

# The latex-lab-mathtagging code\*

Frank Mittelbach, Joseph Wright, L<sup>A</sup>T<sub>E</sub>X Project

June 8, 2023

## Abstract

This is an experimental prototype. It captures math material (basically okay, but the interfaces for packages aren't yet there) and tags the material (which is not yet anywhere near the final state). That part is provided for experimentation and to gather feedback, etc.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
	1.1 Code level interfaces . . . . .	2
	1.2 Document level interfaces . . . . .	2
<b>2</b>	<b>Known current bugs, etc.</b>	<b>2</b>
	2.1 Capture/grabbing problems . . . . .	3
	2.2 Other problems . . . . .	3
	2.3 Other ToDos . . . . .	3
<b>3</b>	<b>The Implementation</b>	<b>3</b>
	3.1 File declaration . . . . .	3
	3.2 Setup . . . . .	4
	3.3 Data structures . . . . .	4
	3.4 Interface commands . . . . .	4
	3.5 Content grabbing . . . . .	5
	3.6 Marking math environments . . . . .	6
	3.7 Document commands . . . . .	11
	3.8 <code>\everymath</code> and <code>\everydisplay</code> . . . . .	12
	3.9 Modifying kernel environments . . . . .	14
	3.10 Modifying <code>amsmath</code> . . . . .	14
	<b>Index</b>	<b>21</b>

---

\*

# 1 Introduction

Todo: update all the documentation! Both here and (what little there is!) in the implementation section.

This file implements capture of all math mode material at the outer level, i.e., a formula is captured in its entirety with inner text blocks (possibly containing further math) absorbed as part of the formula. For example,

```
\[ a \in A \text{ for all } a < 5 \]
```

would only result in a single capture of the tokens “`a \in A \text{ for all } a < 5`”.

## 1.1 Code level interfaces

---

```
\math_register_env:n \math_register_env:n {<env>}  
\math_register_env:nn \math_register_env:nn {<env>} {<options>}
```

Registers the  $\langle env \rangle$  as a math environment which should be captured and made available. This is necessary for all top-level math mode environments: low-level errors may result if these are not correct set up. One or more key-value  $\langle options \rangle$  may also be given:

**arg-spec** The argument specification taken by the beginning of the environment; this is used to remove non-mathematical material.

---

```
\math_processor:n \math_processor:n {<tokens>}
```

Declares that the captured math content should be passed to the  $\langle tokens \rangle$ , which will receive the environment type as **#1** and the content as **#2**.

## 1.2 Document level interfaces

---

```
\RegisterMathEnvironment \RegisterMathEnvironment [[<options>] {<env>}
```

Registers the  $\langle env \rangle$  as a math environment which should be captured and made available. This is necessary for all top-level math mode environments: low-level errors may result if these are not correct set up. One or more key-value  $\langle options \rangle$  may also be given:

**arg-spec** The argument specification taken by the beginning of the environment; this is used to remove non-mathematical material.

# 2 Known current bugs, etc.

New Section, now with subsections.  
As indicated, these lists are probably incomplete.  
Some of these have been addressed in a more recent branch.

## 2.1 Capture/grabbing problems

1. Incorrect grabbing of  $\$$ -math when there is also explicit  $\$$ -math within a *text environment* that is itself within the math that should all be grabbed.
2. Similar incorrect grabbing with  $\$\$$  also.
3. The grabbing, for all the display environments (and  $\backslash$   $\backslash$ ), needs to deal with nesting: `amsmath` contains code for this.
- 4.

## 2.2 Other problems

1. The presence of `\m@th` in association with `\ensuremath` does not necessarily indicate fakemath. This is because wanting `mathsurround` to be zero is very reasonable and common, *even when the math is genuine* (and hence needs to be collected).
2. User-defined environments can create problems; but this area, of new, copied and changed environments, has not yet been developed.

Joseph wrote, inter alia:  
My thinking [regarding] `\RegisterMathEnvironment`  
- (New) Math environments should not be created-then-patched, but only generated by a [(future)] dedicated command (`\DeclareMathEnvironment`, presumably)  
- Math environments created with `ltxcmd` [commands] should not be copied, . . .  
- Package authors should be able to manually set up math environments with a public boolean.

- 3.

## 2.3 Other Todos

1. Add (some of) the math display commands that were “lifted from plain”, e.g., `\displaylines` `\eqalign`(??).
- 2.

`\MaybeStop` (temporarily) not executed, as it is unknown on Chris’ system.

## 3 The Implementation

- 1 `<@=math>`
- 2 `<*kernel>`

### 3.1 File declaration

- 3 `\ProvidesFile{latex-lab-mathtagging.ltx}`
- 4 `[\ltxmathtaggingdate\space`
- 5 `\ltxmathtaggingversion\space`
- 6 `Grab all the math(s) and tag it (experiments)]`

Change description here?

Temp loading ...

```

7 \AddToHook{begindocument/before}{\RequirePackage{latex-lab-testphase-block-tagging}}
8 \ExplSyntaxOn

```

### 3.2 Setup

Loading amsmath is an absolute requirement: this avoids needing to have conditional definitions and deals with how to define  $\lceil/\rceil$  neatly.

```

9 \tl_gput_right:Nn \@kernel@before@begindocument
10 { \RequirePackage { amsmath } }

```

### 3.3 Data structures

$\llcorner\_math\_collected\_bool$

Tracks whether math mode material has been collected, which happens inside amsmath environments as well as those handled directly here.

```

11 \bool_new:N \llcorner\_math\_collected\_bool

```

Change first tl name below: 'env' => 'info'?  
 Or do we need an extra storage?

$\llcorner\_math\_grabbed\_env\_tl$   
 $\llcorner\_math\_grabbed\_math\_tl$

```

12 \tl_new:N \llcorner\_math\_grabbed\_env\_tl
13 \tl_new:N \llcorner\_math\_grabbed\_math\_tl

```

### 3.4 Interface commands

$\llcorner\_math\_process:nn$   
 $\llcorner\_math\_process:Vn$   
 $\llcorner\_math\_process\_auxi:nn$   
 $\llcorner\_math\_process\_auxii:nn$

A no-op place-holder; the internal wrapper means that it does not need to be concerned with internals.

```

14 \cs_new_protected:Npn \llcorner\_math\_process:nn #1#2
15 {
16   \legacy_if:nF { measuring@ }
17   {
18     \tl_if_in:nnF {#2} { \m@th }
19     { \tl_trim_spaces_apply:nN {#2} \llcorner\_math\_process\_auxi:nn {#1} }
20   }
21 }
22 \cs_generate_variant:Nn \llcorner\_math\_process:nn { V }
23 \cs_new_protected:Npn \llcorner\_math\_process\_auxi:nn #1#2
24 {
25   \tl_gset:Nn \llcorner\_math\_grabbed\_env\_tl {#2}
26   \tl_gset:Nn \llcorner\_math\_grabbed\_math\_tl {#1}
27   \llcorner\_math\_process\_auxii:nn {#2} {#1}
28 }
29 \cs_new_protected:Npn \llcorner\_math\_process\_auxii:nn #1#2 { }

```

(End of definition for  $\llcorner\_math\_process:nn$ ,  $\llcorner\_math\_process\_auxi:nn$ , and  $\llcorner\_math\_process\_auxii:nn$ .)

