

The latex-lab-amsmath code*

L^AT_EX Project

September 15, 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

```
3 \ProvidesFile{latex-lab-amsmath.ltx}
4 [2026-04-25 v0.1j amsmath adaptions]

5 \ExplSyntaxOn
```

2.2 Measuring

When measuring we neither want tagging nor the luamml processing.

```
\measuring@true

6 \AddToHook{begindocument}
7 {
8   \def\measuring@true{\let\ifmeasuring@\iftrue\tag_suspend:n{\measuring}\luamml_ignore:}
9 }
```

(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@@@

10 \def\maketag@@@#1
11 {%
12   \ifmeasuring@
13     \hbox{\m@th\normalfont#1}%
14   \else
15     \UseTaggingSocket{math/display/tag/begin}
16     \hbox{\m@th\normalfont#1
17     \UseTaggingSocket{math/luamml/mtable/tag/save}
18     }%
19     \UseTaggingSocket{math/display/tag/end}
20   \fi
21 }
```

(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
22 \def\maketag@@@notag#1{\hbox{\m@th\normalfont#1}}
23 \DeclareRobustCommand{\eqref}[1]
24 {\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`

```

25 \def\common@align@ending {
26   \math@cr \black@\totwidth@
27   \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {\@currenvir}
28   \egroup
29   \ifingather@
30     \restorealignstate@
31     \egroup
32     \nonumber
33     \ifnum0=`{\fi\iffalse}\fi
34   \else
35     \dollar\dollar@end
36   \fi
37   \ignorespacesafterend
38 }
```

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

Now we redefine the display alignments to use these ending.

```

39 \renewenvironment{align}{%
40   \start@align\@ne\st@rredfalse\m@ne
41 }{%
42   \common@align@ending
43 }
44 \renewenvironment{align*}{%
45   \start@align\@ne\st@rredtrue\m@ne
46 }{%
47   \common@align@ending
48 }
49 \renewenvironment{alignat}{%
50   \start@align\z@\st@rredfalse
51 }{%
52   \common@align@ending
53 }
54 \renewenvironment{alignat*}{%
55   \start@align\z@\st@rredtrue
56 }{%
57   \common@align@ending
58 }
59 \renewenvironment{flalign}{%
60   \start@align\tw@\st@rredfalse\m@ne
61 }{%
62   \common@align@ending
63 }
64 \renewenvironment{flalign*}{%
65   \start@align\tw@\st@rredtrue\m@ne
66 }{%
67   \common@align@ending
68 }
69 \renewenvironment{xalignat}{%

```

```

70 \start@align\@ne\st@rredfalse
71 }{%
72 \common@align@ending
73 }
74 \renewenvironment{xalignat*}{%
75 \start@align\@ne\st@rredtrue
76 }{%
77 \common@align@ending
78 }
79 \renewenvironment{xxalignat}{%
80 \start@align\tw@\st@rredtrue
81 }{%
82 \common@align@ending
83 }

```

And register these environments for the math grabbing.

```

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

```

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

\align@preamble

```

93 \def\align@preamble{%
94 &\hfil
95 \strut@
96 \setboxz@h
97 {
98 \@lign
99 $
100 \m@th\displaystyle{##}
101 \ifmeasuring@
102 \luamml_ignore:
103 \else
104 \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
105 \fi
106 $
107 }%
108 \ifmeasuring@
109 \savefieldlength@
110 \else
111 \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
112 \fi
113 \set@field
114 \tabskip\z@skip
115 &\setboxz@h
116 {
117 \@lign

```

```

118      $
119      \m@th\displaystyle{##}
120      \ifmeasuring@
121      \luamml_ignore:
122      \else
123      \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }
124      \fi
125      $
126      }%
127      \ifmeasuring@
128      \savefieldlength@
129      \else
130      \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
131      \fi
132      \set@field
133      \hfil
134      \tabskip\alignsep@
135  }

```

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

\math@cr@@@align

```

136 \def\math@cr@@@align{%
137   \ifst@rred\nonumber\fi
138   \if@eqnsw \global\tag@true \fi
139   \global\advance\row@\@ne
140   \add@amps\maxfields@
141   \omit
142   \kern-\alignsep@
143   \iftag@
144     \setboxz@h{\@lign\strut@{\make@display@tag}}%
145     \place@tag
146   \fi
147   \UseTaggingSocket{math/luamml/mtable/tag/set}
148   \ifst@rred\else\global\@eqnswtrue\fi
149   \global\lineht@\z@
150   \cr
151 }

```

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

2.3.3 gather and gather*

\gather@

```

152 \def\gather@#1{%
153   \ingather@true \let\split\insplit@
154   \let\tag\tag@in@align \let\label\label@in@display
155   \chardef\dspbrk@context\z@
156   \intertext@ \disply@ \Let@
157   \let\math@cr@@@math@cr@@@gather
158   \gmeasure@{#1}%
159   \global\shifftag@false
160   \tabskip\z@skip
161   \global\row@\@ne
162   \halign to\displaywidth\bgroup

```

