

The latex-lab-amsmath code*

L^AT_EX Project

February 22, 2025

Abstract

Contents

1	Introduction	1
2	The Implementation	1
2.1	File declaration	2
2.2	Tagpdf support	2
2.3	Measuring	2
2.4	Display environments	2
2.4.1	Tag	2
2.4.2	align & friends	3
2.4.3	gather and gather*	5
2.4.4	gathered, aligned and alignedat	7
2.4.5	multiline and multiline*	8
2.5	Cases	11
2.5.1	smallmatrix	11
2.6	The split environment	12
2.7	\intertext	17
2.8	\text	18
2.9	\pmb	19

Index	20
--------------	-----------

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

```
3 \ProvidesFile{latex-lab-amsmath.ltx}
4 [2025-01-31 v0.1e amsmath adaptions]
```

2.2 Tagpdf support

To make the code independent from tagging being loaded and active we load the tagpdf-base package:

```
5 \RequirePackage{tagpdf-base}
6 \ExplSyntaxOn
```

2.3 Measuring

When measuring we neither want tagging nor the luamml processing.

```
\measuring@true
7 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_suspend:n{\measuring}\luamml_ignore:}
(End of definition for \measuring@true. This function is documented on page ??.)
```

2.4 Display environments

2.4.1 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@@@
8 \def\maketag@@@#1
9   {%
10     \ifmeasuring@
11       \hbox{\m@th\normalfont#1}%
12     \else
13       \UseTaggingSocket{math/display/tag/begin}
14       \hbox{\m@th\normalfont#1
15         \UseTaggingSocket{math/luamml/mtable/tag/save}
16       }%
17       \UseTaggingSocket{math/display/tag/end}
18     \fi
19   }
```

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

```

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

2.4.2 align & friends

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

```

\common@align@ending
23 \def\common@align@ending {
24   \math@cr \black@totwidth@
25   \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {align}
26   \egroup
27   \ifingather@
28     \restorealignstate@
29     \egroup
30     \nonumber
31     \ifnum0='{ \fi \iffalse } \fi
32   \else
33     $$%
34   \fi
35   \ignorespacesafterend
36 }

```

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

Now we redefine the display alignments to use these ending.

```

37 \renewenvironment{align}{%
38   \start@align\@ne\st@rredfalse\m@ne
39 }{%
40   \common@align@ending
41 }
42 \renewenvironment{align*}{%
43   \start@align\@ne\st@rredtrue\m@ne
44 }{%
45   \common@align@ending
46 }
47 \renewenvironment{alignat}{%
48   \start@align\z@\st@rredfalse
49 }{%
50   \common@align@ending
51 }
52 \renewenvironment{alignat*}{%
53   \start@align\z@\st@rredtrue
54 }{%
55   \common@align@ending
56 }
57 \renewenvironment{flalign}{%
58   \start@align\tw@\st@rredfalse\m@ne
59 }{%
60   \common@align@ending

```

```

61 }
62 \renewenvironment{flalign*}{%
63   \start@align\tw@\st@rredtrue\m@ne
64 }{%
65   \common@align@ending
66 }
67 \renewenvironment{xalignat}{%
68   \start@align\@ne\st@rredfalse
69 }{%
70   \common@align@ending
71 }
72 \renewenvironment{xalignat*}{%
73   \start@align\@ne\st@rredtrue
74 }{%
75   \common@align@ending
76 }
77 \renewenvironment{xxalignat}{%
78   \start@align\tw@\st@rredtrue
79 }{%
80   \common@align@ending
81 }

```

And register these environments for the math grabbing.

```

82 \math_register_halign_env:nn {align}{}
83 \math_register_halign_env:nn {align*}{}
84 \math_register_halign_env:nn {alignat}{}
85 \math_register_halign_env:nn {alignat*}{}
86 \math_register_halign_env:nn {flalign}{}
87 \math_register_halign_env:nn {flalign*}{}
88 \math_register_halign_env:nn {xalignat}{}
89 \math_register_halign_env:nn {xalignat*}{}
90 \math_register_halign_env:nn {xxalignat}{}

```

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

\align@preamble

```

91 \def\align@preamble{%
92   &\hfil
93   \strut@
94   \setboxz@h
95   {
96     \@lign
97     $
98     \m@th\displaystyle{##}
99     \ifmeasuring@
100     \luamml_ignore:
101     \else
102     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
103     \fi
104     $
105   }%
106 \ifmeasuring@
107   \savefieldlength@
108 \else
109   \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}

```

```

110 \fi
111 \set@field
112 \tabskip\z@skip
113 &\setboxz@h
114 {
115 \@lign
116 $
117 \m@th\displaystyle{##}
118 \ifmeasuring@
119 \luamml_ignore:
120 \else
121 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }
122 \fi
123 $
124 }%
125 \ifmeasuring@
126 \savefieldlength@
127 \else
128 \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
129 \fi
130 \set@field
131 \hfil
132 \tabskip\alignsep@
133 }