$\llcorner\_math\_processor:n$

A simple installer

```

30 \cs_new_protected:Npn \llcorner\_math\_processor:n #1
31 { \cs_set_protected:Npn \llcorner\_math\_process\_auxii:nn ##1##2 {#1} }

```

(End of definition for  $\llcorner\_math\_processor:n$ . This function is documented on page 2.)

### 3.5 Content grabbing

`\_math_grab_dollar:w` Grab up to a single \$, for inline math mode, suppressing any processing if the first token is `\m@th`.

what's that test doing?

It is some kind of fix, to avoid the remote possibility that the math is empty, making the code produce an unwanted `$$`.

cf. the code for this in `\@ensuredmath`

It is harmless but unnecessary in the `dollardollar` grabbing below.

what's that test doing?

```

32 \cs_new_protected:Npn \_math_grab_dollar:w % $
33   #1 $
34   {
35     \tl_if_blank:nF {#1}
36     {
37       \_math_process:nn { math } {#1} % $
38     } % fairly simple this one
39     \tagmccend %end P-chunk, in code: \tag_mc_end_push:
40     \@kernel@math@begin
41     #1 $
42     \@kernel@math@end
43     \tagmcbegin{} % restart P-chunk (whatsits in pdftex)
44   }
45 }

```

*(End of definition for `\_math_grab_dollar:w`.)*

`\_math_grab_dollardollar:w` And for the classical T<sub>E</sub>X display structure.

```

46
47 \skip_new:N \l__math_tmpa_skip
48
49 \cs_new_protected:Npn \_math_grab_dollardollar:w % $$
50   #1 $$
51   {
52     \tl_if_blank:nF {#1}
53     {
54       \_math_tag_dollardollar_display:nn { equation* }{#1}
55       #1
56       $$
57     }
58   }
59
60 \cs_new_protected:Npn \@kernel@close@P {
61   \tagmccend %end P-chunk, in code: \tag_mc_end_push:
62   \int_gincr:N \g__tag_para_end_int
63   \bool_if:NT \l__tag_para_show_bool
64     { \tag_mc_begin:n{artifact}
65       \rlap{\color_select:n{red}\tiny\ \int_use:N\g__tag_para_end_int}
66       \tag_mc_end:
67     }
68   \tag_struct_end:
69 }
70
71
72
73 \cs_new_protected:Npn \_math_tag_dollardollar_display:nn #1#2 {
74   \_math_process:nn {#1} {#2}
75   \@kernel@close@P
76   \@kernel@math@begin

```

```

77 % \skip_set:Nn \belowdisplayskip {-\belowdisplayskip}
78 % \skip_set:Nn \belowdisplayshortskip {-\belowdisplayshortskip}
79 % \int_set:Nn \postdisplaypenalty {10000}
80 %%
81 % \group_insert_after:N \_math_tag_dollardollar_display_end:
82 }
83
84 \cs_new_protected:Npn \_math_tag_dollardollar_display_end: {
85 % \typeout{== tag dollarldollar display end}
86 % \ShowTagging{struct-stack}
87 \tagpdfparaOff
88 \para_raw_end:
89 \tagpdfparaOn
90 \l__math_tmpa_skip \lastskip
91 \@kernel@math@end
92 \penalty \postdisplaypenalty
93 \skip_vertical:n { -\l__math_tmpa_skip }
94 %
95 \@doendpe % this has no \end{...} to take care of it
96 }
97
98

```

*(End of definition for \\_math\_grab\_dollardollar:w.)*

`\_math_grab_inline:w` Collect inline math content and deal with the need to move to math mode.

```

99 \cs_new_protected:Npn \_math_grab_inline:w % \(\
100 #1 \)
101 {
102   \tl_if_blank:nF {#1}
103   {
104     \_math_process:nn { math } {#1}
105     $ #1 $
106   }
107   \bool_set_false:N \l__math_collected_bool
108 }

```

*(End of definition for \\_math\_grab\_inline:w.)*

`\_math_grab_eqn:w` For the most common use of `\[/\]`: turn into an environment.

```

109 \cs_new_protected:Npn \_math_grab_eqn:w % \[
110 #1 \]
111 {
112 % \typeout{collected? = \bool_if:NTF \l__math_collected_bool {true}{false}}
113 \begin { equation* } #1 \end { equation* }
114 }

```

*(End of definition for \\_math\_grab\_eqn:w.)*

### 3.6 Marking math environments

A general mechanism for math mode environments that do not grab their content (*cf.* most `amsmath` environments).

`\l__math_env_name_tl` To allow us to carry out “special effects”

```
115 \tl_new:N \l__math_env_name_tl
```

Here we set up specialised handling of environments. The idea for the `arg-spec` key is that if an environment takes arguments, we don’t worry during the main grabbing. Rather, we remove the arguments from the grabbed content and forward only the payload. That is done by (ab)using `lcmd`.

```
116 \keys_define:nn { __math }
117   {
118     arg-spec .code:n =
119     {
120       \ExpandArgs { c } \DeclareDocumentCommand
121         { __math_env \l__math_env_name_tl _aux: }
122         {#1}
123         { \__math_env_forward:w }
124     }
125 }
```

`\math_register_env:nn` Set up to capture environment content and make available.

`\math_register_env:n`  
`\RegisterMathEnvironment`

```
126 \cs_new_protected:Npn \math_register_env:nn #1#2
127   {
128     \tl_set:Nn \l__math_env_name_tl {#1}
129     \keys_set:nn { __math } {#2}
130     \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
131     \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
132   %
133   \ExpandArgs { nnx } \RenewDocumentEnvironment {#1} { b }
134   {
135     % \bool_set_true:N \exp_not:N \l__math_collected_bool
136     % \cs_if_exist:cTF { __math_env #1 _aux: }
137     % {
138     %   \exp_not:c { __math_env #1 _aux: }
139     %   #####1 \exp_not:N \__math_env_end: {#1}
140     % }
141     % { \exp_not:N \__math_process:nn {#1} {#####1} }
142     \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
143     {
144     % \typeout{===>B1}
145     }
146     {
147     % \typeout{===>B2}
148     % \cs_if_exist:cTF { __math_env #1 _aux: }
149     % {
150     %   \exp_not:c { __math_env #1 _aux: }
151     %   #####1 \exp_not:N \__math_env_end: {#1}
152     % }
153     % { \exp_not:N \__math_process:nn {#1} {#####1} }
154     % \exp_not:n { \@kernel@math@registered@begin }
155     % \bool_set_true:N \exp_not:N \l__math_collected_bool
156     % }
157     % \exp_not:N \tracingall
158     \exp_not:c { __math_env_ #1 _begin: }
```

```

159     #####1
160     \exp_not:c { __math_env_ #1 _end: }
161 %     \exp_not:c { __math_env_ #1 _end: }
162 %     \exp_not:N \tracingnone
163 %     \exp_not:n { \@kernel@math@registered@end }
164 }
165 {
166 }
167 }
168
169 \cs_set_protected:Npn \__cs_tmp:w #1
170 {
171   \group_begin:
172   \exp_args:No \__cs_generate_internal_variant:n
173   { \tl_to_str:n {#1} }
174   \group_end:
175 }
176 \__cs_tmp:w { nxxx }
177
178
179 \cs_new_protected:Npn \math_register_halign_env:nn #1#2
180 {
181   \tl_set:Nn \l__math_env_name_tl {#1}
182   \keys_set:nn { __math } {#2}
183   \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
184   \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
185 %
186   \ExpandArgs { nxxx } \RenewDocumentEnvironment {#1} { b }
187   {
188 %     \bool_set_true:N \exp_not:N \l__math_collected_bool
189 %     \cs_if_exist:cTF { __math_env #1 _aux: }
190 %     {
191 %       \exp_not:c { __math_env #1 _aux: }
192 %       #####1 \exp_not:N \__math_env_end: {#1}
193 %     }
194 %     { \exp_not:N \__math_process:nn {#1} {#####1} }
195   \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
196   {
197 %     \typeout{===>B1}
198   }
199   {
200 %     \typeout{===>B2}
201     \cs_if_exist:cTF { __math_env #1 _aux: }
202     {
203       \exp_not:c { __math_env #1 _aux: }
204       #####1 \exp_not:N \__math_env_end: {#1}
205     }
206     { \exp_not:N \__math_process:nn {#1} {#####1} }
207     \exp_not:n { \@kernel@math@registered@begin }
208     \bool_set_true:N \exp_not:N \l__math_collected_bool
209   }
210 %   \exp_not:N \tracingall
211   \exp_not:c { __math_env_ #1 _begin: }
212   #####1