```

163     \strut@
164     \setboxz@h
165     {
166         $\m@th\displaystyle{##}
167         \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
168         $
169     }%
170     \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
171     \calc@shift@gather
172     \set@gather@field
173     \tabskip\@centering
174     &\setboxz@h{\strut@{##}}%
175     \place@tag@gather
176     \UseTaggingSocket{math/luamml/mtable/tag/set}
177     \tabskip \iftagsleft@ \gdisplaywidth@ \else \z@skip \span\fi
178     \crr
179     #1%
180 }

```

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

`\endgather`

```

181 \def\endgather{
182     \math@cr
183     \black@ \totwidth@
184     \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
185     \egroup
186     \dollar@end
187     \ignorespacesafterend
188 }

```

(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.)`

```

189 \renewenvironment{gather*}
190 {
191     \start@gather\st@rredtrue
192 }
193 {
194     \math@cr
195     \black@\totwidth@
196     \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
197     \egroup
198     \dollar@end
199     \ignorespacesafterend
200 }

```

Register both environments

```

201 \math_register_halign_env:n {gather}{}
202 \math_register_halign_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`

```

203 \renewcommand{\start@aligned}[2]{
204   \RifM@
205   \else
206     \nonmatherr@ {\begin{\@currenvir}}
207   \fi
208   \savecolumn@ % Assumption: called inside a group
209   \UseTaggingSocket{math/luamml/annotate/false}{ }{ \alignedspace@left }
210   \ams@start@box {#1} \bgroup
211     \maxfields@ #2 \relax
212     \ifnum \maxfields@ > \m@ne
213       \multiply \maxfields@ \tw@
214       \let \math@cr@@@ \math@cr@@@alignedat
215       \alignsep@ \z@skip
216     \else
217       \let \math@cr@@@ \math@cr@@@aligned
218       \alignsep@ \minalignsep
219     \fi
220     \Let@ \chardef \dspbrk@context \@ne
221     \default@tag
222     \spread@equation % no-op if already called
223     \global \column@ \z@
224     \ialign \bgroup
225       & \column@plus
226       \hfil
227       \strut@
228       $
229         \m@th
230         \displaystyle
231         {##}
232         \UseTaggingSocket{math/luamml/save/nNn}{ { } \displaystyle {mtd} }
233       $
234       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
235       \tabskip \z@skip
236     & \column@plus
237     $
238       \m@th
239       \displaystyle
240       {
241         {}
242         ##
243       }
244       \UseTaggingSocket{math/luamml/save/nNn}{ { } \displaystyle {mtd} }
245     $
246     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
247     \hfil
248     \tabskip\alignsep@
249   \crrc
250   \ams@return@opt@arg

```

251 }

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

`gathered (env.)`

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

This annotates the space

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

This save the cell and then finalize it.

```
262       \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
263       $
264       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
265       \hfil
266       \crr
267   \ams@return@opt@arg
268 }{%
269   \endaligned
270 }
```

`\endaligned`

```
271 \def\endaligned
272 {
273   \relax
274   \ifnum\column@>\z@
275     \math@cr@@@
276   \else
277     \crr
278   \fi
279   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}
280   \egroup
281   \restorecolumn@
282   \egroup
283   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
284 }
```

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

2.3.5 multiline and multiline*

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

`multline* (env.)`

```

285 \renewenvironment{multline*}{\start@multline\st@rredtrue}
286 {%
287 \iftagsleft@ \@xp\lendmultline@ \else \@xp\rendmultline@ \fi
288 \ignorespacesafterend
289 }

```

And now we register both

```

290 \math_register_halign_env:nn {multline}{}
291 \math_register_halign_env:nn {multline*}{}

```

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

`\multline@`

```

292 \def\multline@#1{%
293   \Let@
294   \@display@init{\global\advance\row@ \@ne \global\dspbrk@lvl\m@ne}%
295   \chardef\dspbrk@context\z@
296   \restore@math@cr
297   \let\tag\tag@in@align
298   \global\tag@false \global\let\raise@tag\empty
299   \mmeasure@{#1}%
300   \let\tag\gobble@tag \let\label\@gobble
301   \tabskip \if@fleqn \@mathmargin \else \z@skip \fi
302   \totwidth@\displaywidth
303   \if@fleqn
304     \advance\totwidth@-\@mathmargin
305   \fi
306   \halign\bgroup
307     \hbox to\totwidth@{%
308       \if@fleqn
309         \hskip \@centering \relax
310       \else
311         \hfil
312       \fi
313       \strut@
314       $\m@th\displaystyle{##}\endmultline@math
315       \hfil
316     }%
317   \crcr
318   \if@fleqn
319     \hskip-\@mathmargin
320     \def\multline@indent{\hskip\@mathmargin}% put it back
321   \else
322     \hfilneg
323     \def\multline@indent{\hskip\multlinegap}%
324   \fi
325   \iftagsleft@
326     \iftag@
327       \begingroup
328         \ifshifttag@
329           \rlap{\vbox{%
330             \normalbaselines
331             \hbox{%

