

The latex-lab-amsmath code*

L^AT_EX Project

September 24, 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	multline and multline*	9
2.5	Cases	12
2.5.1	smallmatrix	12
2.6	The split environment	13
2.7	\intertext	18
2.8	\text	19
2.9	\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 [2025-06-20 v0.1h 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 \AddToHook{begindocument}
8 {
9   \def\measuring@true{\let\ifmeasuring@\iftrue\tag_suspend:n{\measuring}\luamml_ignore:}
10 }
```

(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@@@`

```
11 \def\maketag@@@#1
12 {%
13   \ifmeasuring@
14     \hbox{\m@th\normalfont#1}%
15   \else
16     \UseTaggingSocket{math/display/tag/begin}
17     \hbox{\m@th\normalfont#1
18     \UseTaggingSocket{math/luamml/mtable/tag/save}
19     }%
20     \UseTaggingSocket{math/display/tag/end}
21   \fi
22 }
```

(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`

```

23 \def\maketag@@@notag#1{\hbox{\m@th\normalfont#1}}
24 \DeclareRobustCommand{\eqref}[1]
25   {\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`

```

26 \def\common@align@ending {
27   \math@cr \black@totwidth@
28   \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {\@currenvir}
29   \egroup
30   \ifingather@
31     \restorealignstate@
32     \egroup
33     \nonumber
34     \ifnum0=`\fi\iffalse}\fi
35   \else
36     \dollar@end
37     \fi
38     \ignorespacesafterend
39 }

```

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

Now we redefine the display alignments to use these ending.

```

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

```

```

63 \common@align@ending
64 }
65 \renewenvironment{flalign*}{%
66 \start@align\tw@\st@rredtrue\m@ne
67 }{%
68 \common@align@ending
69 }
70 \renewenvironment{xalignat}{%
71 \start@align\@ne\st@rredfalse
72 }{%
73 \common@align@ending
74 }
75 \renewenvironment{xalignat*}{%
76 \start@align\@ne\st@rredtrue
77 }{%
78 \common@align@ending
79 }
80 \renewenvironment{xxalignat}{%
81 \start@align\tw@\st@rredtrue
82 }{%
83 \common@align@ending
84 }

```

And register these environments for the math grabbing.

```

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

```

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

\align@preamble

```

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

```

```

110     \savefieldlength@
111     \else
112         \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
113     \fi
114     \set@field
115     \tabskip\z@skip
116     &\setboxz@h
117     {
118         \@lign
119         $
120         \m@th\displaystyle{ {}##}
121         \ifmeasuring@
122             \luamml_ignore:
123         \else
124             \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
125         \fi
126         $
127     }%
128     \ifmeasuring@
129         \savefieldlength@
130     \else
131         \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
132     \fi
133     \set@field
134     \hfil
135     \tabskip\alignsep@
136 }

```

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

\math@cr@@@align

```

137 \def\math@cr@@@align{%
138     \ifst@rred\nonumber\fi
139     \if@eqnsw \global\tag@true \fi
140     \global\advance\row@ \@ne
141     \add@amps\maxfields@
142     \omit
143     \kern-\alignsep@
144     \iftag@
145         \setboxz@h{\@lign\strut@{\make@display@tag}}%
146         \place@tag
147     \fi
148     \UseTaggingSocket{math/luamml/mtable/tag/set}
149     \ifst@rred\else\global\@eqnswtrue\fi
150     \global\lineht@\z@
151     \cr
152 }

```

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

2.4.3 gather and gather*

\gather@

```

153 \def\gather@#1{%
154     \ingather@true \let\split\insplit@
155     \let\tag\tag@in@align \let\label\label@in@display
156     \chardef\dspbrk@context\z@
157     \intertext@ \display@ \Let@
158     \let\math@cr@@@math@cr@@@gather
159     \gmeasure@{#1}%
160     \global\shifttag@false
161     \tabskip\z@skip
162     \global\row@\@ne
163     \halign to\displaywidth\bgroup
164         \strut@
165         \setboxz@h
166         {
167             $\m@th\displaystyle{##}
168             \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
169             $
170             }%
171         \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
172         \calc@shift@gather
173         \set@gather@field
174         \tabskip\@centering
175         &\setboxz@h{\strut@{##}}}%
176         \place@tag@gather
177         \UseTaggingSocket{math/luamml/mtable/tag/set}
178         \tabskip \iftagsleft@ \gdisplaywidth@ \else \z@skip \span\fi
179         \crr
180         #1%
181     }

```

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

`\endgather`

```

182 \def\endgather{
183     \math@cr
184     \black@ \totwidth@
185     \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
186     \egroup
187     \dollar@end
188     \ignorespacesafterend
189 }

```

(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.)`

```

190 \renewenvironment{gather*}
191 {
192     \start@gather\st@rrredtrue
193 }

```

```

194 {
195   \math@cr
196   \black@totwidth@
197   \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
198   \egroup
199   \dollar@end
200   \ignorespacesafterend
201 }

```

Register both environments

```

202 \math_register_halign_env:nn {gather}{}
203 \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`

```

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

```

```

237      & \column@plus
238      $
239      \m@th
240      \displaystyle
241      {
242      {}
243      ##
244      }
245      \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
246      $
247      \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
248      \hfil
249      \tabskip\alignsep@
250      \crrc
251      \ams@return@opt@arg
252 }

```

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

gathered (env.)