```

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

`\math@cr@@@align`

```

134 \def\math@cr@@@align{%
135 \ifst@rred\nonumber\fi
136 \if@eqnsw \global\tag@true \fi
137 \global\advance\row@ \@one
138 \add@amps\maxfields@
139 \omit
140 \kern-\alignsep@
141 \iftag@
142 \setboxz@h{\@lign\strut@{\make@display@tag}}%
143 \place@tag
144 \fi
145 \UseTaggingSocket{math/luamml/mtable/tag/set}
146 \ifst@rred\else\global\@eqnswtrue\fi
147 \global\lineht@\z@
148 \cr
149 }

```

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

2.4.3 gather and gather*

`\gather@`

```

150 \def\gather@#1{%
151 \ingather@true \let\split\insplit@
152 \let\tag\tag@in@align \let\label\label@in@display
153 \chardef\dspbrk@context\z@
154 \intertext@ \disply@ \Let@

```

```

155 \let\math@cr@@\math@cr@@gather
156 \gmeasure@{#1}%
157 \global\shifttag@false
158 \tabskip\z@skip
159 \global\row@\@ne
160 \halign to\displaywidth\bgroup
161   \strut@
162   \setboxz@h
163   {
164     $\m@th\displaystyle{##}
165     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
166     $
167   }%
168   \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
169   \calc@shift@gather
170   \set@gather@field
171   \tabskip\@centering
172   &\setboxz@h{\strut@{##}}%
173   \place@tag@gather
174   \UseTaggingSocket{math/luamml/mtable/tag/set}
175   \tabskip \iftagsleft@ \gdisplaywidth@ \else \z@skip \span\fi
176   \ccr
177   #1%
178 }

```

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

`\endgather`

```

179 \def\endgather{
180   \math@cr
181   \black@ \totwidth@
182   \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
183   \egroup
184   $$
185   \ignorespacesafterend
186 }

```

(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.*)

```

187 \renewenvironment{gather*}
188 {
189   \start@gather\st@rredtrue
190 }
191 {
192   \math@cr
193   \black@\totwidth@
194   \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
195   \egroup
196   $$\ignorespacesafterend
197 }

```

Register both environments

```
198 \math_register_halign_env:nn {gather}{}  
199 \math_register_halign_env:nn {gather*}{}
```

2.4.4 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`

```
200 \renewcommand{\start@aligned}[2]{  
201   \RIfM@  
202   \else  
203     \nonmatherr@ {\begin{\@currenvir}}  
204   \fi  
205   \savecolumn@ % Assumption: called inside a group  
206   \UseTaggingSocket{math/luamml/annotate/false}{ }{ \alignedspace@left }  
207   \ams@start@box {#1} \bgroup  
208     \maxfields@ #2 \relax  
209     \ifnum \maxfields@ > \m@ne  
210       \multiply \maxfields@ \tw@  
211       \let \math@cr@@@ \math@cr@@@alignedat  
212       \alignsep@ \z@skip  
213     \else  
214       \let \math@cr@@@ \math@cr@@@aligned  
215       \alignsep@ \minalignsep  
216     \fi  
217     \Let@ \chardef \dspbrk@context \@ne  
218     \default@tag  
219     \spread@equation % no-op if already called  
220     \global \column@ \z@  
221     \ialign \bgroup  
222       & \column@plus  
223       \hfil  
224       \strut@  
225       $  
226       \m@th  
227       \displaystyle  
228       {##}  
229       \UseTaggingSocket{math/luamml/save/nNn}{ }{ \displaystyle {mtd} }  
230       $  
231       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}  
232       \tabskip \z@skip  
233     & \column@plus  
234     $  
235     \m@th  
236     \displaystyle  
237     {  
238       {}  
239       ##  
240     }  
241     \UseTaggingSocket{math/luamml/save/nNn}{ }{ \displaystyle {mtd} }  
242     $  
243     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
```

```

244     \hfil
245     \tabskip\alignsep@
246     \crrc
247     \ams@return@opt@arg
248 }

```

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

`gathered` (*env.*)

```

249 \renewenvironment{gathered}[1][c]{%
250   \RIfM@else
251   \nonmatherr@{\begin{gathered}}%
252   \fi

```

This annotates the space

```

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

```

This save the cell and then finalize it.

```

259   \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
260   $
261   \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
262   \hfil
263   \crrc
264   \ams@return@opt@arg
265 }{%
266 \endaligned
267 }

```

`\endaligned`

```

268 \def\endaligned
269 {
270   \crrc
271   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}
272   \egroup
273   \restorecolumn@
274   \egroup
275   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
276 }

```

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

2.4.5 `multiline` and `multiline*`

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

`multiline*` (*env.*)

```

277 \renewenvironment{multiline*}{\start@multiline\st@rredtrue}
278 {%
279   \iftagsleft@ \@xp\lendmultiline@ \else \@xp\rendmultiline@ \fi

```



```

280 \ignorespacesafterend
281 }

```

And now we register both

```

282 \math_register_halign_env:nn {multiline}{-}
283 \math_register_halign_env:nn {multiline*}{-}