```

```

332             \strut@
333             \make@display@tag
334         }%
335         \vbox to\lineht@{}%
336         \raise@tag
337     }}%
338     \multline@indent
339 \else
340     \setbox\z@\hbox{\make@display@tag}%
341     \dimen@\@mathmargin \advance\dimen@-\wd\z@
342     \ifdim\dimen@<\multlinetaggap
343         \dimen@\multlinetaggap
344     \fi
345     \box\z@ \hskip\dimen@\relax
346 \fi
347 \endgroup
348 \else
349     \multline@indent
350 \fi
351 \else
352     \multline@indent
353 \fi
354 \ifmeasuring@ \else
355     \UseTaggingSocket{math/luamml/mtable/aligncol} {left}
356 \fi
357 #1%
358 \ifmeasuring@ \else
359     \UseTaggingSocket{math/luamml/mtable/aligncol} {right}
360 \fi
361 }

```

(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`

```

362 \def\endmultline@math
363 {
364     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
365     $
366     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
367 }

```

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

`\rendmultline@`

```

368 \def\rendmultline@{%
369     \iftag@

```

we need to surround the `math` token with tagging sockets.

```

370     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
371     $
372     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}

```

```

373         \let\endmultline@math\relax
374         \ifshifttag@
375             \hskip\multlinegap
376             \llap{\vtop{%
377                 \raise@tag
378                 \normalbaselines
379                 \setbox\@ne\hbox{\hfil\strut}%
380                 \dp\@ne\lineht@
381                 \box\@ne
382                 \hbox{\strut@make@display@tag}%
383             }}%
384         \else
385             \hskip\multlinetaggap
386             \make@display@tag
387         \fi

```

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

```

388     \UseTaggingSocket{math/luamml/mtable/tag/set}
389     \else
390         \hskip\multlinegap
391     \fi
392     \hfilneg
393     \math@cr

```

Now we finalize the mtable.

```

394     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
395     \egroup
396     \dollar\dollar@end
397 }

```

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

And something similar for the left version

\lendmultline@

```

398 \def\lendmultline@{%
399     \hfilneg
400     \hskip\multlinegap
401     \math@cr
402     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
403     \egroup
404     \dollar\dollar@end
405 }

```

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

2.4 Cases

env@cases

```

406 \def\env@cases{%
407     \let\@ifnextchar\new@ifnextchar
408     \left\lbracket
409     \def\arraystretch{1.2}%
410     \array{@{}l@{\quad}l@{\luamml_ignore:}}%
411 }

```

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

2.4.1 smallmatrix

`smallmatrix (env.)`

```

412 \renewenvironment {smallmatrix} {
413   \UseTaggingSocket{ math/luamml/annotate/false } {} { \null\, }
414   \vcenter \bgroup
415   \Let@
416   \restore@math@cr
417   \default@tag
418   \baselineskip 6 \ex@
419   \lineskip 1.5 \ex@
420   \lineskiplimit \lineskip
421   \ialign \bgroup
422     \hfil
423     $
424     \m@th
425     \scriptstyle
426     ##
427     % No \scriptsize here since we want to add the mstyle nodes
428     \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
429     $
430     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
431     \hfil
432     &&
433     \thickspace
434     \hfil
435     $
436     \m@th
437     \scriptstyle
438     ##
439     % No \scriptsize here since we want to add the mstyle nodes
440     \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
441     $
442     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
443     \hfil
444   \crcr
445 }{%
446   \crcr
447   \UseExpandableTaggingSocket{math/luamml/mtable/smallmatrix/save}
448 \egroup
449 \egroup
450 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
451 \UseTaggingSocket{ math/luamml/annotate/false } {} { \, , }
452 }
```

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`

```
453 \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`

```
454 \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.)`

```
455 \renewenvironment{split}{%
456   \_math_split_debug_typeout:n {begin-split}
457   \if@display
458     \ifinner
459       \@xp\@xp\@xp\split@aligned
460     \else
461       \ifst@rred \else \global\@eqnswtrue \fi
462     \fi
463   \else
464     \let\endsplit\@empty \@xp\collect@body\@xp\split@err
465   \fi
466   \collect@body\gather@split
467 }{\_math_split_debug_typeout:n{end-split}}%
468   \crrc
469   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}%
470   \egroup
471 \egroup
472 \iftagsleft@ \@xp\lendsplit@ \else \@xp\rendsplit@ \fi
473 }
```