```

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

```

This annotates the space

```

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

```

This save the cell and then finalize it.

```

263   \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
264   $
265   \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
266   \hfil
267   \crrc
268   \ams@return@opt@arg
269 }{%
270   \endaligned
271 }

```

\endaligned

```

272 \def\endaligned
273 {
274   \crrc
275   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}
276   \egroup

```



```

277 \restorecolumn@
278 \egroup
279 \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
280 }

```

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

2.4.5 `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.*)

```

281 \renewenvironment{multline*}{\start@multline\st@rredtrue}
282 {%
283 \iftagsleft@ \exp\lendmultline@ \else \exp\rendmultline@ \fi
284 \ignorespacesafterend
285 }

```

And now we register both

```

286 \math_register_halign_env:n {multline}{}
287 \math_register_halign_env:n {multline*}{}

```

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

`\multline@`

```

288 \def\multline@#1{%
289 \let@
290 \@display@init{\global\advance\row@ \@one \global\dspbrk@lvl\m@ne}%
291 \chardef\dspbrk@context\z@
292 \restore@math@cr
293 \let\tag\tag@in@align
294 \global\tag@false \global\let\raise@tag@empty
295 \mmeasure@{#1}%
296 \let\tag@gobble@tag \let\label@\gobble
297 \tabskip \if@fleqn \@mathmargin \else \z@skip \fi
298 \totwidth@\displaywidth
299 \if@fleqn
300 \advance\totwidth@-\@mathmargin
301 \fi
302 \halign\bgroup
303 \hbox to\totwidth@{%
304 \if@fleqn
305 \hskip \@centering \relax
306 \else
307 \hfil
308 \fi
309 \strut@
310 $\m@th\displaystyle{##}\endmultline@math
311 \hfil
312 }%
313 \crcr

```

```

314 \if@fleqn
315 \hskip-\@mathmargin
316 \def\multline@indent{\hskip\@mathmargin}% put it back
317 \else
318 \hfilneg
319 \def\multline@indent{\hskip\multlinegap}%
320 \fi
321 \iftagsleft@
322 \iftag@
323 \begingroup
324 \ifshifttag@
325 \rlap{\vbox{%
326 \normalbaselines
327 \hbox{%
328 \strut@
329 \make@display@tag
330 }%
331 \vbox to\lineht@{}}%
332 \raise@tag
333 }}%
334 \multline@indent
335 \else
336 \setbox\z@\hbox{\make@display@tag}%
337 \dimen@\@mathmargin \advance\dimen@-\wd\z@
338 \ifdim\dimen@<\multlinetaggap
339 \dimen@\multlinetaggap
340 \fi
341 \box\z@ \hskip\dimen@\relax
342 \fi
343 \endgroup
344 \else
345 \multline@indent
346 \fi
347 \else
348 \multline@indent
349 \fi
350 \ifmeasuring@ \else
351 \UseTaggingSocket{math/luamml/mtable/aligncol} {left}
352 \fi
353 #1%
354 \ifmeasuring@ \else
355 \UseTaggingSocket{math/luamml/mtable/aligncol} {right}
356 \fi
357 }

```

(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`

```

358 \def\endmultline@math
359 {

```

```

360   \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
361   $
362   \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
363 }

```

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

`\rendmultiline@`

```

364 \def\rendmultiline@{%
365   \iftag@

```

we need to surround the math token with tagging sockets.

```

366   \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
367   $
368   \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
369   \let\endmultiline@math\relax
370   \ifshifttag@
371     \hskip\multlinegap
372     \llap{\vtop{%
373       \raise@tag
374       \normalbaselines
375       \setbox\@ne\null
376       \dp\@ne\lineht@
377       \box\@ne
378       \hbox{\strut@make@display@tag}%
379     }}%
380   \else
381     \hskip\multlinetaggap
382     \make@display@tag
383   \fi

```

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

```

384   \UseTaggingSocket{math/luamml/mtable/tag/set}
385   \else
386     \hskip\multlinegap
387   \fi
388   \hfilneg
389   \math@cr

```

Now we finalize the mtable.

```

390   \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
391   \egroup
392   \dollar@end
393 }

```

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

And something similar for the left version

`\lendmultiline@`

```

394 \def\lendmultline@{%
395     \hfilneg
396     \hskip\multlinegap
397     \math@cr
398     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
399     \egroup
400     \dollar\dollar@end
401 }

```

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

2.5 Cases

env@cases

```

402 \def\env@cases{%
403     \let@ifnextchar\new@ifnextchar
404     \left\lbrace
405     \def\arraystretch{1.2}%
406     \array{@{}l@{\quad}l@{\luamml_ignore:}}%
407 }

```

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

2.5.1 smallmatrix

smallmatrix (env.)