```

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

`\multiline@`

```

284 \def\multiline@#1{%
285   \Let@
286   \@display@init{\global\advance\row@\@ne \global\dspbrk@l\l\m@ne}%
287   \chardef\dspbrk@context\z@
288   \restore@math@cr
289   \let\tag\tag@in@align
290   \global\tag@false \global\let\raise@tag\@empty
291   \mmeasure@{#1}%
292   \let\tag@gobble@tag \let\label\@gobble
293   \tabskip \if@fleqn \@mathmargin \else \z@skip \fi
294   \totwidth@\displaywidth
295   \if@fleqn
296     \advance\totwidth@-\@mathmargin
297   \fi
298   \halign\bgroup
299     \hbox to\totwidth@{%
300       \if@fleqn
301         \hskip \@centering \relax
302       \else
303         \hfil
304       \fi
305       \strut@
306       $\m@th\displaystyle{##}\endmultiline@math
307       \hfil
308     }%
309     \crrc
310     \if@fleqn
311       \hskip-\@mathmargin
312       \def\multiline@indent{\hskip\@mathmargin}% put it back
313     \else
314       \hfilneg
315       \def\multiline@indent{\hskip\multilinegap}%
316     \fi
317     \iftagsleft@
318       \iftag@
319         \beginngroup
320           \ifshifftag@
321             \rlap{\vbox{%
322               \normalbaselines
323               \hbox{%
324                 \strut@
325                 \make@display@tag
326               }%
327               \vbox to\lineht@{%
328                 \raise@tag

```

```

329         }}%
330         \multline@indent
331     \else
332         \setbox\z@\hbox{\make@display@tag}%
333         \dimen@\@mathmargin \advance\dimen@-\wd\z@
334         \ifdim\dimen@<\multlinetaggap
335             \dimen@\multlinetaggap
336         \fi
337         \box\z@ \hskip\dimen@\relax
338     \fi
339 \endgroup
340 \else
341     \multline@indent
342 \fi
343 \else
344     \multline@indent
345 \fi
346 \ifmeasuring@ \else
347     \UseTaggingSocket{math/luamml/mtable/aligncol} {left}
348 \fi
349 #1%
350 \ifmeasuring@ \else
351     \UseTaggingSocket{math/luamml/mtable/aligncol} {right}
352 \fi
353 }

```

(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`

```

354 \def\endmultline@math
355 {
356     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
357     $
358     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
359 }

```

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

`\rendmultline@`

```

360 \def\rendmultline@{
361     \iftag@

```

we need to surround the math token with tagging sockets.

```

362         \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
363         $
364         \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
365         \let\endmultline@math\relax
366         \ifshifttag@
367             \hskip\multlinegap
368             \llap{\vtop{
369                 \raise@tag
370                 \normalbaselines

```

```

371             \setbox\@ne\null
372             \dp\@ne\lineht@
373             \box\@ne
374             \hbox{\strut\make@display@tag}%
375         }}%
376     \else
377         \hskip\multlinetaggap
378         \make@display@tag
379     \fi

```

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

```

380     \UseTaggingSocket{math/luamml/mtable/tag/set}
381 \else
382     \hskip\multlinegap
383 \fi
384 \hfilneg
385 \math@cr

```

Now we finalize the mtable.

```

386     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
387 \egroup$$$%
388 }

```

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

And something similar for the left version

`\lendmultiline@`

```

389 \def\lendmultiline@{%
390     \hfilneg
391     \hskip\multlinegap
392     \math@cr
393     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
394 \egroup
395 $$$%
396 }

```

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

2.5 Cases

`env@cases`

```

397 \def\env@cases{%
398     \let\@ifnextchar\new@ifnextchar
399     \left\lbrace
400     \def\arraystretch{1.2}%
401     \array{@f}l@{\quad}l@{\luamml_ignore:}}%
402 }

```

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

2.5.1 smallmatrix

`smallmatrix` (*env.*)

```
403 \renewenvironment {smallmatrix} {
404   \UseTaggingSocket{ math/luamml/annotate/false } {} { \null\, }
405   \vcenter \bgroup
406     \Let@
407     \restore@math@cr
408     \default@tag
409     \baselineskip 6 \ex@
410     \lineskip 1.5 \ex@
411     \lineskiplimit \lineskip
412     \ialign \bgroup
413       \hfil
414       $
415       \m@th
416       \scriptstyle
417       ##
418       % No \scriptsize here since we want to add the mstyle nodes
419       \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
420       $
421       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
422       \hfil
423     &&
424     \thickspace
425     \hfil
426     $
427     \m@th
428     \scriptstyle
429     ##
430     % No \scriptsize here since we want to add the mstyle nodes
431     \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
432     $
433     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
434     \hfil
435   \crcr
436 }{%
437   \crcr
438   \UseExpandableTaggingSocket{math/luamml/mtable/smallmatrix/save}
439   \egroup
440   \egroup
441   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
442   \UseTaggingSocket{ math/luamml/annotate/false } {} { \, }
443 }
```