```



```

213 %       \exp_not:c { __math_env_ #1 _end: }
214 %       \exp_not:N \tracingnone
215     }
216     {
217       \exp_not:c { __math_env_ #1 _end: }
218     }
219   }
220
221 \cs_new_protected:Npn \math_register_odd_env:nn #1#2
222   {
223     \tl_set:Nn \l__math_env_name_tl {#1}
224     \keys_set:nn { __math } {#2}
225     \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
226     \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
227   %
228   \ExpandArgs { nxxx } \RenewDocumentEnvironment {#1} { b }
229     {
230       \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
231         {
232           %       \typeout{==>B1}
233         }
234         {
235           %       \typeout{==>B2}
236           \cs_if_exist:cTF { __math_env #1 _aux: }
237             {
238               \exp_not:c { __math_env #1 _aux: }
239               #####1 \exp_not:N \__math_env_end: {#1}
240             }
241             { \exp_not:N \__math_process:nn {#1} {#####1} }
242           \exp_not:n { \@kernel@math@registered@begin }
243           \bool_set_true:N \exp_not:N \l__math_collected_bool
244         }
245     %       \exp_not:N \tracingall
246     \exp_not:c { __math_env_ #1 _begin: }
247     #####1
248   }
249   {
250     \exp_not:c { __math_env_ #1 _end: }
251 % needed if we don't have $$...$$
252 %       \exp_not:n { \typeout{--> @kernel@math@registered@end }}
253     \exp_not:n { \@kernel@math@registered@end }
254   }
255 }
256
257
258 % FMi: compare with block change!
259 %
260 % \DeclareRobustCommand*\begin[1]{%
261 % \UseHook{env/#1/before}%
262 % \@ifundefined{#1}%
263 %   {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
264 %   {\def\reserved@a{\def\@currenvir{#1}%
265 %     \edef\@currencline{\on@line}%
266 %     \@execute@begin@hook{#1}%

```

```

267 % \csname #1\endcsname}}%
268 % \@ignorefalse
269 % \begingroup
270 % \@endpefalse % tmp!!! is it ok to drop this here?
271 % \reserved@a}
272
273
274 \cs_new:Npn \@kernel@math@registered@begin {
275 % \ShowTagging{struct-stack}
276 % \typeout{==>A1}\ShowTagging{struct-stack,mc-current}
277 \mode_if_vertical:TF
278 {
279 % \legacy_if:nTF { @endpe }
280 % { \legacy_if_set_false:n { @endpe } }
281 % { \__block_list_beginpar_vmode: }
282 %
283 % \typeout{==>~ at:~ \g__tag_struct_tag_tl}
284 %
285 \exp_args:Noo\str_if_eq:nnF \g__tag_struct_tag_tl { \l__tag_para_main_tag_tl } %
286 {
287 % \typeout{==>A2}
288 \__block_beginpar_vmode:
289 } % needs correction!
290 }
291 {
292 % \typeout{==>A3}
293 \@kernel@close@P
294 % \tagmcent % needs correction!
295 }
296 \@kernel@math@begin
297 \tagpdfparaOff
298 % \typeout{==>MC1}\ShowTagging{mc-current}
299 }
300
301 \cs_new:Npn \@kernel@math@registered@end {
302 % \typeout{==>MC2}\ShowTagging{mc-current}
303 \para_raw_end:
304 \tagpdfparaOn
305 \@kernel@math@end
306 % \typeout{==>MC3}\ShowTagging{mc-current}
307 \@endpetrue
308 }
309
310 \cs_new_protected:Npn \math_register_env:n #1
311 { \math_register_env:nn {#1} { } }
312 \NewDocumentCommand \RegisterMathEnvironment { 0{ } m }
313 { \math_register_env:nn {#2} {#1} }

(End of definition for \math_register_env:nn, \math_register_env:n, and \RegisterMathEnvironment.
These functions are documented on page 2.)

\__math_env_forward:w
314 \cs_new_protected:Npn \__math_env_forward:w #1 \__math_env_end: #2
315 { \__math_process:nn {#2} {#1} }

```

(End of definition for `\_math_env_forward:w`.)

### 3.7 Document commands

Add one more here: `displaymath`, which is equivalent to `\[ , \]` and hence to the basic `equation*`. Added in more recent branch.

`\equation` These environments are not set up by `amsmath` to collect their body, so we do that here.

`\_math_equation_begin:` This has to be done *after* we can be sure `amsmath` is loaded.

`\equation*` Note that with `amsmath` loaded, `equation*` and `equation`

`\_math_equation_star_begin:` are the two basics: they are used to define the other single-row display environments, etc.

`\endequation`

`\_math_equation_end:`

`\endequation*`

`\_math_equation_star_end:`

```
316 \tl_gput_right:Nn \@kernel@before@begindocument
317   {
318     \math_register_env:n { equation }
319     \math_register_env:n { equation* }
320 % at the moment register_env can only do display math
321 %   \math_register_env:n { math }
322   \RenewDocumentEnvironment{math} {b}{\$#1$}{}
323 % and this one doesn't work either
324 %   \math_register_env:n { displaymath }
325   \RenewDocumentEnvironment{displaymath} {b}{\[#1\]}{}
326 }
```

(End of definition for `\equation` and others. These functions are documented on page ??.)