```

408 \renewenvironment {smallmatrix} {
409     \UseTaggingSocket{ math/luamml/annotate/false } {} { \null\, }
410     \vcenter \bgroup
411     \Let@
412     \restore@math@cr
413     \default@tag
414     \baselineskip 6 \ex@
415     \lineskip 1.5 \ex@
416     \lineskiplimit \lineskip
417     \ialign \bgroup
418         \hfil
419         $
420         \m@th
421         \scriptstyle
422         ##
423         % No \scriptsize here since we want to add the mstyle nodes
424         \UseTaggingSocket{math/luamml/save/nn}{ {} {mtd} }
425         $
426         \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
427         \hfil
428     &&
429     \thickspace
430     \hfil
431     $
432     \m@th
433     \scriptstyle

```

```

434     ##
435     % No \scriptsize here since we want to add the mstyle nodes
436     \UseTaggingSocket{math/luamml/save/nn}{ }{ }{mtd} }
437     $
438     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
439     \hfil
440     \crrcr
441 }{%
442     \crrcr
443     \UseExpandableTaggingSocket{math/luamml/mtable/smallmatrix/save}
444     \egroup
445     \egroup
446     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
447     \UseTaggingSocket{ math/luamml/annotate/false } { }{ \, }
448 }

```

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 `tbtag`s is used. The alignment of the cells is currently not handled. A simple debug command until everything is sorted.

`_math_split_debug_typeout:n`

```

449 \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`

```

450 \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.)`

```

451 \renewenvironment{split}{%
452   \_math\_split\_debug\_typeout:n {begin-split}
453   \if@display
454     \ifinner
455       \@xp\@xp\@xp\split@aligned
456     \else
457       \ifst@rred \else \global\@eqnswtrue \fi
458     \fi
459   \else
460     \let\endsplit\@empty \@xp\collect@body\@xp\split@err

```

```

461 \fi
462 \collect@body\gather@split
463 }\_math_split_debug_typeout:n{end~split}%
464 \crr
465 \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}%
466 \egroup
467 \egroup
468 \iftagsleft@ \exp\lendsplit@ \else \exp\rendsplitsplit@ \fi
469 }

```

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`

```

470 \def\gather@split#1#2#3{
471 \_math_split_debug_typeout:n{gather@split}%
472 \exp\endgroup \reset@equation % math@cr will handle equation numbering
473 \iftag@
474 \toks@{\exp{\df@tag}}%
475 \edef\split@tag{%
476 \gdef\@nx\df@tag{\the\toks@}%
477 \global\@nx\tag@true \@nx\nonumber
478 }%
479 \else \let\split@tag\empty
480 \fi
481 \bool_set_true:N\l__math_gathersplit_bool
482 \spread@equation
483 \vcenter\bgroup
484 \gather@{\split@tag
485 \begin{split}#1\end{split}}}%
486 \def\endmathdisplay@a{%
487 \_math_split_debug_typeout:n{endmathdisplay@a}
488 \_math_split_debug_typeout:n{finalize~innertable~endmathdisplay@a}
489 \math@cr
490 \black@ \totwidth@
491 \egroup
492 \egroup
493 \UseExpandableTaggingSocket{math/luamml/mtable/innertable/finalize}%
494 }%
495 }

```

(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.

```

496 \def\insplit@{\_math_split_debug_typeout:n{insplit@}%
497 \global\setbox\z@\vbox\bgroup
498 \Let@ \chardef\dsprk@context\@ne \restore@math@cr
499 \default@tag % disallow use of \tag here
500 \ialign\bgroup
501 \hfil
502 \strut@

```

```

503      $\m@th\displaystyle {##}%
504      \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
505      $%
506      \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
507      &$\m@th\displaystyle {{}##}%
508      \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }%
509      $%
510      \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
511      \hfill % Why not \hfil?---dmj, 1994/12/28
512      \crrr
513  }

```

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

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

```

514 \def\lendsplit@{%
515   \global\setbox9\vtop{\unvcopy\z@}%
516   \iffinalign@
517     \setbox\@ne\vbox{%
518       \unvcopy\z@
519       \global\setbox8\lastbox
520     }%
521     \setbox\@ne\hbox{%
522       \unhcopy8%
523       \unskip
524       \setbox\tw@\lastbox
525       \unskip
526       \global\setbox\thr@@\lastbox
527     }%
528     \_math_split_debug_typeout:n{lendsplit@/aligncase}
529     \ifctagsplit@
530       \_math_split_debug_typeout:n{lendsplit@/aligncase/centertags}
531       \gdef\split@{%
532         \hbox to\wd\thr@@{%
533           &\vcenter{\vbox{\moveleft\wd\thr@@\box9}}%
534           \_math_split_debug_typeout:n{finalize~innertable~aligncase}
535           \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
536         }%
537       \else
538       %TODO, not correct yet.
539       \_math_split_debug_typeout:n{lendsplit@/aligncase/tbtags}
540       \gdef\split@{%
541         \hbox to\wd\thr@@{%
542           &\vbox{\moveleft\wd\thr@@\box9}%
543           \_math_split_debug_typeout:n{finalize~innertable~aligncase}
544           \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
545         }%
546       \fi
547     \else

```

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