2.6 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`

```
444 \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`

```
445 \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.)`

```
446 \renewenvironment{split}{%
447   \_math_split_debug_typeout:n {begin~split}
448   \if@display
449     \ifinner
450       \@xp\@xp\@xp\split@aligned
451     \else
452       \ifst@rred \else \global\@eqnswtrue \fi
453     \fi
454   \else
455     \let\endsplit\@empty \@xp\collect@body\@xp\split@err
456   \fi
457   \collect@body\gather@split
458 }{\_math_split_debug_typeout:n{end~split}}%
459   \crrc
460   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}%
461   \egroup
462 \egroup
463 \iftagsleft@ \@xp\lendsplit@ \else \@xp\rendsplit@ \fi
464 }
```

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`

```
465 \def\gather@split#1#2#3{
466   \_math_split_debug_typeout:n{gather@split}%
467   \@xp\endgroup \reset@equation % math@cr will handle equation numbering
468   \iftag@
469     \toks@\@xp{\df@tag}%
470     \edef\split@tag{%
471       \gdef\@nx\df@tag{\the\toks@}%
472       \global\@nx\tag@true \@nx\nonumber
473     }%
474   \else \let\split@tag\@empty
475   \fi
476   \bool_set_true:N\l__math_gathersplit_bool
477   \spread@equation
```

```

478 \vcenter\bgroup
479 \gather@{\split@tag
480 \begin{split}#1\end{split}}%
481 \def\endmathdisplay@a{%
482 \_math_split_debug_typeout:n{endmathdisplay@a}
483 \_math_split_debug_typeout:n{finalize~innertable~endmathdisplay@a}
484 \math@cr
485 \black@ \totwidth@
486 \egroup
487 \egroup
488 \UseExpandableTaggingSocket{math/luamml/mtable/innertable/finalize}%
489 }%
490 }

```

(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.

```

491 \def\insplit@{\_math_split_debug_typeout:n{insplit@}}%
492 \global\setbox\z@\vbox\bgroup
493 \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
494 \default@tag % disallow use of \tag here
495 \ialign\bgroup
496 \hfil
497 \strut@
498 $\m@th\displaystyle {##}%
499 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }%
500 $%
501 \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
502 &$\m@th\displaystyle { }##}%
503 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }%
504 $%
505 \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
506 \hfill % Why not \hfil?---dmj, 1994/12/28
507 \crcr
508 }

```

(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`

```

509 \def\lendsplit@{%
510 \global\setbox9\vtop{\unvcopy\z@}%
511 \ifinalign@
512 \setbox\@ne\vbox{%
513 \unvcopy\z@
514 \global\setbox8\lastbox
515 }%
516 \setbox\@ne\hbox{%
517 \unhcopy8%
518 \unskip
519 \setbox\tw@\lastbox
520 \unskip
521 \global\setbox\thr@@\lastbox
522 }%

```

```

523     \_math_split_debug_typeout:n{lendsplit@/aligncase}
524     \ifctagsplit@
525     \_math_split_debug_typeout:n{lendsplit@/aligncase/centertags}
526     \gdef\split@{%
527         \hbox to\wd\thr@@{%
528             &\vcenter{\vbox{\moveleft\wd\thr@@\box9}}}%
529             \_math_split_debug_typeout:n{finalize~innertable~aligncase}
530             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
531         }%
532     \else
533     %TODO, not correct yet.
534     \_math_split_debug_typeout:n{lendsplit@/aligncase/tbtags}
535     \gdef\split@{%
536         \hbox to\wd\thr@@{%
537             &\vbox{\moveleft\wd\thr@@\box9}}%
538             \_math_split_debug_typeout:n{finalize~innertable~aligncase}
539             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
540         }%
541     \fi
542 \else

```

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

```

543     \ifctagsplit@
544     \bool_if:NTF \l__math_gathersplit_bool
545     {
546         \_math_split_debug_typeout:n{lendsplit/equationcase/centertags}
547         \gdef\split@%
548             {\UseTaggingSocket{math/luamml/annotate/false}}{\vcenter{\box9}}
549     }
550     {
551         \_math_split_debug_typeout:n {lendsplit/gathercase/centertags}
552         \gdef\split@{\vcenter{\box9}}%
553         \_math_split_debug_typeout:n {finalize~innertable~gathercase}
554         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
555     }
556 \else
557 % TODO tbtags not correct yet.
558 \bool_if:NTF \l__math_gathersplit_bool
559 {
560     \_math_split_debug_typeout:n {lendsplit/equationcase/tbtags}
561     \gdef\split@%
562         {\UseTaggingSocket{math/luamml/annotate/false}}{\box9}
563 }
564 {
565     \_math_split_debug_typeout:n {lendsplit/gathercase/tbtags}
566     \gdef\split@{
567         \box9%
568         \_math_split_debug_typeout:n {finalize~innertable~gathercase}
569         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
570 }
571 \fi
572 \fi
573 \aftergroup\split@
574 }

