

The `latex-lab-math` code*

Frank Mittelbach, Joseph Wright, L^AT_EX Project

v0.5c 2023-09-11

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

1	Introduction	2
1.1	Code level interfaces	2
1.2	Document level interfaces	2
2	Known current bugs, etc.	2
2.1	Capture/grabbing problems	3
2.2	Other problems	3
2.3	Other Todos	3
3	The Implementation	3
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	7
3.7	Document commands	11
3.8	<code>\everymath</code> and <code>\everydisplay</code>	13
3.9	Modifying kernel environments	14
3.10	Modifying <code>amsmath</code>	15
	Index	21

*

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

<code>\math_register_env:n</code>	<code>\math_register_env:n {<env>}</code>
<code>\math_register_env:nn</code>	<code>\math_register_env:nn {<env>} {<options>}</code>

Registers the `<env>` 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 `<options>` may also be given:

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

<code>\math_processor:n</code>	<code>\math_processor:n {<tokens>}</code>
--------------------------------	---

Declares that the captured math content should be passed to the `<tokens>`, which will receive the environment type as `#1` and the content as `#2`.

1.2 Document level interfaces

<code>\RegisterMathEnvironment</code>	<code>\RegisterMathEnvironment [<options>] {<env>}</code>
---------------------------------------	---

Registers the `<env>` 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 `<options>` 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 $\backslash\lrcorner\lrcorner$), 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-math.ltx}
4     [\ltxlabmathdate\space
5     v\ltxlabmathversion\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}}
8 \ExplSyntaxOn
```

3.2 Setup

Loading `amsmath` is an absolute requirement: this avoids needing to have conditional definitions and deals with how to define `\[/\]` neatly.

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

3.3 Data structures

`\l__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 \l__math_collected_bool
```

Change first tl name below: 'env' => 'info'?

Or do we need an extra

`\g__math_grabbed_env_tl`
`\g__math_grabbed_math_tl`

```
12 \tl_new:N \g__math_grabbed_env_tl
13 \tl_new:N \g__math_grabbed_math_tl
```

3.4 Interface commands

```
\__math_process:nn
\__math_process:Vn
\__math_process_auxi:nn
\__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 \__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} \__math_process_auxi:nn {#1} }
20   }
21 }
22 \cs_generate_variant:Nn \__math_process:nn { V }
23 \cs_new_protected:Npn \__math_process_auxi:nn #1#2
24 {
25   \tl_gset:Nn \g__math_grabbed_env_tl {#2}
26   \tl_gset:Nn \g__math_grabbed_math_tl {#1}
27   \__math_process_auxii:nn {#2} {#1}
28 }
29 \cs_new_protected:Npn \__math_process_auxii:nn #1#2 { }
```

(End of definition for `__math_process:nn`, `__math_process_auxi:nn`, and `__math_process_auxii:nn`.)

`\math_processor:n` A simple installer

```
30 \cs_new_protected:Npn \math_processor:n #1
31 { \cs_set_protected:Npn \__math_process_auxii:nn ##1##2 {#1} }
```

(End of definition for `\math_processor:n`. This function is documented on page 2.)

3.5 Content grabbing

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       \legacy_if:nTF { measuring@ }
40         { #1 $ }
41         {
42           \tl_if_in:nnTF {#1} { \m@th }
43             { #1 $ }
44             {
45               \tagmccend %end P-chunk, in code: \tag_mc_end_push:
46               \@kernel@math@begin
47               #1 $
48               \@kernel@math@end
49               \tagmcbegin{} % restart P-chunk (whatsits in pdftex)
50             }
51           }
52         }
53       }
54     }

```

(End of definition for `__math_grab_dollar:w`.)

And for the classical \TeX display structure.

```

55
56 \skip_new:N \l__math_tmpa_skip
57
58 \cs_new_protected:Npn \__math_grab_dollardollar:w % $$
59   #1 $$
60   {
61     \tl_if_blank:nF {#1}
62     {
63       \__math_tag_dollardollar_display:nn { equation* }{#1}
64       #1
65       $$
66     }
67   }

```

To allow to use the code without tagging we guard. But probably `tagpdf` should provide some tools for such manual para-ends.

```

68 \cs_new_protected:Npn \@kernel@close@P {
69   \tag_if_active:T
70   {
71     \tagmccend %end P-chunk, should perhaps be \tag_mc_end_push: ...

```