```

548 \ifctagsplit@
549 \bool_if:NTF \l__math_gathersplit_bool
550 {
551   \__math_split_debug_typeout:n{lendsplit/equationcase/centertags}
552   \gdef\split@%
553     {\UseTaggingSocket{math/luamml/annotate/false}{\vcenter{\box9}}}
554 }
555 {
556   \__math_split_debug_typeout:n {lendsplit/gathercase/centertags}
557   \gdef\split@{\vcenter{\box9}%
558     \__math_split_debug_typeout:n {finalize~innertable~gathercase}
559     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
560 }
561 \else
562 % TODO tbtags not correct yet.
563 \bool_if:NTF \l__math_gathersplit_bool
564 {
565   \__math_split_debug_typeout:n {lendsplit/equationcase/tbtags}
566   \gdef\split@%
567     {\UseTaggingSocket{math/luamml/annotate/false}{\box9}}
568 }
569 {
570   \__math_split_debug_typeout:n {lendsplit/gathercase/tbtags}
571   \gdef\split@{
572     \box9%
573     \__math_split_debug_typeout:n {finalize~innertable~gathercase}
574     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
575 }
576 \fi
577 \fi
578 \aftergroup\split@
579 }

```

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

```

580 \def\rendsplit@{%
581   \iffinalign@
582     \global\setbox9 \vtop{%
583       \unvcopy\z@
584       \global\setbox8 \lastbox
585       \unskip
586     }%
587     \setbox\@ne\hbox{%
588       \unhcopy8
589       \unskip
590       \global\setbox\tw@\lastbox
591       \unskip
592       \global\setbox\thr@@\lastbox
593     }%
594     \ifctagsplit@
595       \gdef\split@{%
596         \hbox to\wd\thr@@{
597           &\vcenter{\vbox{\moveleft\wd\thr@@\boxz@}}%
598         }
599       \__math_split_debug_typeout:n {rendsplit/aligncase/centertags}

```



```

599         \_math_split_debug_typeout:n {finalize~innertable~aligncase}
600         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
601     }%
602 \else
603     \_math_split_debug_typeout:n{rendsplit@/aligncase/tbtags}
604     %TODO tbtags is not correct yet
605     \global\setbox7 \hbox{\unhbox\tw@\unskip}%
606     \gdef\split@{%
607         \global\@tempcnta\column@
608         &\setboxz@h}%
609         \savetaglength@
610         \global\advance\row@\@ne
611         \vbox{\moveleft\wd\thr@@\box9}%
612     \crr
613     \noalign{\global\lineht@\z@}%
614     \add@amps\@tempcnta
615     \UseTaggingSocket{math/luamml/annotate/false}{}{\box\thr@@}
616     &\box7
617     \_math_split_debug_typeout:n {finalize~innertable~aligncase}
618     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
619 }%
620 \fi

```

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

```

621 \else
622     \ifctagsplit@
623     \bool_if:NTF \l__math_gathersplit_bool
624     {
625         \_math_split_debug_typeout:n {rendsplit/equationcase/centertags}
626         \gdef\split@%
627         {\UseTaggingSocket{math/luamml/annotate/false}{}{\vcenter{\boxz@}}}
628     }
629     {
630         \_math_split_debug_typeout:n {rendsplit/gathercase/centertags}
631         \gdef\split@{\vcenter{\boxz@}%
632         \_math_split_debug_typeout:n {finalize~innertable~gathercase}
633         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
634     }
635 \else
636     \bool_if:NTF \l__math_gathersplit_bool
637     {
638         \_math_split_debug_typeout:n {rendsplit/equationcase/tbtags}
639         \gdef\split@{%
640             \UseTaggingSocket{math/luamml/annotate/false}{}{\boxz@}
641         }
642     }
643     {
644         \_math_split_debug_typeout:n {rendsplit/gathercase/tbtags}
645         \gdef\split@{%
646             \boxz@
647             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
648     }
649 \fi

```

```

650 \aftergroup\split@
651 }

```

(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!

```

652 \def\intertext@{%
653   \def\intertext##1{%
654     \ifvmode\else\\\@empty\fi
655     \noalign{%
656       \penalty\postdisplaypenalty\vskip\belowdisplayskip
657       \vbox{

```

Stop tagging when measuring:

```

658       \ifmeasuring@\tag_suspend:n{\measuring}\fi
659       \normalbaselines
660       \ifdim\linewidth=\columnwidth
661       \else \parshape\@ne \@totalleftmargin \linewidth
662       \fi

```

End the previous mc:

```

663       \tag_mc_end_push:

```

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

```

664       \tagpdfsetup{para/tag=P}%

```

TODO why `\tagpdfpara0n` needed?