```

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

```

575 \def\rendsplit@{%
576   \ifalign@
577     \global\setbox9 \vtop{%
578       \unvcopy\z@
579       \global\setbox8 \lastbox
580       \unskip
581     }%
582   \setbox\@ne\hbox{%
583     \unhcopy8
584     \unskip
585     \global\setbox\tw@\lastbox
586     \unskip
587     \global\setbox\thr@@\lastbox
588   }%
589   \ifctagsplit@
590     \gdef\split@{%
591       \hbox to\wd\thr@@{%
592         &\vcenter{\vbox{\moveleft\wd\thr@@\boxz}}}%
593       \__math_split_debug_typeout:n {rendsplit/aligncase/centertags}
594       \__math_split_debug_typeout:n {finalize~innertable~aligncase}
595       \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
596     }%
597   \else
598     \__math_split_debug_typeout:n{rendsplit@/aligncase/tbtags}
599     %TODO tbtags is not correct yet
600     \global\setbox7 \hbox{\unhbox\tw@\unskip}%
601     \gdef\split@{%
602       \global\@tempcnta\column@
603       &\setboxz@h{}%
604       \savetaglength@
605       \global\advance\row@\@ne
606       \vbox{\moveleft\wd\thr@@\box9}%
607       \crcr
608       \noalign{\global\lineht@\z@}%
609       \add@amps\@tempcnta
610       \UseTaggingSocket{math/luamml/annotate/false}{\box\thr@@}
611       &\box7
612       \__math_split_debug_typeout:n {finalize~innertable~aligncase}
613       \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
614     }%
615   \fi

```

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

```

616   \else
617     \ifctagsplit@
618     \bool_if:NTF \l__math_gathersplit_bool
619     {
620       \__math_split_debug_typeout:n {rendsplit/equationcase/centertags}
621       \gdef\split@%
622       {\UseTaggingSocket{math/luamml/annotate/false}{\vcenter{\boxz}}}
623     }
624     {
625       \__math_split_debug_typeout:n {rendsplit/gathercase/centertags}

```



```

626         \gdef\split@{\vcenter{\boxz@}%
627         \_math_split_debug_typeof:n {finalize-innertable-gathercase}
628         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
629     }
630 \else
631     \bool_if:NTF \l__math_gathersplit_bool
632     {
633         \_math_split_debug_typeof:n {rendsplit/equationcase/tbtags}
634         \gdef\split@{%
635             \UseTaggingSocket{math/luamml/annotate/false}{\boxz@}}
636     }
637     {
638         \_math_split_debug_typeof:n {rendsplit/gathercase/tbtags}
639         \gdef\split@{%
640             \boxz@
641             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
642     }
643 \fi
644 \fi
645 \aftergroup\split@
646 }

```

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

2.7 `\intertext`

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

`\intertext@` TODO: review and add sockets!

```

647 \def\intertext@{%
648   \def\intertext##1{%
649     \ifvmode\else\\\@empty\fi
650     \noalign{%

```

we have to flip the sign and use a negative `\belowdisplayskip` as we flipped the sign at the outside.

```

651     \penalty\postdisplaypenalty\vskip-\belowdisplayskip
652     \vbox{

```

Stop tagging when measuring:

```

653     \ifmeasuring@\tag_suspend:n{\measuring}\fi
654     \normalbaselines
655     \ifdim\linewidth=\columnwidth
656     \else \parshape\@ne \@totalleftmargin \linewidth
657     \fi

```

End the previous mc:

```

658     \tag_mc_end_push:

```

We are already in a par so we change now to text:

```

659     \tagpdfsetup{para/tag=P}%

```

TODO why `\tagpdfpara0n` needed?

```

660     \tagpdfpara0n
661     \noindent\ignorespaces##1\par

```

Restart the MC

```
662     \tag_mc_begin_pop:n{}}%
663     \penalty\predisdisplaypenalty\vskip\abovedisplayskip%
664   }%
665 }}
```

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

2.8 `\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 possible to redefine `\text` not to use `\mathchoice`

`\text@`

```
666 \AddToHook{package/amstext/after}
667 {
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.
668   \tag_if_active:T
669   {
670     \sys_if_engine_luatex:TF
671     {
672       \def\text@#1{
673         \tag_socket_use:nnn {math/luamml/hbox}{-}
674         {%-
675         \ifcase\mathstyle
676         \hbox{#{#1}}\or
677         \hbox{#{#1}}\or
678         \hbox{#{#1}}\or
679         \hbox{#{#1}}\or
680         \hbox{{\let\f@size\sf@size\selectfont#1}}\or
681         \hbox{{\let\f@size\sf@size\selectfont#1}}\or
682         \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
683         \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
684         \ERROR
685         \fi
686         \check@mathfonts
687       }}
688     }
689     {
690       \def\text@#1
691       {{
```

```

692         \int_gincr:N\g__math_mathchoice_int
693         \tag_suspend:n{\text@}
694         \mathchoice
695         {
696           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{0}
697           \textdef@{\displaystyle}\f@size{#1}
698         }
699         {
700           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{2}
701           \textdef@{\textstyle}\f@size{\firstchoice@false #1}
702         }
703         {
704           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{4}
705           \textdef@{\textstyle}\sf@size{\firstchoice@false #1}
706         }
707         {
708           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{6}
709           \textdef@{\textstyle} \ssf@size{\firstchoice@false #1}
710         }
711         \check@mathfonts
712     }}
713 }
714 }
715 }