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`

```
474 \def\gather@split#1#2#3{
475   \_math_split_debug_typeout:n{gather@split}%
476   \@xp\endgroup \reset@equation % math@cr will handle equation numbering
477   \iftag@
478     \toks@\@xp{\df@tag}%
479     \edef\split@tag{%
480       \gdef\@nx\df@tag{\the\toks@}%
481       \global\@nx\tag@true \@nx\nonumber
482     }%
483   \else \let\split@tag\@empty
484   \fi
485   \bool_set_true:N\l_math_gathersplit_bool
```

```

486 \spread@equation
487 \vcenter\bgroup
488   \gather@{\split@tag
489   \begin{split}#1\end{split}}%
490   \def\endmathdisplay@a{%
491     \_math_split_debug_typeout:n{endmathdisplay@a}
492     \_math_split_debug_typeout:n{finalize~innertable~endmathdisplay@a}
493     \math@cr
494     \black@ \totwidth@
495     \egroup
496     \egroup
497     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/finalize}%
498   }%
499 }

```

(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.

```

500 \def\insplit@{\_math_split_debug_typeout:n{insplit@}}%
501 \global\setbox\z@\vbox\bgroup
502   \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
503   \default@tag % disallow use of \tag here
504   \ialign\bgroup
505     \hfil
506     \strut@
507     $\m@th\displaystyle {##}%
508     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
509     $%
510     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
511     &$\m@th\displaystyle { {}##}%
512     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
513     $%
514     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
515     \hfill % Why not \hfil?---dmj, 1994/12/28
516     \crcr
517 }

```

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

And now the difficult part. Depending on the options `leqno/reqno` `\lendsplit@` or `\rendsplitle@` 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`

```

518 \def\lendsplit@{%
519   \global\setbox9\vtop{\unvcopy\z@}%
520   \ifalign@
521     \setbox\@ne\vbox{%
522       \unvcopy\z@
523       \global\setbox8\lastbox
524     }%
525     \setbox\@ne\hbox{%
526       \unhcopy8%
527       \unskip
528       \setbox\tw@\lastbox
529       \unskip
530       \global\setbox\thr@@\lastbox

```

```

531 }%
532 \_math_split_debug_typeout:n{lendsplit@/aligncase}
533 \ifctagsplit@
534 \_math_split_debug_typeout:n{lendsplit@/aligncase/centertags}
535 \gdef\split@{%
536 \hbox to\wd\thr@@{%
537 &\vcenter{\vbox{\moveleft\wd\thr@@\box9}}}%
538 \_math_split_debug_typeout:n{finalize~innertable~aligncase}
539 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
540 }%
541 \else
542 %TODO, not correct yet.
543 \_math_split_debug_typeout:n{lendsplit@/aligncase/tbtags}
544 \gdef\split@{%
545 \hbox to\wd\thr@@{%
546 &\vbox{\moveleft\wd\thr@@\box9}}%
547 \_math_split_debug_typeout:n{finalize~innertable~aligncase}
548 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
549 }%
550 \fi
551 \else

```

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

```

552 \ifctagsplit@
553 \bool_if:NTF \l__math_gathersplit_bool
554 {
555 \_math_split_debug_typeout:n{lendsplit/equationcase/centertags}
556 \gdef\split@%
557 {\UseTaggingSocket{math/luamml/annotate/false}{\vcenter{\box9}}}
558 }
559 {
560 \_math_split_debug_typeout:n {lendsplit/gathercase/centertags}
561 \gdef\split@{\vcenter{\box9}}%
562 \_math_split_debug_typeout:n {finalize~innertable~gathercase}
563 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
564 }
565 \else
566 % TODO tbtags not correct yet.
567 \bool_if:NTF \l__math_gathersplit_bool
568 {
569 \_math_split_debug_typeout:n {lendsplit/equationcase/tbtags}
570 \gdef\split@%
571 {\UseTaggingSocket{math/luamml/annotate/false}{\box9}}
572 }
573 {
574 \_math_split_debug_typeout:n {lendsplit/gathercase/tbtags}
575 \gdef\split@{
576 \box9%
577 \_math_split_debug_typeout:n {finalize~innertable~gathercase}
578 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
579 }
580 \fi
581 \fi
582 \aftergroup\split@

```

583 }

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

```

584 \def\rendsplit@{%
585   \ifalign@
586     \global\setbox9 \vtop{%
587       \unvcopy\z@
588       \global\setbox8 \lastbox
589       \unskip
590     }%
591     \setbox\@ne\hbox{%
592       \unhcopy8
593       \unskip
594       \global\setbox\tw@\lastbox
595       \unskip
596       \global\setbox\thr@\lastbox
597     }%
598     \ifctagsplit@
599       \gdef\split@{%
600         \hbox to\wd\thr@{%
601           &\vcenter{\vbox{\moveleft\wd\thr@\boxz}}}%
602           \_math\_split\_debug\_typeout:n {rendsplit/aligncase/centertags}
603           \_math\_split\_debug\_typeout:n {finalize~innertable~aligncase}
604           \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
605         }%
606       \else
607         \_math\_split\_debug\_typeout:n{rendsplit@/aligncase/tbtags}
608         %TODO tbtags is not correct yet
609         \global\setbox7 \hbox{\unhbox\tw@\unskip}%
610         \gdef\split@{%
611           \global\@tempcnta\column@
612           &\setboxz@h{}%
613           \savetaglength@
614           \global\advance\row@\@ne
615           \vbox{\moveleft\wd\thr@\box9}%
616           \crrc
617           \noalign{\global\lineht@z@}%
618           \add@amps\@tempcnta
619           \UseTaggingSocket{math/luamml/annotate/false}{\box\thr@}
620           &\box7
621           \_math\_split\_debug\_typeout:n {finalize~innertable~aligncase}
622           \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
623         }%
624       \fi