```

665       \tagpdfpara0n
666       \noindent\ignorespaces##1\par

```

Restart the MC

```

667       \tag_mc_begin_pop:n{}}%
668       \penalty\predisplaypenalty\vskip\abovedisplayskip%
669   }%
670 }}

```

(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@`

```
671 \AddToHook{package/amstext/after}
672 {
```

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.

```
673   \tag_if_active:T
674   {
675     \sys_if_engine_luatex:TF
676     {
677       \def\text@#1{
678         \tag_socket_use:nnn {math/luamml/hbox}{ }
679         { { %
680           \ifcase\mathstyle
681             \hbox{{#1}}\or
682             \hbox{{#1}}\or
683             \hbox{{#1}}\or
684             \hbox{{#1}}\or
685             \hbox{{\let\f@size\sf@size\selectfont#1}}\or
686             \hbox{{\let\f@size\sf@size\selectfont#1}}\or
687             \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
688             \hbox{{\let\f@size\ssf@size\selectfont#1}}\or
689             \ERROR
690             \fi
691             \check@mathfonts
692           } } }
693     }
694     {
695       \def\text@#1
696       { {
697         \int_gincr:N\g__math_mathchoice_int
698         \tag_suspend:n{\text@}
699         \mathchoice
700         {
701           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{0}
702           \textdef@\displaystyle\f@size{#1}
```

```

703     }
704     {
705         \_math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{2}
706         \textdef@{\textstyle\int@size{\firstchoice@false #1}}
707     }
708     {
709         \_math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{4}
710         \textdef@{\textstyle\intsf@size{\firstchoice@false #1}}
711     }
712     {
713         \_math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{6}
714         \textdef@{\textstyle \intssf@size{\firstchoice@false #1}}
715     }
716     \check@mathfonts
717 }}
718 }
719 }
720 }

```

(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@
721 \AddToHook{package/amsbsy/after}
722 {
723     \def\pmb@@#1#2#3{\leavevmode\hbox_set:Nn\l__math_tmpa_box{xxx#3}
724         \dimen@-\box_wd:N\l__math_tmpa_box
725         \kern-.5\ex@\box_use:N\l__math_tmpa_box
726         \tag_mc_end:\tag_mc_begin:n{artifact}
727         \tag_mc_reset_box:N\l__math_tmpa_box
728         \kern\dimen@\kern.25\ex@\raise.4\ex@\box_use:N\l__math_tmpa_box
729         \kern\dimen@\kern.25\ex@\box_use_drop:N\l__math_tmpa_box
730         \tag_mc_end:\tag_mc_begin:n{}}
731 }
732 \def\pmb@#1#2{\hbox_set:Nn\l__math_tmpa_box{\$m@th#1{#2}$}
733     \setboxz@h{\$m@th#1\mkern.5mu}\pmbraise@wdz@
734     \binrel@{#2}
735     \dimen@-\box_wd:N\l__math_tmpa_box
736     \binrel@@{
737         \mkern-.8mu\box_use:N\l__math_tmpa_box
738         \tag_mc_end:\tag_mc_begin:n{artifact}
739         \tag_mc_reset_box:N\l__math_tmpa_box
740         \kern\dimen@\mkern.4mu\raise\pmbraise@\box_use:N\l__math_tmpa_box
741         \kern\dimen@\mkern.4mu\box_use_drop:N\l__math_tmpa_box
742         \tag_mc_end:\tag_mc_begin:n{}}
743     }
744 }
745 }