```

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

2.9 \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.

```

\pmb@@
\pmb@
716 \AddToHook{package/amsbsy/after}
717 {
718   \def\pmb@@#1#2#3{\leavevmode\hbox_set:Nn\l_@@_tmpa_box{xxx#3}
719     \dimen@-\box_wd:N\l_@@_tmpa_box
720     \kern-.5\ex@\box_use:N\l_@@_tmpa_box
721     \tag_mc_end:\tag_mc_begin:n{artifact}
722     \tag_mc_reset_box:N\l_@@_tmpa_box
723     \kern\dimen@\kern.25\ex@\raise.4\ex@\box_use:N\l_@@_tmpa_box
724     \kern\dimen@\kern.25\ex@\box_use_drop:N\l_@@_tmpa_box
725     \tag_mc_end:\tag_mc_begin:n{}}
726 }
727 \def\pmb@@#1#2{\hbox_set:Nn\l_@@_tmpa_box{\m@th#1{#2}}
728   \setboxz@h{\m@th#1\mkern.5mu}\pmbraise@\wdz@
729   \binrel@{#2}
730   \dimen@-\box_wd:N\l_@@_tmpa_box
731   \binrel@@{
732     \mkern-.8mu\box_use:N\l_@@_tmpa_box
733     \tag_mc_end:\tag_mc_begin:n{artifact}
734     \tag_mc_reset_box:N\l_@@_tmpa_box
735     \kern\dimen@\mkern.4mu\raise\pmbraise@\box_use:N\l_@@_tmpa_box

```

```

736     \kern\dimen@\mkern.4mu\box_use_drop:N\l_@@_tmpa_box
737     \tag_mc_end:\tag_mc_begin:n{
738     }
739   }
740 }