```

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

```

625   \else
626     \ifctagsplit@
627       \bool_if:NTF \l__math_gathersplit_bool
628       {
629         \_math\_split\_debug\_typeout:n {rendsplit/equationcase/centertags}
630         \gdef\split@%
631         {\UseTaggingSocket{math/luamml/annotate/false}{\vcenter{\boxz}}}
632       }

```

```

633     {
634         \_math_split_debug_typeout:n {rendsplit/gathercase/centertags}
635         \gdef\split@{\vcenter{\boxz@}}%
636         \_math_split_debug_typeout:n {finalize~innertable~gathercase}
637         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
638     }
639     \else
640     \bool_if:NTF \l__math_gathersplit_bool
641     {
642         \_math_split_debug_typeout:n {rendsplit/equationcase/tbtags}
643         \gdef\split@{%
644             \UseTaggingSocket{math/luamml/annotate/false}}{\boxz@}}
645     }
646     {
647         \_math_split_debug_typeout:n {rendsplit/gathercase/tbtags}
648         \gdef\split@{%
649             \boxz@
650             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
651     }
652     \fi
653     \fi
654     \aftergroup\split@
655 }

```

(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@

```

656 \NewTaggingSocket {math/intertext}{2}
657 \NewTaggingSocketPlug {math/intertext}{kernel}
658 {
659     \tag_mc_end_push:
660     \tagpdfparaOn
661     #2
662     \tagpdfparaOff
663     \tag_mc_begin_pop:n{ }
664 }
665 \AssignTaggingSocketPlug{math/intertext}{kernel}

```

This can go once luamml defines the plug directly

```

666 \IfSocketPlugExistsF {math/intertext}{luamml}
667 {
668     \int_new:N \l__luamml_struct_tmpa_int
669     \NewTaggingSocketPlug{math/intertext}{luamml}
670     {
671         \tag_mc_end_push:
672         \tag_struct_begin:n{
673             tag = mtext,
674             stash,
675         }

```

```

676     \int_set:Nn \l__luamml_struct_tmpa_int { \tag_get:n{struct_num} }
677     \tag_mc_begin:n{}
678     #2
679     \tag_mc_end:
680     \tag_struct_end:
681     \__luamml_amsmath_save_intertext_with_struct_elem:N \l__luamml_struct_tmpa_int
682     \tag_mc_begin_pop:n{}
683   }
684   \AddToHook{package/luamml/after}{\AssignTaggingSocketPlug{math/intertext}{luamml}}
685 }

686 \def\intertext@{%
687   \def\intertext##1{%
688     \ifvmode\else\\@empty\fi
689     \noalign{%
690       \penalty\postdisplaypenalty\vskip\belowdisplayskip
691       \vbox{

```

Stop tagging when measuring:

```

692     \ifmeasuring@\tag_suspend:n{\measuring}\fi
693     \normalbaselines
694     \ifdim\linewidth=\columnwidth
695       \else \parshape\@ne \@totalleftmargin \linewidth
696     \fi

```

Use the tagging socket

```

697     \UseTaggingSocket{math/intertext}{}
698     {
699       \noindent\ignorespaces##1\par
700     }
701   }% end vbox
702   \penalty\predisplaypenalty\vskip\abovedisplayskip%
703 }%
704 }}

```

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

```

\text@
705 \AddToHook{package/amstext/after}

```

706 {

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.

```

707 \sys_if_engine luatex:TF
708 {
709   \def\text@#1{{
710     \tag_socket_use:nnn {math/luamml/hbox}{}
711     {%
712       \ifcase\mathstyle
713       \hbox{{#1}}\or
714       \hbox{{#1}}\or
715       \hbox{{#1}}\or
716       \hbox{{#1}}\or
717       \hbox{{\let\f@size\sf@size\selectfont#1}}\or
718       \hbox{{\let\f@size\sf@size\selectfont#1}}\or
719       \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
720       \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
721       \ERROR
722     \fi
723     \check@mathfonts
724   }}}
725 }
726 {
727   \tag_if_active:T
728   {
729     \def\text@#1
730     {{
731       \int_gincr:N\g__math_mathchoice_int
732       \tag_suspend:n{\text@}
733       \mathchoice
734       {
735         \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{0}
736         \textdef@\displaystyle\f@size{#1}
737       }
738       {
739         \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{2}
740         \textdef@\textstyle\f@size{\firstchoice@false #1}
741       }
742       {
743         \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{4}
744         \textdef@\textstyle\sf@size{\firstchoice@false #1}
745       }
746       {
747         \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{6}
748         \textdef@\textstyle \ssf@size{\firstchoice@false #1}
749       }
750       \check@mathfonts
751     }}
752   }
753 }
754 }

```

(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.