```

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

⁷⁴⁶ `\ExplSyntaxOff`

⁷⁴⁷ `\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		
<code>\,</code>	409, 447
<code>\%</code>	654
A		
<code>\abovedisplayskip</code>	668
<code>\AddToHook</code>	7, 671, 721
<code>\advance</code>	140, 290, 300, 337, 610
<code>\aftergroup</code>	578, 650
<code>\array</code>	406
<code>\arraystretch</code>	405
B		
<code>\baselineskip</code>	414
<code>\begin</code>	207, 255, 485
<code>\begingroup</code>	323
<code>\belowdisplayskip</code>	656
<code>\bgroup</code>	163, 211, 225, 258, 261, 302, 410, 417, 483, 497, 500
bool commands:		
<code>\bool_if:NTF</code>	549, 563, 623, 636
<code>\bool_new:N</code>	450
<code>\bool_set_true:N</code>	481
bool internal commands:		
<code>\l_math_gathersplit_bool</code>	450, 481, 549, 563, 623, 636
<code>\box</code>	341, 377, 533, 542, 553, 557, 567, 572, 611, 615, 616
box commands:		
<code>\box_use:N</code>	725, 728, 737, 740
<code>\box_use_drop:N</code>	729, 741
<code>\box_wd:N</code>	724, 735
box internal commands:		
<code>\l_math_tmpa_box</code>	723, 724, 725, 727, 728, 729, 732, 735, 737, 739, 740, 741	
C		
<code>\chardef</code>	156, 221, 259, 291, 498
<code>\columnwidth</code>	660
<code>\cr</code>	151
<code>\crrcr</code>	179, 250, 267, 274, 313, 440, 442, 464, 512, 612
cs commands:		
<code>\cs_new:Npn</code>	449
D		
<code>\DeclareRobustCommand</code>	24
<code>\def</code>	9, 11, 23, 26, 94, 137, 153, 182, 272, 288, 316, 319, 358, 364, 394, 402, 405, 470, 486, 496, 514, 580, 652, 653, 677, 695, 723, 732
<code>\displaystyle</code>	101, 105, 120, 124, 167, 168, 231, 233, 240, 245, 262, 263, 310, 360, 366, 503, 504, 507, 508, 702
<code>\displaywidth</code>	163, 298
<code>\dp</code>	376
E		
<code>\edef</code>	475
<code>\egroup</code>	29, 32, 186, 198, 276, 278, 391, 399, 444, 445, 466, 467, 491, 492
<code>\else</code>	15, 35, 104, 111, 123, 130, 149, 178, 206, 217, 254, 283, 297, 306, 317, 335, 344, 347, 350, 354, 380, 385, 456, 457, 459, 468, 479, 537, 547, 561, 602, 621, 635, 654, 661
<code>\end</code>	485
<code>\endaligned</code>	270, 272
<code>\endgather</code>	6, 182
<code>\endgroup</code>	343, 472
<code>\endmultline</code>	9
<code>\endsplit</code>	460
environments:		
<code>gather*</code>	190
<code>gathered</code>	253
<code>multline*</code>	281
<code>smallmatrix</code>	408
<code>split</code>	451
<code>\eqref</code>	2, 23
<code>\ERROR</code>	689
<code>\ExplSyntaxOff</code>	746

<code>\ExplSyntaxOn</code>	6		
F		K	
<code>\fi</code>	21, 34, 37, 106, 113, 125, 132, 138, 139, 147, 149, 178, 208, 220, 256, 283, 297, 301, 308, 320, 340, 342, 346, 349, 352, 356, 383, 387, 457, 458, 461, 468, 480, 546, 576, 577, 620, 648, 649, 654, 658, 662, 690	<code>\kern</code>	143, 725, 728, 729, 740, 741
G		L	
<code>gather* (env.)</code>	190	<code>\label</code>	19, 155, 296
<code>gathered (env.)</code>	253	<code>\lastbox</code>	519, 524, 526, 584, 590, 592
<code>\gathersplit</code>	17	<code>\lbrace</code>	404
<code>\gdef</code>	476, 531, 540, 552, 557, 566, 571, 595, 606, 626, 631, 639, 644	<code>\leavevmode</code>	723
<code>\global</code>	139, 140, 149, 150, 160, 162, 224, 290, 294, 457, 477, 497, 515, 519, 526, 582, 584, 590, 592, 605, 607, 610, 613	<code>\left</code>	404
H		<code>\let</code>	9, 25, 154, 155, 158, 215, 218, 293, 294, 296, 369, 403, 460, 479, 685, 686, 687, 688
<code>\halign</code>	163, 302	<code>\lineskip</code>	415, 416
<code>\hbox</code>	14, 17, 23, 303, 327, 336, 378, 521, 532, 541, 587, 596, 605, 681, 682, 683, 684, 685, 686, 687, 688	<code>\lineskiplimit</code>	416
hbox commands:		<code>\linewidth</code>	660, 661
<code>\hbox_set:Nn</code>	723, 732	<code>\llap</code>	372
<code>\hfil</code>	95, 134, 227, 248, 262, 266, 307, 311, 418, 427, 430, 439, 501, 511	luamml commands:	
<code>\hfill</code>	511	<code>\luamml_ignore:</code>	9, 103, 122, 406
<code>\hfilneg</code>	318, 388, 395	M	
<code>\hskip</code>	305, 315, 316, 319, 341, 371, 381, 386, 396	math commands:	
I		<code>\math_register_halign_env:n</code>	85, 86, 87, 88, 89, 90, 91, 92, 93, 202, 203, 286, 287
<code>\ialign</code>	225, 261, 417, 500	math internal commands:	
<code>\ifcase</code>	680	<code>__math_split_debug_typeof:n</code>	449, 449, 452, 463, 471, 487, 488, 496, 528, 530, 534, 539, 543, 551, 556, 558, 565, 570, 573, 598, 599, 603, 617, 625, 630, 632, 638, 643
<code>\ifdim</code>	338, 660	<code>__math_tag_if_mathstyle:n</code>	701, 705, 709, 713
<code>\iffalse</code>	34	<code>\mathchoice</code>	19, 699
<code>\ifinner</code>	454	<code>\mathstyle</code>	680
<code>\ifnum</code>	34, 213	<code>\measuring</code>	9, 658
<code>\iftrue</code>	9	<code>\minalignsep</code>	219
<code>\ifvmode</code>	654	<code>\mkern</code>	733, 737, 740, 741
<code>\ignorespaces</code>	666	<code>\moveleft</code>	533, 542, 597, 611
<code>\ignorespacesafterend</code> ..	38, 188, 200, 284	<code>\multiply</code>	214
int commands:		<code>\multline* (env.)</code>	281
<code>\int_gincr:N</code>	697	<code>\multlinegap</code>	319, 371, 386, 396
<code>\int_use:N</code>	701, 705, 709, 713	<code>\multlinetaggap</code>	338, 339, 381
int internal commands:		N	
<code>\g__math_mathchoice_int</code>	697, 701, 705, 709, 713	<code>\noalign</code>	613, 655
<code>\intertext</code>	18, 653	<code>\noindent</code>	666
		<code>\nonumber</code>	33, 138, 477
		<code>\normalbaselines</code>	326, 374, 659
		<code>\normalfont</code>	14, 17, 23
		<code>\null</code>	375, 409
		O	
		<code>\omit</code>	142
		<code>\or</code> ..	681, 682, 683, 684, 685, 686, 687, 688