```

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

```

741 \ExplSyntaxOff
742 </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	C
<code>\</code> , 404, 442	<code>\chardef</code> 153, 217, 255, 287, 493
<code>@@</code> commands:	<code>\columnwidth</code> 655
<code>\l_@@_tmpa_box</code> . 718, 719, 720, 722, 723, 724, 727, 730, 732, 734, 735, 736	<code>\cr</code> 148
<code>\</code> 649	<code>\crr</code> 176, 246, 263, 270, 309, 435, 437, 459, 507, 607
A	cs commands:
<code>\abovedisplayskip</code> 663	<code>\cs_new:Npn</code> 444
<code>\AddToHook</code> 666, 716	
<code>\advance</code> 137, 286, 296, 333, 605	D
<code>\aftergroup</code> 573, 645	<code>\DeclareRobustCommand</code> 21
<code>\array</code> 401	<code>\def</code> 7, 8, 20, 23, 91, 134, 150, 179, 268, 284, 312, 315, 354, 360, 389, 397, 400, 465, 481, 491, 509, 575, 647, 648, 672, 690, 718, 727
<code>\arraystretch</code> 400	<code>\displaystyle</code> . . 98, 102, 117, 121, 164, 165, 227, 229, 236, 241, 258, 259, 306, 356, 362, 498, 499, 502, 503, 697
B	<code>\displaywidth</code> 160, 294
<code>\baselineskip</code> 409	<code>\dp</code> 372
<code>\begin</code> 203, 251, 480	E
<code>\begingroup</code> 319	<code>\edef</code> 470
<code>\belowdisplayskip</code> 17, 651	<code>\egroup</code> 26, 29, 183, 195, 272, 274, 387, 394, 439, 440, 461, 462, 486, 487
<code>\bgroup</code> 160, 207, 221, 254, 257, 298, 405, 412, 478, 492, 495	<code>\else</code> 12, 32, 101, 108, 120, 127, 146, 175, 202, 213, 250, 279, 293, 302, 313, 331, 340, 343, 346, 350, 376, 381, 451, 452, 454, 463, 474, 532, 542, 556, 597, 616, 630, 649, 656
bool commands:	<code>\end</code> 480
<code>\bool_if:NTF</code> 544, 558, 618, 631	<code>\endaligned</code> 266, <u>268</u>
<code>\bool_new:N</code> 445	<code>\endgather</code> 6, <u>179</u>
<code>\bool_set_true:N</code> 476	<code>\endgroup</code> 339, 467
bool internal commands:	<code>\endmultiline</code> 8
<code>\l_math_gathersplit_bool</code> <u>445</u> , 476, 544, 558, 618, 631	<code>\endsplit</code> 455
<code>\box</code> 337, 373, 528, 537, 548, 552, 562, 567, 606, 610, 611	
box commands:	
<code>\box_use:N</code> 720, 723, 732, 735	
<code>\box_use_drop:N</code> 724, 736	
<code>\box_wd:N</code> 719, 730	

environments:	int commands:
gather* 187	\int_gincr:N 692
gathered 249	\int_use:N 696 , 700 , 704 , 708
multline* 277	int internal commands:
smallmatrix 403	\g_math_mathchoice_int
split 446 692 , 696 , 700 , 704 , 708
\eqref 2 , 20	\intertext 17 , 648
\ERROR 684	
\ExplSyntaxOff 741	K
\ExplSyntaxOn 6	\kern 140 , 720 , 723 , 724 , 735 , 736
F	L
\fi 18 , 31 , 34 , 103 , 110 , 122 , 129 ,	\label 18 , 152 , 292
135 , 136 , 144 , 146 , 175 , 204 , 216 ,	\lastbox 514 , 519 , 521 , 579 , 585 , 587
252 , 279 , 293 , 297 , 304 , 316 , 336 ,	\lbrace 399
338 , 342 , 345 , 348 , 352 , 379 , 383 ,	\leavevmode 718
452 , 453 , 456 , 463 , 475 , 541 , 571 ,	\left 399
572 , 615 , 643 , 644 , 649 , 653 , 657 , 685	\let 7 , 22 , 151 ,
	152 , 155 , 211 , 214 , 289 , 290 , 292 ,
G	365 , 398 , 455 , 474 , 680 , 681 , 682 , 683
gather* (env.) 187	\lineskip 410 , 411
gathered (env.) 249	\lineskiplimit 411
\gathered 16	\linewidth 655 , 656
\gdef 471 , 526 , 535 , 547 , 552 ,	\llap 368
561 , 566 , 590 , 601 , 621 , 626 , 634 , 639	luamml commands:
\global 136 ,	\luamml_ignore: 7 , 100 , 119 , 401
137 , 146 , 147 , 157 , 159 , 220 , 286 ,	
290 , 452 , 472 , 492 , 510 , 514 , 521 ,	M
577 , 579 , 585 , 587 , 600 , 602 , 605 , 608	math commands:
	\math_register_halign_env:nn
H 82 , 83 , 84 ,
\halign 160 , 298	85 , 86 , 87 , 88 , 89 , 90 , 198 , 199 , 282 , 283
\hbox 11 , 14 , 20 , 299 , 323 , 332 ,	math internal commands:
374 , 516 , 527 , 536 , 582 , 591 , 600 ,	_math_split_debug_typeof:n
676 , 677 , 678 , 679 , 680 , 681 , 682 , 683 444 , 444 , 447 , 458 , 466 , 482 ,
hbox commands:	483 , 491 , 523 , 525 , 529 , 534 , 538 ,
\hbox_set:Nn 718 , 727	546 , 551 , 553 , 560 , 565 , 568 , 593 ,
\hfil 92 , 131 , 223 , 244 , 258 , 262 ,	594 , 598 , 612 , 620 , 625 , 627 , 633 , 638
303 , 307 , 413 , 422 , 425 , 434 , 496 , 506	_math_tag_if_mathstyle:nn
\hfill 506 696 , 700 , 704 , 708
\hfilneg 314 , 384 , 390	\mathchoice 18 , 694
\hskip 301 ,	\mathstyle 675
311 , 312 , 315 , 337 , 367 , 377 , 382 , 391	\measuring 7 , 653
	\minalignsep 215
I	\mkern 728 , 732 , 735 , 736
\ialign 221 , 257 , 412 , 495	\moveleft 528 , 537 , 592 , 606
\ifcase 675	\multiply 210
\ifdim 334 , 655	multline* (env.) 277
\iffalse 31	\multlinegap 315 , 367 , 382 , 391