`\(` If math mode has not been collected, we need to do that; otherwise, worry about whether  
`\)` we are in math mode or not. The closing command here can only occur inside a collected math block: otherwise it will be simply used as a delimiter.

```
327 \cs_gset_protected:Npn \(\ % \)
328   {
329     \bool_if:NTF \l__math_collected_bool
330       {
331         \mode_if_math:TF
332           { \@badmath }
333           { $ }
334       }
335     {
336       \bool_set_true:N \l__math_collected_bool
337       \__math_grab_inline:w
338     }
339   } % \)
340 \cs_gset_protected:Npn \)
341   {
342     \mode_if_math:TF
343       { $ }
344       { \@badmath }
345   }
```

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

\[ Again, we need to watch for when `amsmath` is loaded after this code. The flag usage here is to cover the case where `\[/\]` is hidden inside another environment. In this case the grabbing happens on the outer level and should not be repeated.

```

346 \tl_gput_right:Nn \@kernel@before@begindocument
347 {
348   \cs_gset_protected:Npn \[ % \]
349   {
350     \bool_if:NTF \l__math_collected_bool
351     { \begin { equation* } }
352     { \__math_grab_eqn:w }
353   } % \[
354   \cs_gset_protected:Npn \]
355   {
356     \bool_if:NTF \l__math_collected_bool
357     { \end{ equation* } }
358     { \@badmath }
359   }
360 }

```

*(End of definition for `\[` and `\]`. These functions are documented on page ??.)*

why does `ensuremath` need handling at all?

Indeed! Currently, this is setup to process the math that it has anyways already captured as its argument; thus it is more efficient than leaving the capture to be repeated by the `\everymath`

A bit of nesting fun to make sure we collect only if required.

```

361 %\cs_gset_protected:Npn \ensuremath #1
362 % {
363 %   \mode_if_math:TF
364 %     {#1}
365 %     {
366 %       \bool_if:NTF \l__math_collected_bool
367 %       { \@ensuredmath {#1} }
368 %       {
369 %         \bool_set_true:N \l__math_collected_bool
370 %         \__math_process:nn { math } {#1}
371 %         \@ensuredmath {#1}
372 %         \bool_set_false:N \l__math_collected_bool
373 %       }
374 %     }
375 % }

```

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

### 3.8 `\everymath` and `\everydisplay`

The business end for grabbing inline math and “raw” T<sub>E</sub>X display. Most display math mode is actually handled elsewhere, as we have macro control.

```

376
377 \tl_new:N\tmpmathcontent
378
379
380 \def\@kernel@math@begin {
381 % \typeout{=>-math-begin}
382 % needs different handling if we support nesting
383 \tl_gset:Nx\tmpmathcontent
384 {

```

```

385 LaTeX~ formula~ starts~
386 \exp_not:N\begin{\g__math_grabbed_env_tl}
387 \space
388 \exp_not:V\g__math_grabbed_math_tl
389 \space
390 \exp_not:N\end{\g__math_grabbed_env_tl}
391 \space LaTeX~ formula~ ends~
392 }
393 \tagstructbegin{tag=Formula,
394 AFinline-o=\tmpmathcontent,
395 title-o=\g__math_grabbed_env_tl,
396 actualtext=\tmpmathcontent
397 % alt=\tmpmathcontent
398 }
399 % inner formula if multiple parts (not really implemented yet)
400 \grabaformulapartandstart
401 % the above does:
402 % \tagstructbegin{tag=Formula}\tagmcbegin{}
403 % or just
404 % \tagmcbegin{}
405 }
406 \def\@kernel@math@end {
407 % \typeout{==>~math~end}
408 % \ShowTagging{struct-stack}
409 \tagmcbend
410 \if@subformulas
411 \tagstructend
412 \else
413 \fi
414 \tagstructend
415 % \ShowTagging{struct-stack}
416 }
417
418 \exp_args:No \tex_everymath:D
419 {
420 \tex_the:D \tex_everymath:D
421 \bool_if:NF \l__math_collected_bool
422 {
423 \bool_set_true:N \l__math_collected_bool
424 \__math_grab_dollar:w
425 }
426 }
427 \exp_args:No \tex_everydisplay:D
428 {
429 \tex_the:D \tex_everydisplay:D
430 \iftrue % this may have to be a settable flag!
431 % {
432 % \typeout{==>~ in~ everydisplay}
433 \skip_set:Nn \belowdisplayskip {-\belowdisplayskip}
434 \skip_set:Nn \belowdisplayshortskip {-\belowdisplayshortskip}
435 \int_set:Nn \postdisplaypenalty {10000}
436 %
437 \group_insert_after:N \__math_tag_dollardollar_display_end:
438 % }

```

```

439 \fi
440 \bool_if:NF \l__math_collected_bool
441 {
442   \bool_set_true:N \l__math_collected_bool
443   \__math_grab_dollardollar:w
444 }
445 }

```

### 3.9 Modifying kernel environments

We need to cover this even though it is, of course, not encouraged.

```

446 \math_register_env:n { eqnarray }
447 \math_register_env:n { eqnarray* }
      Places where math mode is (ab)used.
448 \clist_map_inline:nn
449   { tabular }
450   {
451     \AddToHook{ env / #1 / begin }
452     { \bool_set_true:N \l__math_collected_bool }
453   }

```

`\__math_m@th:` Handle non-math use of math mode. At present nesting isn't supported as `\m@th` pops up in a few places that *are* math mode!

```

454 \cs_new_eq:NN \__math_m@th: \m@th
455 \cs_gset_protected:Npn \m@th
456   {
457     \bool_set_true:N \l__math_collected_bool
458     \__math_m@th:
459   }

```

(End of definition for `\__math_m@th:` and `\m@th`. This function is documented on page ??.)

### 3.10 Modifying amsmath

`\__math_amsmath_align@:nn` Mark up all of the display environments as the content is captured anyway. We then use  
`\__math_amsmath_gather@:n` an internal macro in each environment type to insert the processing code. Each of these  
`\__math_amsmath_multline@:n` is slightly different, so we cannot use a simple loop here. The test for `\split@tag` is  
`\align@` required as the `split` environment internally uses `gather` when not within an *amsmath*  
`\gather@` environment, for example inside `equation`. Without the precaution, we'd get two copies  
`\multline@` of the grabbed math, the second of which would start with `\split@tag`.

```

460
461
462
463 \tl_gput_right:Nn \@kernel@before@begindocument {
464   %
465   \renewenvironment{gather*}{%
466     \start@gather\st@rredtrue
467   }
468   {%
469     % this redirection doesn't work if we alter "gather"!
470     % \endgather
471     % so replace it with its real meaning

```

```

472 \math@cr \black@totwidth@ \egroup
473 $$\ignorespacesafterend
474 }
475 \def\common@align@ending {
476 \math@cr \black@totwidth@
477 \egroup
478 \ifingather@
479 \restorealignstate@
480 \egroup
481 \nonumber
482 \ifnum0='{\fi\iffalse}\fi
483 \else
484 $$%
485 \fi
486 \ignorespacesafterend
487 }
488 \renewenvironment{alignat}{%
489 \start@align\z@\st@rredfalse
490 }{%
491 \common@align@ending
492 }
493 \renewenvironment{alignat*}{%
494 \start@align\z@\st@rredtrue
495 }{%
496 \common@align@ending
497 }
498 \renewenvironment{xalignat}{%
499 \start@align\@ne\st@rredfalse
500 }{%
501 \common@align@ending
502 }
503 \renewenvironment{xalignat*}{%
504 \start@align\@ne\st@rredtrue
505 }{%
506 \common@align@ending
507 }
508 \renewenvironment{xxalignat}{%
509 \start@align\tw@\st@rredtrue
510 }{%
511 \common@align@ending
512 }
513 \renewenvironment{align}{%
514 \start@align\@ne\st@rredfalse\m@ne
515 }{%
516 \common@align@ending
517 }
518 \renewenvironment{align*}{%
519 \start@align\@ne\st@rredtrue\m@ne
520 }{%
521 \common@align@ending
522 }
523 \renewenvironment{flalign}{%
524 \start@align\tw@\st@rredfalse\m@ne
525 }{%

