) 412
\span 177
split (env.) 455
\split 153
sys commands:
\sys_if_engine_luatex:TF 707

T

\tabskip 114, 134, 160, 173, 177, 235, 248, 301
\tag 154, 297, 300, 503
tag commands:
\tag_get:n 676
\tag_if_active:TF 727
\tag_mc_begin:n 677, 760, 764
\tag_mc_begin_pop:n 663, 682
\tag_mc_end: 679, 760, 763
\tag_mc_end_push: 659, 671
\tag_mc_reset_box:N 761
\tag_socket_use:nnn 710
\tag_struct_begin:n 672
\tag_struct_end: 680
\tag_suspend:n 8, 692, 732
\tagpdfparaOff 662
\tagpdfparaOn 660
TEX and LATEX 2ε commands:
\@centering 173, 309
\@currenvir 27, 206
\@display@init 294
\@empty 298, 464, 483, 688
\@eqnswtrue 148, 461
\@gobble 300
\@ifnextchar 407
\@lign 98, 117, 144
\@mathmargin . . . 301, 304, 319, 320, 341
\@ne 40, 45, 70, 75, 139, 161, 220, 258, 294, 379, 380, 381, 502, 521, 525, 591, 614, 695
\@nx 480, 481
\@tempcnta 611, 618
\@totalleftmargin 695
\@xp 287, 459, 464, 472, 476, 478
\add@amps 140, 618
\align@ *4*
\align@preamble 93
\alignedspace@left 209, 256
\alignsep@ 134, 142, 215, 218, 248
\ams@return@opt@arg 250, 267
\ams@start@box 210, 257
\binrel@ 778
\binrel@@ 780
\black@ 26, 183, 195, 494
\boxz@ 601, 631, 635, 644, 649
\calc@shift@gather 171
\check@mathfonts 723, 750
\collect@body 464, 466
\column@ 223, 274, 611
\column@plus 225, 236
\common@align@ending 25, 42, 47, 52, 57, 62, 67, 72, 77, 82
\default@tag 221, 417, 503
\df@tag 478, 480
\dimen@ 341, 342, 343, 345, 768, 772, 773, 779, 784, 785
\displ@y@ 156
\dollardollar@end 35, 186, 198, 396, 404
\dspbrk@context 155, 220, 258, 295, 502
\dspbrk@lvl 294
\endmathdisplay@a 490
\endmultline@math . . *10*, 314, 362, 373
\env@cases 406
env@cases 406
\ex@ 418, 419, 769, 772, 773
\f@size . . . 717, 718, 719, 720, 736, 740
\firstchoice@false 740, 744, 748
\gather@ 152, 488
\gather@split *13–15*, 466, 474
\gdisplaywidth@ 177
\gmeasure@ 158
\gobble@tag 300
\if@display 457
\if@eqnsw 138
\if@fleqn 301, 303, 308, 318
\ifctagsplit@ 533, 552, 598, 626
\ifinalign@ 520, 585
\ifingather@ 29

\ifmeasuring@ 8, 12, 101, 108, 120, 127, 354, 358, 692
\ifshifttag@ 328, 374
\ifst@rred 137, 148, 461
\iftag@ 143, 326, 369, 477
\iftagsleft@ 177, 287, 325, 472
\ingather@true 153
\insplit@ *14*, 153, 500
\intertext@ 156, 656
\label@in@display 154
\lendmultline@ 287, 398
\lendsplit@ *14*, 472, 518
\Let@ 156, 220, 258, 293, 415, 502
\lineht@ 149, 335, 380, 617
\m@ne 40, 45, 60, 65, 212, 294
\m@th 13, 16, 22, 100, 119, 166, 229, 238, 261, 314, 424, 436, 507, 511, 776, 777
\make@display@tag 144, 333, 340, 382, 386
\maketag@@@ *2*, 10, 24
\maketag@@@notag 22, 24
\maketag@@@notog 22
\math@cr . . . 26, 182, 194, 393, 401, 493
\math@cr@@@ 157, 214, 217, 275
\math@cr@@@align 136
\math@cr@@@aligned 217
\math@cr@@@alignedat 214
\math@cr@@@gather 157
\maxfields@ 140, 211, 212, 213
\measuring@true 6
\mmeasure@ 299
\multline@ 292
\multline@indent 320, 323, 338, 349, 352
\new@ifnextchar 407
\nonmatherr@ 206, 254
\pbm@@ 755
\place@tag 145
\place@tag@gather 175
\pmb@ 755
\pmb@@ 767
\pmbraise@ 777, 784
\raise@tag 298, 336, 377
\rendmultline@ 287, 368
\rendsplit@ *14*, *16*, 472, 584
\reset@equation 476
\restore@math@cr . . 258, 296, 416, 502
\restorealignstate@ 30
\restorecolumn@ 281
\RIfM@ 204, 253
\row@ 139, 161, 294, 614
\savecolumn@ 208
\savefieldlength@ 109, 128
\savetaglength@ 613
\set@field 113, 132
\set@gather@field 172
\setboxz@h 96, 115, 144, 164, 174, 612, 777
\sf@size 717, 718, 744
\shifttag@false 159
\split@ 535, 544, 556, 561, 570, 575, 582, 599, 610, 630, 635, 643, 648, 654
\split@aligned 459
\split@err 464
\split@tag 479, 483, 488
\spread@equation 222, 259, 486
\ssf@size 719, 720, 748
\st@rredfalse 40, 50, 60, 70
\st@rredtrue 45, 55, 65, 75, 80, 191, 285
\start@align 40, 45, 50, 55, 60, 65, 70, 75, 80
\start@aligned 203
\start@gather 191
\start@multline 285
\strut@ 95, 144, 163, 174, 227, 261, 313, 332, 382, 506
\tag@false 298
\tag@in@align 154, 297
\tag@true 138, 481
\tagform@ *2*, 24
\text@ 705
\textdef@ 736, 740, 744, 748
\thr@@ 530, 536, 537, 545, 546, 596, 600, 601, 615, 619
\toks@ 478, 480
\totwidth@ 26, 183, 195, 302, 304, 307, 494
\tw@ 60, 65, 80, 213, 528, 594, 609
\wdz@ 777
\z@ . . 50, 55, 149, 155, 223, 274, 295, 340, 341, 345, 501, 519, 522, 587, 617
\z@skip . . . 114, 160, 177, 215, 235, 301
\text *18*
\textstyle 740, 744, 748
\textup 24
\the 480
\thickspace 433

U

\unhbox 609
\unhcopy 526, 592
\unskip 527, 529, 589, 593, 595, 609
\unvcopy 519, 522, 587
use commands:
\use_none:n 453
\UseExpandableTaggingSocket 27, 184, 196, 279, 394, 402, 447, 469, 497

\UseTaggingSocket 15, 17, 19, 104, 111, 123, 130, 147, 167, 170, 176, 209, 232, 234, 244, 246, 256, 262, 264, 283, 355, 359, 364, 366, 370, 372, 388, 413, 428, 430, 440, 442, 450, 451, 508, 510, 512, 514, 539, 548, 557, 563, 571, 578, 604, 619, 622, 631, 637, 644, 650, 697, 770, 782

V

\vbox 329, 335, 501, 521, 537, 546, 601, 615, 691
\vcenter 414, 487, 537, 557, 561, 601, 631, 635
\vskip 690, 702
\vtop 376, 519, 586

W

\wd . . 341, 536, 537, 545, 546, 600, 601, 615