\ifinner 449	\multlinetaggap 334 , 335 , 377
\ifnum 31 , 209	
\iftrue 7	N
\ifvmode 649	\noalign 608 , 650
\ignorespaces 661	\noindent 661
\ignorespacesafterend 35 , 185 , 196 , 280	\nonumber 30 , 135 , 472

<code>\normalbaselines</code>	322, 370, 654	<code>\tag_suspend:n</code>	7, 653, 693
<code>\normalfont</code>	11, 14, 20	<code>\tagpdfparaOn</code>	17, 660
<code>\null</code>	371, 404	<code>\tagpdfsetup</code>	659
O			
<code>\omit</code>	139	TeX and L ^A T _E X 2 _ε commands:	
<code>\or</code> ..	676, 677, 678, 679, 680, 681, 682, 683	<code>\@centering</code>	171, 301
P			
<code>\par</code>	661	<code>\@currenvir</code>	203
<code>\parshape</code>	656	<code>\@display@init</code>	286
<code>\pdfannot</code>	18	<code>\@empty</code>	290, 455, 474, 649
<code>\penalty</code>	651, 663	<code>\@eqnswtrue</code>	146, 452
<code>\pmb</code>	19	<code>\@gobble</code>	292
<code>\postdisplaypenalty</code>	651	<code>\@ifnextchar</code>	398
<code>\predisplaypenalty</code>	663	<code>\@lign</code>	96, 115, 142
<code>\ProvidesFile</code>	3	<code>\@mathmargin</code> ..	293, 296, 311, 312, 333
Q			
<code>\quad</code>	401	<code>\@ne</code>	38, 43, 68, 73, 137, 159, 217, 255, 286, 371, 372, 373, 493, 512, 516, 582, 605, 656
R			
<code>\raise</code>	723, 735	<code>\@nx</code>	471, 472
<code>\ref</code>	22	<code>\@tempcnta</code>	602, 609
<code>\relax</code>	208, 301, 337, 365	<code>\@totalleftmargin</code>	656
<code>\renewcommand</code>	200	<code>\@xp</code>	279, 450, 455, 463, 467, 469
<code>\renewenvironment</code> ..	37, 42, 47, 52, 57, 62, 67, 72, 77, 187, 249, 277, 403, 446	<code>\add@amps</code>	138, 609
<code>\RequirePackage</code>	5	<code>\align@</code>	4
<code>\rlap</code>	321	<code>\align@preamble</code>	91
S			
<code>\scriptsize</code>	418, 430	<code>\alignedspace@left</code>	206, 253
<code>\scriptstyle</code>	416, 428	<code>\alignsep@</code>	132, 140, 212, 215, 245
<code>\selectfont</code>	680, 681, 682, 683	<code>\ams@return@opt@arg</code>	247, 264
<code>\setbox</code> 332, 371, 492, 510, 512, 514, 516, 519, 521, 577, 579, 582, 585, 587, 600		<code>\ams@start@box</code>	207, 254
<code>smallmatrix</code> (env.)	403	<code>\binrel@</code>	729
<code>\span</code>	175	<code>\binrel@@</code>	731
<code>split</code> (env.)	446	<code>\black@</code>	24, 181, 193, 485
<code>\split</code>	151	<code>\boxz@</code>	592, 622, 626, 635, 640
sys commands:		<code>\calc@shift@gather</code>	169
<code>\sys_if_engine luatex:TF</code>	670	<code>\check@mathfonts</code>	686, 711
T			
<code>\tabskip</code> 112, 132, 158, 171, 175, 232, 245, 293		<code>\collect@body</code>	455, 457
<code>\tag</code>	152, 289, 292, 494	<code>\column@</code>	220, 602
tag commands:		<code>\column@plus</code>	222, 233
<code>\tag_if_active:TF</code>	668	<code>\common@align@ending</code>	
<code>\tag_mc_begin:n</code> ..	721, 725, 733, 737	. 23, 40, 45, 50, 55, 60, 65, 70, 75, 80	
<code>\tag_mc_begin_pop:n</code>	662	<code>\default@tag</code>	218, 408, 494
<code>\tag_mc_end:</code>	721, 725, 733, 737	<code>\df@tag</code>	469, 471
<code>\tag_mc_end_push:</code>	658	<code>\dimen@</code>	333, 334, 335, 337, 719, 723, 724, 730, 735, 736
<code>\tag_mc_reset_box:N</code>	722, 734	<code>\displ@y@</code>	154
<code>\tag_socket_use:nmn</code>	673	<code>\dspbrk@context</code> 153, 217, 255, 287, 493	
		<code>\dspbrk@lvl</code>	286
		<code>\endmathdisplay@a</code>	481
		<code>\endmultline@math</code> ..	10, 306, 354, 365
		<code>\env@cases</code>	397
		<code>env@cases</code>	397
		<code>\ex@</code>	409, 410, 720, 723, 724
		<code>\f@size</code> ..	680, 681, 682, 683, 697, 701
		<code>\firstchoice@false</code>	701, 705, 709
		<code>\gather@</code>	150, 479
		<code>\gather@split</code>	12–15, 457, 465

<code>\unhcopy</code>	517, 583	421, 431, 433, 441, 442, 499, 501,
<code>\unskip</code>	518, 520, 580, 584, 586, 600	503, 505, 530, 539, 548, 554, 562,
<code>\unvcopy</code>	510, 513, 578	569, 595, 610, 613, 622, 628, 635, 641
use commands:			
<code>\use_none:n</code>	444	
<code>\UseExpandableTaggingSocket</code>	25,	
		182, 194, 271, 386, 393, 438, 460, 488	
<code>\UseTaggingSocket</code>	13,	
		15, 17, 102, 109, 121, 128, 145,	
		165, 168, 174, 206, 229, 231, 241,	
		243, 253, 259, 261, 275, 347, 351,	
		356, 358, 362, 364, 380, 404, 419,	
V			
<code>\vbox</code>	321,	
		327, 492, 512, 528, 537, 592, 606, 652	
<code>\vcenter</code>		405, 478, 528, 548, 552, 592, 622, 626	
<code>\vskip</code>	651, 663	
<code>\vtop</code>	368, 510, 577	
W			
<code>\wd</code>	..	333, 527, 528, 536, 537, 591, 592, 606	