```

```

526 \common@align@ending
527 }
528 \renewenvironment{flalign*}{%
529 \start@align\tw@\st@rredtrue\m@ne
530 }{%
531 \common@align@ending
532 }
533 %
534 \renewenvironment{multline*}{\start@multline\st@rredtrue}
535 {%
536 \iftagsleft@ \@xp\lendmultline@ \else \@xp\rendmultline@ \fi
537 \ignorespacesafterend
538 }

```

Also for false?

```

539 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_stop:}
540 %
541 \math_register_halign_env:nn {align}{}
542 \math_register_halign_env:nn {align*}{}
543 \math_register_halign_env:nn {flalign}{}
544 \math_register_halign_env:nn {flalign*}{}
545 \math_register_halign_env:nn {gather}{}
546 \math_register_halign_env:nn {gather*}{}
547 \math_register_halign_env:nn {multline}{}
548 \math_register_halign_env:nn {multline*}{}
549 \math_register_halign_env:nn {xalignat}{}
550 \math_register_halign_env:nn {xalignat*}{}
551 \math_register_halign_env:nn {xxalignat}{}
552 %
553 \@namedef{maketag @ @ @} #1{%
554 % \typeout{--->maketag @ @ @}
555 \ifmeasuring@
556 \hbox{\m@th\normalfont#1}%
557 \else
558 \tagmccend \tagstructbegin{tag=Lbl}%
559 \tagmccbegin{tag=Lbl}%
560 \hbox{\m@th\normalfont#1}%
561 \tagmccend \tagstructend \tagmccbegin{}%
562 \fi
563 }
564 \def\intertext@{%
565 \def\intertext##1{%
566 \ifvmode\else\\\@empty\fi
567 \noalign{%
568 % we have to flip the sign on the skip because we flipped it on the outside
569 \penalty\postdisplaypenalty\vskip-\belowdisplayskip
570 \vbox{

```

Stop tagging when measuring:

```

571 \ifmeasuring@\tag_stop:\fi
572 \normalbaselines
573 \ifdim\linewidth=\columnwidth
574 \else \parshape\@ne \@totalleftmargin \linewidth
575 \fi

```



if we use 2 levels of formulas this would need changing

not true any longer

End the previous mc:

```
576         \tag_mc_end_push:
```

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

```
577         \tagpdfsetup{paratag=P}%
578         \tagpdfpara0n
579         \noindent\ignorespaces##1\par
```

Restart the MC

```
580         \tag_mc_begin_pop:n{}}%
581         \penalty\predisplaypenalty\vskip\abovedisplayskip%
582     }%
583 }
584 }
```

```
585 \@namedef{math@cr @ @ @ gather}{%
586     \ifst@rred\nonumber\fi
587     &\relax
588     \make@display@tag
589 %
590     \maybestartnewformulatag
591 %
592     \ifst@rred\else\global\@eqnswtrue\fi
593     \global\advance\row@\@ne
594     \cr
595 }
```

```
596 \@namedef{math@cr @ @ @ align}{%
597     \ifst@rred\nonumber\fi
598     \if@eqnsw \global\tag@true \fi
599     \global\advance\row@\@ne
600     \add@amps\maxfields@
601     \omit
602     \kern-\alignsep@
603     \iftag@
604         \setboxz@h{\@lign\strut@{\make@display@tag}}%
605         \place@tag
606     \fi
607 %
608     \maybestartnewformulatag
609 %
610     \ifst@rred\else\global\@eqnswtrue\fi
611     \global\lineht@\z@
612     \cr
613 }
```

```
614 \def\restore@math@cr{\@namedef{math@cr @ @ @}{
615 %
616     \maybestartnewformulatag
617 %
618     \cr}}
619 \restore@math@cr
620 }
```

(End of definition for `\_math_amsmath_align@:nn` and others. These functions are documented on page ??.)

```

621 \cs_new:Npn \__math_split_at_nl_first:w #1 \ \ #2 \ \ #3 \s_stop
622 {
623   \quark_if_nil:nTF {#2}
624     { {#1} { } }
625     {
626       \__math_split_chk_if_begin:ww #1 \begin \q_nil \s_mark
627         #2 \ \ #3 \s_stop
628     }
629 }
630 \cs_new:Npn \__math_split_chk_if_begin:ww #1 \begin #2 #3 \s_mark
631   #4 \ \ \q_nil \ \ \s_stop
632 {
633   \quark_if_nil:nTF {#2}
634     { {#1} {#4} }
635     {
636       \exp_after:wN \__math_split_collect_one_end:w
637       \__math_split_cleanup_begin_q_nil:w #1 \begin{#2} #3 \ \ #4 \s_stop
638       { } { 1 }
639     }
640 }
641 \cs_new:Npn \__math_split_cleanup_begin_q_nil:w #1 \begin \q_nil {#1}
642 \cs_new:Npn \__math_split_collect_one_end:w #1 \end #2 #3 \s_stop #4 #5
643 {
644   \exp_args:Nf \__math_split_check_count_begins:n
645   { \__math_split_count_begins:n { #4 #1 } } {#5}
646   { #4 #1 \end{#2} } {#3}
647 }
648 \cs_new:Npn \__math_split_count_begins:n #1
649 { \int_eval:n { 0 \__math_split_count_begins:w #1 \begin \q_nil } }
650 \cs_new:Npn \__math_split_count_begins:w #1 \begin #2
651 { \quark_if_nil:nF {#2} { +1 \__math_split_count_begins:w } }
652 \cs_new:Npn \__math_split_check_count_begins:n #1 #2 #3 #4
653 {
654   \int_compare:nNnTF {#1} = {#2}
655   {
656     \exp_last_unbraced:Nf \__math_split_final_cleanup:nn
657     { \split:n { \__math_split_guard:n {#3} #4 } }
658   }
659   {
660     \exp_args:No \use_ii_i:nn
661     { \exp_after:wN { \int_value:w \int_eval:n { #2 + 1 } } }
662     { \__math_split_collect_one_end:w #4 \s_stop {#3} }
663   }
664 }
665 \cs_new:Npn \__math_split_final_cleanup:nn #1 #2
666 {
667   \exp:w \__math_split_final_cleanup:w #1
668   \__math_split_guard:n \q_nil \s_mark { }
669   {#2}
670 }
671 \cs_new:Npn \__math_split_final_cleanup:w #1 \__math_split_guard:n #2 #3 \s_mark #4
672 {
673   \quark_if_nil:nTF {#2}