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

```

755 \cs_new:Npn\__math_patch_amsbsy_pmb:
756 {
757   \NewTaggingSocket{amsbsy/pmb}{2}
758   \NewTaggingSocketPlug{amsbsy/pmb}{default}
759   {
760     \tag_mc_end:\tag_mc_begin:n{artifact}
761     \tag_mc_reset_box:N##1
762     ##2
763     \tag_mc_end:
764     \tag_mc_begin:n{ }
765   }
766   \AssignTaggingSocketPlug{amsbsy/pmb}{default}
767   \def\pmb@@##1##2##3{\leavevmode\hbox_set:Nn\l__math_tmpa_box{##3}
768     \dimen@-\box_wd:N\l__math_tmpa_box
769     \kern-.5\ex@\box_use:N\l__math_tmpa_box
770     \UseTaggingSocket{amsbsy/pmb}{\l__math_tmpa_box}
771     {
772       \kern\dimen@\kern.25\ex@\raise.4\ex@\box_use:N\l__math_tmpa_box
773       \kern\dimen@\kern.25\ex@\box_use_drop:N\l__math_tmpa_box
774     }
775   }
776   \def\pmb@@##1##2{\hbox_set:Nn\l__math_tmpa_box{${\m@th##1}{##2}$}
777     \setboxzz@h{${\m@th##1}\mkern.5mu$}\pmbraise@\wdz@
778     \binrel@{##2}
779     \dimen@-\box_wd:N\l__math_tmpa_box
780     \binrel@@{
781       \mkern-.8mu\box_use:N\l__math_tmpa_box
782       \UseTaggingSocket{amsbsy/pmb}{\l__math_tmpa_box}
783       {
784         \kern\dimen@\mkern.4mu\raise\pmbraise@\box_use:N\l__math_tmpa_box
785         \kern\dimen@\mkern.4mu\box_use_drop:N\l__math_tmpa_box
786       }
787     }
788   }
789 }
790 \AddToHook{package/amsbsy/after}
791 {
792   \__math_patch_amsbsy_pmb:
793 }