P	
<code>\par</code>	666
<code>\parshape</code>	661
<code>\pdfannot</code>	19
<code>\penalty</code>	656, 668
<code>\pmb</code>	20
<code>\postdisplaypenalty</code>	656
<code>\predisplaypenalty</code>	668
<code>\ProvidesFile</code>	3
Q	
<code>\quad</code>	406
R	
<code>\raise</code>	728, 740
<code>\ref</code>	25
<code>\relax</code>	212, 305, 341, 369
<code>\renewcommand</code>	204
<code>\renewenvironment</code> ..	40, 45, 50, 55, 60, 65, 70, 75, 80, 190, 253, 281, 408, 451
<code>\RequirePackage</code>	5
<code>\rlap</code>	325
S	
<code>\scriptsize</code>	423, 435
<code>\scriptstyle</code>	421, 433
<code>\selectfont</code>	685, 686, 687, 688
<code>\setbox</code> 336, 375, 497, 515, 517, 519, 521, 524, 526, 582, 584, 587, 590, 592, 605	
<code>smallmatrix (env.)</code>	408
<code>\span</code>	178
<code>split (env.)</code>	451
<code>\split</code>	154
sys commands:	
<code>\sys_if_engine luatex:TF</code>	675
T	
<code>\tabskip</code> 115, 135, 161, 174, 178, 236, 249, 297	
<code>\tag</code>	155, 293, 296, 499
tag commands:	
<code>\tag_if_active:TF</code>	673
<code>\tag_mc_begin:n</code> ...	726, 730, 738, 742
<code>\tag_mc_begin_pop:n</code>	667
<code>\tag_mc_end:</code>	726, 730, 738, 742
<code>\tag_mc_end_push:</code>	663
<code>\tag_mc_reset_box:N</code>	727, 739
<code>\tag_socket_use:nnn</code>	678
<code>\tag_suspend:n</code>	9, 658, 698
<code>\tagpdfparaOn</code>	18, 665
<code>\tagpdfsetup</code>	664
T _E X and L ^A T _E X 2 _ε commands:	
<code>\@centering</code>	174, 305
<code>\@currenvir</code>	28, 207
<code>\@display@init</code>	290
<code>\@empty</code>	294, 460, 479, 654
<code>\@eqnswtrue</code>	149, 457
<code>\@gobble</code>	296
<code>\@ifnextchar</code>	403
<code>\@lign</code>	99, 118, 145
<code>\@mathmargin</code> ...	297, 300, 315, 316, 337
<code>\@ne</code>	41, 46, 71, 76, 140, 162, 221, 259, 290, 375, 376, 377, 498, 517, 521, 587, 610, 661
<code>\@nx</code>	476, 477
<code>\@tempcnta</code>	607, 614
<code>\@totalleftmargin</code>	661
<code>\exp</code>	283, 455, 460, 468, 472, 474
<code>\add@amps</code>	141, 614
<code>\align@</code>	4
<code>\align@preamble</code>	94
<code>\alignedspace@left</code>	210, 257
<code>\alignsep@</code> ...	135, 143, 216, 219, 249
<code>\ams@return@opt@arg</code>	251, 268
<code>\ams@start@box</code>	211, 258
<code>\binrel@</code>	734
<code>\binrel@@</code>	736
<code>\black@</code>	27, 184, 196, 490
<code>\boxz@</code>	597, 627, 631, 640, 645
<code>\calc@shift@gather</code>	172
<code>\check@mathfonts</code>	691, 716
<code>\collect@body</code>	460, 462
<code>\column@</code>	224, 607
<code>\column@plus</code>	226, 237
<code>\common@align@ending</code>	
... 26, 43, 48, 53, 58, 63, 68, 73, 78, 83	
<code>\default@tag</code>	222, 413, 499
<code>\df@tag</code>	474, 476
<code>\dimen@</code>	337, 338, 339, 341, 724, 728, 729, 735, 740, 741
<code>\display@</code>	157
<code>\dollar@end</code> 36, 187, 199, 392, 400	
<code>\dspbrk@context</code> 156, 221, 259, 291, 498	
<code>\dspbrk@lvl</code>	290
<code>\endmathdisplay@a</code>	486
<code>\endmultline@math</code> ...	10, 310, 358, 369
<code>\env@cases</code>	402
<code>env@cases</code>	402
<code>\ex@</code>	414, 415, 725, 728, 729
<code>\f@size</code> ...	685, 686, 687, 688, 702, 706
<code>\firstchoice@false</code> ...	706, 710, 714
<code>\gather@</code>	153, 484
<code>\gather@split</code>	13–15, 462, 470
<code>\gdisplaywidth@</code>	178
<code>\gmeasure@</code>	159
<code>\gobble@tag</code>	296
<code>\if@display</code>	453
<code>\if@eqnsw</code>	139
<code>\if@fleqn</code>	297, 299, 304, 314
<code>\ifctagsplit@</code> ...	529, 548, 594, 622