```

```

674     { \exp_end: { #4 #1 } }
675     { \_math_split_final_cleanup:w #3 \s_mark { #4 #1 #2 } }
676   }
677 \NewDocumentCommand \splitnl { mm +m }
678 {
679   \tl_set:Nf \l_tmpa_tl { \split:n {#3} }
680   \show \l_tmpa_tl
681   \exp_after:wN \_splitnl_aux:nnNN \l_tmpa_tl #1 #2
682 }
683
684
685 \cs_new:Npn \split:n #1 {
686   \_math_split_at_nl_first:w #1 \ \q_nil \ \s_stop }
687
688 \cs_new:Npn \_math_split_at_nl:NN #1#2 {
689   \tl_set:Nf \l_tmpa_tl {
690     \exp_after:wN \_math_split_at_nl_first:w #1 \ \q_nil \ \s_stop }
691   \exp_after:wN \_math_split_at_nl_aux:nnNN \l_tmpa_tl #1 #2
692 }
693
694 \cs_new_protected:Npn \_math_split_at_nl_aux:nnNN #1 #2 #3 #4
695 {
696   \tl_gset:Nn #4 {#1}
697   \tl_gset:Nn #3 {#2}
698 }
699

```

(End of definition for .)

\maybestartnewformulatag

```

700
701 \newif\if@subformulas
702 \tl_new:N \result
703
704 \cs_new_protected:Npn\grabaformulapartandstart {
705   \_math_split_at_nl:NN \g__math_grabbed_math_tl \result
706   \typeout{====>first-result=\meaning\result}
707   \typeout{====>first-tmpmathcontent=\meaning\g__math_grabbed_math_tl}
708   \tl_if_empty:NTF \g__math_grabbed_math_tl
709     {
710       \typeout{====>formula~ has~ no~ subparts}
711       \global\@subformulasfalse
712     }
713     {
714       \typeout{====>formula~ has~ subparts}
715       \global\@subformulastrue
716       \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
717       \tagstructbegin{tag=Formula,

```

For now we don't put anything in /alt or /ActualText on subformulas

```

718 %       alt=\result,
719       title-o=\resulttitle
720     }
721   }
722   \tagmcbegin{}