```

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

```

794 \ExplSyntaxOff

```

```

795 \</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	
$\backslash$,	413, 451
$\backslash\backslash$	688
A	
$\backslash$ abovedisplayskip	702
$\backslash$ AddToHook	6, 684, 705, 790
$\backslash$ advance	139, 294, 304, 341, 614
$\backslash$ aftergroup	582, 654
$\backslash$ array	410
$\backslash$ arraystretch	409
$\backslash$ AssignTaggingSocketPlug . .	665, 684, 766
B	
$\backslash$ baselineskip	418
$\backslash$ begin	206, 254, 489
$\backslash$ begingroup	327
$\backslash$ belowdisplayskip	690
$\backslash$ bgroup	162, 210, 224, 257, 260, 306, 414, 421, 487, 501, 504
bool commands:	
$\backslash$ bool_if:NTF	553, 567, 627, 640
$\backslash$ bool_new:N	454
$\backslash$ bool_set_true:N	485
bool internal commands:	
$\backslash$ l_math_gathersplit_bool	454, 485, 553, 567, 627, 640
$\backslash$ box	345, 381, 537, 546, 557, 561, 571, 576, 615, 619, 620
box commands:	
$\backslash$ box_use:N	769, 772, 781, 784
$\backslash$ box_use_drop:N	773, 785
$\backslash$ box_wd:N	768, 779
box internal commands:	
$\backslash$ l_math_tmpa_box	767, 768, 769, 770, 772, 773, 776, 779, 781, 782, 784, 785
C	
$\backslash$ chardef	155, 220, 258, 295, 502
$\backslash$ columnwidth	694
$\backslash$ cr	150
$\backslash$ crcr	178, 249, 266, 277, 317, 444, 446, 468, 516, 616
cs commands:	
$\backslash$ cs_new:Npn	453, 755
D	
$\backslash$ DeclareRobustCommand	23
$\backslash$ 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
$\backslash$ displaystyle	100, 104, 119, 123, 166, 167, 230, 232, 239, 244, 261, 262, 314, 364, 370, 507, 508, 511, 512, 736
$\backslash$ displaywidth	162, 302
$\backslash$ dp	380
E	
$\backslash$ edef	479
$\backslash$ egroup	28, 31, 185, 197, 280, 282, 395, 403, 448, 449, 470, 471, 495, 496
$\backslash$ 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
$\backslash$ end	489
$\backslash$ endaligned	269, <u>271</u>
$\backslash$ endgather	6, <u>181</u>
$\backslash$ endgroup	347, 476
$\backslash$ endmultiline	8
$\backslash$ endsplit	464
environments:	
gather*	<u>189</u>
gathered	<u>252</u>
multline*	<u>285</u>
smallmatrix	412
split	<u>455</u>
$\backslash$ eqref	2, <u>22</u>
$\backslash$ ERROR	721
$\backslash$ ExplSyntaxOff	794
$\backslash$ ExplSyntaxOn	5
F	
$\backslash$ 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.)	<u>189</u>
gathered (env.)	<u>252</u>
$\backslash$ gathersplit	<u>16</u>

<code>\gdef</code>	480, 535, 544, 556, 561, 570, 575, 599, 610, 630, 635, 643, 648
<code>\global</code>	138, 139, 148, 149, 159, 161, 223, 294, 298, 461, 481, 501, 519, 523, 530, 586, 588, 594, 596, 609, 611, 614, 617
H	
<code>\halign</code>	162, 306
<code>\hbox</code>	13, 16, 22, 307, 331, 340, 382, 525, 536, 545, 591, 600, 609, 713, 714, 715, 716, 717, 718, 719, 720
hbox commands:	
<code>\hbox_set:Nn</code>	767, 776
<code>\hfil</code>	94, 133, 226, 247, 261, 265, 311, 315, 422, 431, 434, 443, 505, 515
<code>\hfill</code>	515
<code>\hfilneg</code>	322, 392, 399
<code>\hskip</code>	309, 319, 320, 323, 345, 375, 385, 390, 400
I	
<code>\ialign</code>	224, 260, 421, 504
<code>\ifcase</code>	712
<code>\ifdim</code>	342, 694
<code>\iffalse</code>	33
<code>\ifinner</code>	458
<code>\ifnum</code>	33, 212, 274
<code>\IfSocketPlugExistsF</code>	666
<code>\iftrue</code>	8
<code>\ifvmode</code>	688
<code>\ignorespaces</code>	699
<code>\ignorespacesafterend</code> ..	37, 187, 199, 288
int commands:	
<code>\int_gincr:N</code>	731
<code>\int_new:N</code>	668
<code>\int_set:Nn</code>	676
<code>\int_use:N</code>	735, 739, 743, 747
int internal commands:	
<code>\g__math_mathchoice_int</code>	731, 735, 739, 743, 747
<code>\intertext</code>	17, 687
K	
<code>\kern</code>	142, 769, 772, 773, 784, 785
L	
<code>\label</code>	18, 154, 300
<code>\lastbox</code>	523, 528, 530, 588, 594, 596
<code>\lbrace</code>	408
<code>\leavevmode</code>	767
<code>\left</code>	408
<code>\let</code>	8, 24, 153, 154, 157, 214, 217, 297, 298, 300, 373, 407, 464, 483, 717, 718, 719, 720
<code>\lineskip</code>	419, 420
<code>\lineskiplimit</code>	420
<code>\linewidth</code>	694, 695
<code>\llap</code>	376
luamml commands:	
<code>\luamml_ignore:</code>	8, 102, 121, 410
luamml internal commands:	
<code>__luamml_amsmath_save_intertext_-</code> <code>with_struct_elem:N</code>	681
<code>\l__luamml_struct_tmpa_int</code>	668, 676, 681
M	
math commands:	
<code>\math_register_halign_env:nn</code> ...	84, 85, 86, 87, 88, 89, 90, 91, 92, 201, 202, 290, 291
math internal commands:	
<code>__math_patch_amsbsy_pmb:</code> .	755, 792
<code>__math_split_debug_typeout:n</code> ..	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
<code>__math_tag_if_mathstyle:nn</code>	735, 739, 743, 747
<code>\mathchoice</code>	18, 733
<code>\mathstyle</code>	712
<code>\measuring</code>	8, 692
<code>\minalignsep</code>	218