\ifinalign@	516, 581
\ifingather@	30
\ifmeasuring@	9,
13, 102, 109, 121, 128, 350, 354, 658		
\iftshifttag@	324, 370
\ifst@rred	138, 149, 457
\iftag@	144, 322, 365, 473
\iftagsleft@	178, 283, 321, 468
\lingather@true	154
\insplit@	14, 154, <u>496</u>
\intertext@	157, <u>652</u>
\label@in@display	155
\lendmultline@	283, <u>394</u>
\lendsplit@	15, 468, 514
\Let@	157, 221, 259, 289, 411, 498
\lineht@	150, 331, 376, 613
\m@ne	41, 46, 61, 66, 213, 290
\m@th	14,
17, 23, 101, 120, 167, 230, 239,		
262, 310, 420, 432, 503, 507, 732, 733		
\make@display>tag	
.....		145, 329, 336, 378, 382
\maketag@@@	2, <u>11</u> , 25
\maketag@@@notag	23, 25
\maketag@@@notog	<u>23</u>
\math@crr .. .	27, 183, 195, 389, 397, 489	
\math@cr@@@	158, 215, 218
\math@cr@@@align	<u>137</u>
\math@cr@@@aligned	218
\math@cr@@@alignedat	215
\math@cr@@@gather	158
\maxfields@	141, 212, 213, 214
\measuring@true	<u>7</u>
\mmmeasure@	295
\multiline@	<u>288</u>
\multiline@indent	316, 319, 334, 345, 348	
\new@ifnextchar	403
\nonmatherr@	207, 255
\pbm@@	<u>721</u>
\place@tag	146
\place@tag@gather	176
\pmb@	<u>721</u>
\pmb@@	723
\pmbraise@	733, 740
\raise@tag	294, 332, 373
\rendmultline@	283, <u>364</u>
\rendsplitlet	15, 16, 468, <u>580</u>
\reset@equation	472
\restore@math@cr ..	259, 292, 412, 498	
\restorealgnstate@	31
\restorecolumn@	277
\RIfM@	205, 254
\rowc@	140, 162, 290, 610
\savecolumn@	209
\savefieldlength@	110, 129
\savetaglengthe	609
\set@field	114, 133
\set@gatherefield	173
\setboxz@h	
.....		97, 116, 145, 165, 175, 608, 733
\sf@size	685, 686, 710
\shiffitag@false	160
\split@	531, 540, 552, 557, 566, 571,	
578, 595, 606, 626, 631, 639, 644, 650		
\splitlealigned	455
\splitleerr	460
\splitletag	475, 479, 484
\spread@equation	223, 260, 482
\ssf@size	687, 688, 714
\storredfalse	41, 51, 61, 71
\storreddtrue	46, 56, 66, 76, 81, 192, 281	
\start@algn	
.....		41, 46, 51, 56, 61, 66, 71, 76, 81
\start@aligned	204
\start@gatheer	192
\start@multline	281
\strut@	96, 145,
164, 175, 228, 262, 309, 328, 378, 502		
\tag@false	294
\tag@in@align	155, 293
\tag@true	139, 477
\tagform@	2, 25
\text@	<u>671</u>
\texdef@	702, 706, 710, 714
\thro@@	526, 532,
533, 541, 542, 592, 596, 597, 611, 615		
\toks@	474, 476
\totwidth@	
.....		27, 184, 196, 298, 300, 303, 490
\tw@	61, 66, 81, 214, 524, 590, 605
\wdze	733
\ze	51, 56, 150, 156, 224, 291,
336, 337, 341, 497, 515, 518, 583, 613		
\zeskip	... 115, 161, 178, 216, 236, 297	
\text	19
\textstyle	706, 710, 714
\textup	25
\the	476
\thickspace	429
U		
\unhbox	605
\unhcopy	522, 588
\unskip	523, 525, 585, 589, 591, 605
\uncvopy	515, 518, 583
use commands:		
\use none:n	449