```

```

723 }
724
725 \cs_new_protected:Npn\grabaformulapartandmaybe restart {
726   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
727   \typeout{====>result=\meaning\result}
728   \typeout{====>tmpmathcontent=\meaning\g__math_grabbed_math_tl}
729   % \tl_if_empty:NTF \g__math_grabbed_math_tl
730   % {
731   %   \typeout{====>tmpmathcontent=empty}
732   % }
733   % {
734   %   \typeout{====>tmpmathcontent=not-empty}
735   %   \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
736   %   \tagstructbegin{tag=Formula,
737   %     alt=\result,
738   %     title-o=\resulttitle
739   %   }
740   % }
741   \tagmcbegin{}
742 }

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

743 \def\maybestartnewformulatag {
744   \if@subformulas
745   \ifmeasuring@\else
746   %
747   \tl_if_empty:NF \g__math_grabbed_math_tl
748   {
749     \tagmcbegin{}
750     \tagstructend
751     \grabaformulapartandmaybe restart
752   }
753   \fi
754   \fi
755 }

The breqn packages changes catcodes and that isn't yet covered by our mechanism.

756 %\AddToHook{package/breqn/after}{
757 % \typeout{====>~ in~ hook}
758 % \math_register_halign_env:nn {dmath}{}
759 % \math_register_halign_env:nn {dgroup*}{}
760 %}

761 \ExplSyntaxOff
762 <@@=
763 %
764 </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	
\(	99
\(	<u>327</u>
\)	100
\)	<u>327</u>
\[	109, 325
\[	<u>4, 6, 11, 12, 346</u>
\]	566, 621, 627, 631, 637, 686, 690
\_	65
\]	110, 325
\]	<u>4, 6, 11, 12, 346</u>
<b>A</b>	
\abovedisplayskip	581
\AddToHook	7, 451, 756
\advance	593, 599
<b>B</b>	
\begin	113, 260, 351, 386, 626, 630, 637, 641, 649, 650
\begingroup	269
\belowdisplayskip	78, 434
\belowdisplayskip	77, 433, 569
block internal commands:	
\_block_beginpar_vmode:	288
\_block_list_beginpar_vmode:	281
bool commands:	
\bool_if:NTF	63, 112, 142, 195, 230, 329, 350, 356, 366, 421, 440
\bool_new:N	11
\bool_set_false:N	107, 372
\bool_set_true:N	135, 155, 188, 208, 243, 336, 369, 423, 442, 452, 457
bool internal commands:	
\l_math_collected_bool	4, 11, 107, 112, 135, 142, 155, 188, 195, 208, 230, 243, 329, 336, 350, 356, 366, 369, 372, 421, 423, 440, 442, 452, 457
<b>C</b>	
clist commands:	
\clist_map_inline:nn	448
color commands:	
\color_select:n	65
\columnwidth	573
\cr	594, 612, 618
cs commands:	
\cs_generate_variant:Nn	22
\cs_gset_eq:NN	130, 131, 183, 184, 225, 226
\cs_gset_protected:Npn	327, 340, 348, 354, 361, 455
\cs_if_exist:NTF	136, 148, 189, 201, 236
\cs_new:Npn	274, 301, 621, 630, 641, 642, 648, 650, 652, 665, 671, 685, 688
\cs_new_eq:NN	454
\cs_new_protected:Npn	14, 23, 29, 30, 32, 49, 60, 73, 84, 99, 109, 126, 179, 221, 310, 314, 694, 704, 725
\cs_set_protected:Npn	31, 169
cs internal commands:	
\_cs_generate_internal_variant:n	172
\_cs_tmp:w	169, 176
\csname	267
<b>D</b>	
\DeclareDocumentCommand	120
\DeclareMathEnvironment	3
\DeclareRobustCommand	260
\def	263, 264, 380, 406, 475, 539, 564, 565, 614, 743
\displaylines	3
<b>E</b>	
\edef	265, 716, 735
\egroup	472, 477, 480
\else	412, 483, 536, 557, 566, 574, 592, 610, 745
\end	95, 113, 357, 390, 642, 646
\endcsname	267
\endequation	<u>316</u>
\endequation*	<u>316</u>
\endgather	470
\ensuremath	3, <u>361</u>
\eqalign	3
\equation	<u>316</u>
\equation*	<u>316</u>
\everydisplay	12
\everymath	12
exp commands:	
\exp:w	667
\exp_after:wN	636, 661, 681, 690, 691
\exp_args:Nf	644
\exp_args:No	172, 418, 427, 660
\exp_args:Noo	285

<code>\exp_end:</code> .....	674				<b>L</b>
<code>\exp_last_unbraced:Nf</code> .....	656	<code>\lastskip</code> .....	90		
<code>\exp_not:N</code> .....	135, 138,	legacy commands:			
	139, 141, 142, 150, 151, 153, 155,	<code>\legacy_if:nTF</code> .....	16, 279		
	157, 158, 160, 161, 162, 188, 191,	<code>\legacy_if_set_false:n</code> .....	280		
	192, 194, 195, 203, 204, 206, 208,	<code>\let</code> .....	539		
	210, 211, 213, 214, 217, 230, 238,	<code>\linewidth</code> .....	573, 574		
	239, 241, 243, 245, 246, 250, 386, 390	<code>\ltmathtaggingdate</code> .....	4		
<code>\exp_not:n</code> .....		<code>\ltmathtaggingversion</code> .....	5		
	..... 154, 163, 207, 242, 252, 253, 388				<b>M</b>
<code>\ExpandArgs</code> .....	120, 133, 186, 228	math commands:			
<code>\ExplSyntaxOff</code> .....	761	<code>\math_processor:n</code> .....	2, 30, 30		
<code>\ExplSyntaxOn</code> .....	8	<code>\math_register_env:n</code> .....	2,		
			126, 310, 318, 319, 324, 446, 447		
	<b>F</b>	<code>\math_register_env:nn</code> .....			
<code>\fi</code> .....	413, 439,	..... 2, 126, 126, 311, 313			
	482, 485, 536, 562, 566, 571, 575,	<code>\math_register_halign_env:nn</code> ...			
	586, 592, 597, 598, 606, 610, 753, 754	..... 179, 541, 542, 543, 544, 545,			
		546, 547, 548, 549, 550, 551, 758, 759			
	<b>G</b>	<code>\math_register_odd_env:nn</code> .....	221		
<code>\global</code> 592, 593, 598, 599, 610, 611, 711, 715		math internal commands:			
<code>\grabaformulapartandmaybe restart</code> ...	..... 725, 751	<code>\__math_amsmath_align@:nn</code> .....	460		
<code>\grabaformulapartandstart</code> .....	400, 704	<code>\__math_amsmath_gather@:n</code> .....	460		
group commands:		<code>\__math_amsmath_multline@:n</code> ...	460		
<code>\group_begin:</code> .....	171	<code>\__math_env_end:</code> .....			
<code>\group_end:</code> .....	174	..... 139, 151, 192, 204, 239, 314			
<code>\group_insert_after:N</code> .....	81, 437	<code>\__math_env_forward:w</code> .	123, 314, 314		
		<code>\__math_equation_begin:</code> .....	316		
	<b>H</b>	<code>\__math_equation_end:</code> .....	316		
<code>\hbox</code> .....	556, 560	<code>\__math_equation_star_begin:</code> .	316		
		<code>\__math_equation_star_end:</code> ....	316		
	<b>I</b>	<code>\__math_grab_dollar:w</code> ...	32, 32, 424		
<code>\ifdim</code> .....	573	<code>\__math_grab_dollardollar:w</code> .....			
<code>\iffalse</code> .....	482	..... 46, 49, 443			
<code>\ifnum</code> .....	482	<code>\__math_grab_eqn:w</code> ....	109, 109, 352		
<code>\iftrue</code> .....	430, 539	<code>\__math_grab_inline:w</code> ...	99, 99, 337		
<code>\ifvmode</code> .....	566	<code>\__math_m@th:</code> .....	454, 454, 458		
<code>\ignorespaces</code> .....	579	<code>\__math_process:nn</code> 14, 14, 22, 37, 74,			
<code>\ignorespacesafterend</code> ....	473, 486, 537	104, 141, 153, 194, 206, 241, 315, 370			
int commands:		<code>\__math_process_auxi:nn</code> ...	14, 19, 23		
<code>\int_compare:nNnTF</code> .....	654	<code>\__math_process_auxii:nn</code> 14, 27, 29, 31			
<code>\int_eval:n</code> .....	649, 661	<code>\__math_split_at_nl:NN</code> .	688, 705, 726		
<code>\int_gincr:N</code> .....	62	<code>\__math_split_at_nl_aux:nnNN</code>	691, 694		
<code>\int_set:Nn</code> .....	79, 435	<code>\__math_split_at_nl_first:w</code> .....			
<code>\int_use:N</code> .....	65	..... 621, 686, 690			
<code>\int_value:w</code> .....	661	<code>\__math_split_check_count_-</code>			
<code>\intertext</code> .....	565	<code>begins:nnnn</code> .....	644, 652		
		<code>\__math_split_chk_if_begin:ww</code> ...			
	<b>K</b>	..... 626, 630			
<code>\kern</code> .....	602	<code>\__math_split_cleanup_begin_q_-</code>			
keys commands:		<code>nil:w</code> .....	637, 641		