<code>\mkern</code>	777, 781, 784, 785
<code>\moveleft</code>	537, 546, 601, 615
<code>\multiply</code>	213
<code>\multline*(env.)</code>	285
<code>\multlinegap</code>	323, 375, 390, 400
<code>\multlinetaggap</code>	342, 343, 385
N	
<code>\NewTaggingSocket</code>	656, 757
<code>\NewTaggingSocketPlug</code>	657, 669, 758
<code>\noalign</code>	617, 689
<code>\noindent</code>	699
<code>\nonumber</code>	32, 137, 481
<code>\normalbaselines</code>	330, 378, 693
<code>\normalfont</code>	13, 16, 22
<code>\null</code>	379, 413
O	
<code>\omit</code>	141
<code>\or</code> ..	713, 714, 715, 716, 717, 718, 719, 720
P	
<code>\par</code>	699
<code>\parshape</code>	695
<code>\pdfannot</code>	18

<code>\penalty</code>	690, 702	<code>\ifnextchar</code>	407
<code>\pmb</code>	20	<code>\@align</code>	98, 117, 144
<code>\postdisplaypenalty</code>	690	<code>\@mathmargin</code> ...	301, 304, 319, 320, 341
<code>\predisplaypenalty</code>	702	<code>\@one</code>	40, 45,
<code>\ProvidesFile</code>	3		70, 75, 139, 161, 220, 258, 294, 379,
			380, 381, 502, 521, 525, 591, 614, 695
Q			
<code>\quad</code>	410	<code>\@nx</code>	480, 481
R			
<code>\raise</code>	772, 784	<code>\@tempcnta</code>	611, 618
<code>\ref</code>	24	<code>\@totalleftmargin</code>	695
<code>\relax</code>	211, 273, 309, 345, 373	<code>\@xp</code>	287, 459, 464, 472, 476, 478
<code>\renewcommand</code>	203	<code>\add@amps</code>	140, 618
<code>\renewenvironment</code> ..	39, 44, 49, 54, 59,	<code>\align@</code>	4
	64, 69, 74, 79, 189, 252, 285, 412, 455	<code>\align@preamble</code>	93
<code>\rlap</code>	329	<code>\alignedspace@left</code>	209, 256
S			
<code>\scriptsize</code>	427, 439	<code>\alignsep@</code>	134, 142, 215, 218, 248
<code>\scriptstyle</code>	425, 437	<code>\ams@return@opt@carg</code>	250, 267
<code>\selectfont</code>	717, 718, 719, 720	<code>\ams@start@box</code>	210, 257
<code>\setbox</code> 340, 379, 501, 519, 521, 523, 525,		<code>\binrel@</code>	778
	528, 530, 586, 588, 591, 594, 596, 609	<code>\binrel@@</code>	780
<code>smallmatrix</code> (env.)	412	<code>\black@</code>	26, 183, 195, 494
<code>\span</code>	177	<code>\boxz@</code>	601, 631, 635, 644, 649
<code>split</code> (env.)	455	<code>\calc@shift@gather</code>	171
<code>\split</code>	153	<code>\check@mathfonts</code>	723, 750
<code>sys</code> commands:		<code>\collect@body</code>	464, 466
<code>\sys_if_engine luatex:TF</code>	707	<code>\column@</code>	223, 274, 611
T			
<code>\tabskip</code> 114, 134, 160, 173, 177, 235, 248, 301		<code>\column@plus</code>	225, 236
<code>\tag</code>	154, 297, 300, 503	<code>\common@align@ending</code>	
<code>tag</code> commands:		.. 25, 42, 47, 52, 57, 62, 67, 72, 77, 82	
<code>\tag_get:n</code>	676	<code>\default@tag</code>	221, 417, 503
<code>\tag_if_active:TF</code>	727	<code>\df@tag</code>	478, 480
<code>\tag_mc_begin:n</code>	677, 760, 764	<code>\dimen@</code>	341, 342,
<code>\tag_mc_begin_pop:n</code>	663, 682		343, 345, 768, 772, 773, 779, 784, 785
<code>\tag_mc_end:</code>	679, 760, 763	<code>\display@</code>	156
<code>\tag_mc_end_push:</code>	659, 671	<code>\dollar@end</code> 35, 186, 198, 396, 404	
<code>\tag_mc_reset_box:N</code>	761	<code>\dspbrk@context</code> 155, 220, 258, 295, 502	
<code>\tag_socket_use:nnn</code>	710	<code>\dspbrk@lvl</code>	294
<code>\tag_struct_begin:n</code>	672	<code>\endmathdisplay@a</code>	490
<code>\tag_struct_end:</code>	680	<code>\endmultline@math</code> ..	10, 314, 362, 373
<code>\tag_suspend:n</code>	8, 692, 732	<code>\env@cases</code>	406
<code>\tagpdfparaOff</code>	662	<code>env@cases</code>	406
<code>\tagpdfparaOn</code>	660	<code>\ex@</code>	418, 419, 769, 772, 773
<code>T_EX</code> and <code>L^AT_EX 2_ε</code> commands:		<code>\f@size</code> ...	717, 718, 719, 720, 736, 740
<code>\@centering</code>	173, 309	<code>\firstchoice@false</code>	740, 744, 748
<code>\@currenvir</code>	27, 206	<code>\gather@</code>	152, 488
<code>\@display@init</code>	294	<code>\gather@split</code>	13–15, 466, 474
<code>\@empty</code>	298, 464, 483, 688	<code>\gdisplaywidth@</code>	177
<code>\@eqnswtrue</code>	148, 461	<code>\gmeasure@</code>	158
<code>\@gobble</code>	300	<code>\gobble@tag</code>	300
		<code>\if@display</code>	457
		<code>\if@eqnsw</code>	138
		<code>\if@fleqn</code>	301, 303, 308, 318
		<code>\ifctagsplit@</code>	533, 552, 598, 626
		<code>\iffinalign@</code>	520, 585
		<code>\ifingather@</code>	29

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

U

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

<code>\UseTaggingSocket</code>	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
<code>\vbox</code>	329, 335, 501, 521, 537, 546, 601, 615, 691	
<code>\vcenter</code>	414, 487, 537, 557, 561, 601, 631, 635	
<code>\vskip</code>	690, 702	
<code>\vtop</code>	376, 519, 586	
		W
<code>\wd</code>	341, 536, 537, 545, 546, 600, 601, 615	