<code>\keys_define:nn</code> .....	116	<code>\__math_split_collect_one_end:w</code> .			
<code>\keys_set:nn</code> .....	129, 182, 224	..... 636, 642, 662			

<code>\_math\_split\_count\_begins:n</code>	645, 648	<code>\result</code>	702, 705, 706, 718, 726, 727, 737
<code>\_math\_split\_count\_begins:w</code>	...	<code>\resulttitle</code>	716, 719, 735, 738
	649, 650, 651	<code>\rlap</code>	65
<code>\_math\_split\_final\_cleanup:nn</code>	...	<b>S</b>	
	656, 665	scan commands:	
<code>\_math\_split\_final\_cleanup:w</code>	...	<code>\s\_mark</code>	626, 630, 668, 671, 675
	667, 671, 675	<code>\s\_stop</code>	621, 627, 631, 637, 642, 662, 686, 690
<code>\_math\_split\_guard:n</code>	657, 668, 671	<code>\show</code>	680
<code>\_math\_tag\_dollardollar\_</code>		<code>\ShowTagging</code>	86, 275, 276, 298, 302, 306, 408, 415
<code>display:nn</code>	54, 73	skip commands:	
<code>\_math\_tag\_dollardollar\_</code>		<code>\skip\_new:N</code>	47
<code>display\_end:</code>	81, 84, 437	<code>\skip\_set:Nn</code>	77, 78, 433, 434
<code>\maybe\_start\_new\_formulatag</code>	...	<code>\skip\_vertical:n</code>	93
	590, 608, 616, 700, 743	skip internal commands:	
<code>\MaybeStop</code>	3	<code>\l\_math\_tmpa\_skip</code>	47, 90, 93
<code>\meaning</code>	706, 707, 727, 728	<code>\space</code>	4, 5, 387, 389, 391, 716, 735
mode commands:		split commands:	
<code>\mode\_if\_math:TF</code>	331, 342, 363	<code>\split:n</code>	657, 679, 685
<code>\mode\_if\_vertical:TF</code>	277	<code>\splitnl</code>	677
<b>N</b>		splitnl internal commands:	
<code>\NewDocumentCommand</code>	312, 677	<code>\_splitnl\_aux:nnNN</code>	681
<code>\newif</code>	701	str commands:	
<code>\noalign</code>	567	<code>\str\_if\_eq:nnTF</code>	285
<code>\noindent</code>	579	<b>T</b>	
<code>\nonumber</code>	481, 586, 597	tag commands:	
<code>\normalbaselines</code>	572	<code>\tag\_mc\_begin:n</code>	64
<code>\normalfont</code>	556, 560	<code>\tag\_mc\_begin\_pop:n</code>	580
<b>O</b>		<code>\tag\_mc\_end:</code>	66
<code>\omit</code>	601	<code>\tag\_mc\_end\_push:</code>	39, 61, 576
<b>P</b>		<code>\tag\_stop:</code>	539, 571
<code>\par</code>	579	<code>\tag\_struct\_end:</code>	68
para commands:		tag internal commands:	
<code>\para\_raw\_end:</code>	88, 303	<code>\g\_tag\_para\_end\_int</code>	62, 65
<code>\parshape</code>	574	<code>\l\_tag\_para\_main\_tag\_tl</code>	285
<code>\penalty</code>	92, 569, 581	<code>\l\_tag\_para\_show\_bool</code>	63
<code>\postdisplaypenalty</code>	79, 92, 435, 569	<code>\g\_tag\_struct\_tag\_tl</code>	283, 285
<code>\predisplaypenalty</code>	581	<code>\tag\_mc\_begin</code>	43, 402, 404, 559, 561, 722, 741
<code>\ProvidesFile</code>	3	<code>\tag\_mc\_end</code>	39, 61, 294, 409, 558, 561, 749
<b>Q</b>		<code>\tag\_pdf\_para\_off</code>	87, 297
quark commands:		<code>\tag\_pdf\_para\_on</code>	89, 304, 578
<code>\q\_nil</code>	626, 631, 641, 649, 668, 686, 690	<code>\tag\_pdf\_setup</code>	577
<code>\quark\_if\_nil:nTF</code>	623, 633, 651, 673	<code>\tag\_struct\_begin</code>	393, 402, 558, 717, 736
<b>R</b>		<code>\tag\_struct\_end</code>	411, 414, 561, 750
<code>\RegisterMathEnvironment</code>	2, 3, 126	$\TeX$ and $\LaTeX 2_{\epsilon}$ commands:	
<code>\relax</code>	587	<code>\@badmath</code>	332, 344, 358
<code>\RenewDocumentEnvironment</code>	...	<code>\@currenenv</code>	264
	133, 186, 228, 322, 325	<code>\@currencline</code>	265
<code>\renewenvironment</code>	465, 488, 493,	<code>\@doendpe</code>	95
	498, 503, 508, 513, 518, 523, 528, 534	<code>\@eha</code>	263
<code>\RequirePackage</code>	7, 10	<code>\@empty</code>	566

<code>\@endpefalse</code>	270
<code>\@endpetrue</code>	307
<code>\@ensuredmath</code>	5, 367, 371
<code>\@eqnswtrue</code>	592, 610
<code>\@execute@begin@hook</code>	266
<code>\@ifundefined</code>	262
<code>\@ignorefalse</code>	268
<code>\@kernel@before@begindocument</code>	9, 316, 346, 463
<code>\@kernel@close@P</code>	60, 75, 293
<code>\@kernel@math@begin</code>	40, 76, 296, 380
<code>\@kernel@math@end</code>	42, 91, 305, 406
<code>\@kernel@math@registered@begin</code>	154, 207, 242, 274
<code>\@kernel@math@registered@end</code>	163, 253, 301
<code>\@latex@error</code>	263
<code>\@lign</code>	604
<code>\@namedef</code>	553, 585, 596, 614
<code>\@one</code>	499, 504, 514, 519, 574, 593, 599
<code>\@subformulasfalse</code>	711
<code>\@subformulastrue</code>	715
<code>\@totalleftmargin</code>	574
<code>\@xp</code>	536
<code>\add@amps</code>	600
<code>\align@</code>	460
<code>\alignsep@</code>	602
<code>\black@</code>	472, 476
<code>\common@align@ending</code>	475, 491, 496, 501, 506, 511, 516, 521, 526, 531
<code>\gather@</code>	460
<code>\if@eqnsw</code>	598
<code>\if@subformulas</code>	410, 701, 744
<code>\ifingather@</code>	478
<code>\ifmeasuring@</code>	539, 555, 571, 745
<code>\ifst@rred</code>	586, 592, 597, 610
<code>\iftag@</code>	603
<code>\iftagsleft@</code>	536
<code>\intertext@</code>	564
<code>\lendmultline@</code>	536
<code>\lineht@</code>	611
<code>\m@ne</code>	514, 519, 524, 529
<code>\m@th</code>	3, 5, 14, 18, 454, 556, 560
<code>\make@display@tag</code>	588, 604
<code>\math@cr</code>	472, 476
<code>\maxfields@</code>	600
<code>\measuring@true</code>	539
<code>\multline@</code>	460
<code>\on@line</code>	265
<code>\place@tag</code>	605
<code>\rendmultline@</code>	536
<code>\reserved@a</code>	263, 264, 271
<code>\restore@math@cr</code>	614, 619
<code>\restorealignstate@</code>	479
<code>\row@</code>	593, 599
<code>\setboxz@h</code>	604
<code>\split@tag</code>	14
<code>\st@rredfalse</code>	489, 499, 514, 524
<code>\st@rredtrue</code>	466, 494, 504, 509, 519, 529, 534
<code>\start@align</code>	489, 494, 499, 504, 509, 514, 519, 524, 529
<code>\start@gather</code>	466
<code>\start@multline</code>	534
<code>\strut@</code>	604
<code>\tag@true</code>	598
<code>\totwidth@</code>	472, 476
<code>\tw@</code>	509, 524, 529
<code>\z@</code>	489, 494, 611
tex commands:	
<code>\tex_everydisplay:D</code>	427, 429
<code>\tex_everymath:D</code>	418, 420
<code>\tex_the:D</code>	420, 429
<code>\tiny</code>	65
tl commands:	
<code>\tl_gput_right:Nn</code>	9, 316, 346, 463
<code>\tl_gset:Nn</code>	25, 26, 383, 696, 697
<code>\tl_if_blank:nTF</code>	35, 52, 102
<code>\tl_if_empty:NTF</code>	708, 729, 747
<code>\tl_if_in:nnTF</code>	18
<code>\tl_new:N</code>	12, 13, 115, 377, 702
<code>\tl_set:Nn</code>	128, 181, 223, 679, 689
<code>\tl_to_str:n</code>	173
<code>\tl_trim_spaces_apply:nN</code>	19
<code>\l_tmpa_tl</code>	679, 680, 681, 689, 691
tl internal commands:	
<code>\l_math_env_name_tl</code>	7, 115, 121, 128, 181, 223
<code>\g_math_grabbed_env_tl</code>	4, 12, 25, 386, 390, 395, 716, 735
<code>\g_math_grabbed_math_tl</code>	4, 13, 26, 388, 705, 707, 708, 726, 728, 729, 747
<code>\tmpmathcontent</code>	377, 383, 394, 396, 397
<code>\tracingall</code>	157, 210, 245
<code>\tracingnone</code>	162, 214
<code>\typeout</code>	85, 112, 144, 147, 197, 200, 232, 235, 252, 276, 283, 287, 292, 298, 302, 306, 381, 407, 432, 554, 706, 707, 710, 714, 727, 728, 731, 734, 757
<b>U</b>	
use commands:	
<code>\use_ii_i:nn</code>	660
<code>\UseHook</code>	261
<b>V</b>	
<code>\vbox</code>	570
<code>\vskip</code>	569, 581