```

72      \int_gincr:N \g__tag_para_end_int
73      \bool_if:NT \l__tag_para_show_bool
74      { \tag_mc_begin:n{artifact}
75        \rlap{\color_select:n{red}\tiny\ \int_use:N\g__tag_para_end_int}
76        \tag_mc_end:
77      }
78      \tag_struct_end:
79    }
80  }
81
82
83
84  \cs_new_protected:Npn \__math_tag_dollardollar_display:nn #1#2 {
85    \__math_process:nn {#1} {#2}
86    \@kernel@close@P
87    \@kernel@math@begin
88    %      \skip_set:Nn \belowdisplayskip {-\belowdisplayskip}
89    %      \skip_set:Nn \belowdisplayskip {-\belowdisplayskip}
90    %      \int_set:Nn \postdisplaypenalty {10000}
91    %%
92    %      \group_insert_after:N \__math_tag_dollardollar_display_end:
93  }
94
95  \cs_new_protected:Npn \__math_tag_dollardollar_display_end: {
96    % \typeout{== tag dollarldollar display end}
97    % \ShowTagging{struct-stack}
98    \tagpdfparaOff
99    \para_raw_end:
100   \tagpdfparaOn
101   \l__math_tmpa_skip \lastskip
102   \@kernel@math@end
103   \penalty \postdisplaypenalty

```

This reinserts the below display skips. It must be doubled to get the right amount:

```

104   \skip_vertical:n { -\l__math_tmpa_skip * 2 }
105   %
106   \@doendpe          % this has no \end{...} to take care of it
107 }
108

```

(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.

```

109 \cs_new_protected:Npn \__math_grab_inline:w % \ (
110   #1 \)
111   {
112     \tl_if_blank:nF {#1}
113     {
114       \__math_process:nn { math } {#1}
115       $ #1 $
116     }
117     \bool_set_false:N \l__math_collected_bool
118   }

```

(End of definition for __math_grab_inline:w.)

```

\__math_grab_eqn:w For the most common use of \[/\]: turn into an environment.
119 \cs_new_protected:Npn \__math_grab_eqn:w % \[
120   #1 \]
121   {
122   % \typeout{collected? = \bool_if:NTF \l__math_collected_bool {true}{false}}
123   \begin { equation* } #1 \end { equation* }
124   }

(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”

```

125 \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 `ltxcmd`.

```

126 \keys_define:nn { __math }
127   {
128     arg-spec .code:n =
129     {
130       \ExpandArgs { c } \DeclareDocumentCommand
131       { __math_env \l__math_env_name_tl _aux: }
132       { #1 }
133       { \__math_env_forward:w }
134     }
135   }

```

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

```

136 \cs_new_protected:Npn \math_register_env:nn #1#2
137   {
138     \tl_set:Nn \l__math_env_name_tl { #1 }
139     \keys_set:nn { __math } { #2 }
140     \cs_gset_eq:cc { __math_env_ #1 _begin: } { #1 }
141     \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
142     %
143     \ExpandArgs { nnx } \RenewDocumentEnvironment { #1 } { b }
144     {
145       % \bool_set_true:N \exp_not:N \l__math_collected_bool
146       % \cs_if_exist:cTF { __math_env_ #1 _aux: }
147       % {
148       %   \exp_not:c { __math_env_ #1 _aux: }
149       %   #####1 \exp_not:N \__math_env_end: { #1 }
150       % }
151       % { \exp_not:N \__math_process:nn { #1 } { #####1 } }
152       \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
153       {

```

```

154 % \typeout{==>B1}
155 }
156 {
157 % \typeout{==>B2}
158 \cs_if_exist:cTF { __math_env #1 _aux: }
159 {
160 \exp_not:c { __math_env #1 _aux: }
161 #####1 \exp_not:N \__math_env_end: {#1}
162 }
163 { \exp_not:N \__math_process:nn {#1} {####1} }
164 \exp_not:n { \@kernel@math@registered@begin }
165 \bool_set_true:N \exp_not:N \l__math_collected_bool
166 }
167 % \exp_not:N \tracingall
168 \exp_not:c { __math_env_ #1 _begin: }
169 #####1
170 \exp_not:c { __math_env_ #1 _end: }
171 % \exp_not:c { __math_env_ #1 _end: }
172 % \exp_not:N \tracingnone
173 % \exp_not:n { \@kernel@math@registered@end }
174 }
175 {
176 }
177 }
178
179 \cs_set_protected:Npn \__cs_tmp:w #1
180 {
181 \group_begin:
182 \exp_args:No \__cs_generate_internal_variant:n
183 { \tl_to_str:n {#1} }
184 \group_end:
185 }
186 \__cs_tmp:w { nnxx }
187
188
189 \cs_new_protected:Npn \math_register_halign_env:nn #1#2
190 {
191 \tl_set:Nn \l__math_env_name_tl {#1}
192 \keys_set:nn { __math } {#2}
193 \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
194 \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
195 %
196 \ExpandArgs { nnxx } \RenewDocumentEnvironment {#1} { b }
197 {
198 % \bool_set_true:N \exp_not:N \l__math_collected_bool
199 % \cs_if_exist:cTF { __math_env #1 _aux: }
200 % {
201 % \exp_not:c { __math_env #1 _aux: }
202 % #####1 \exp_not:N \__math_env_end: {#1}
203 % }
204 % { \exp_not:N \__math_process:nn {#1} {####1} }
205 \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
206 {
207 % \typeout{==>B1}

```



```

208     }
209     {
210 %       \typeout{==>B2}
211       \cs_if_exist:cTF { __math_env #1 _aux: }
212       {
213         \exp_not:c { __math_env #1 _aux: }
214         #####1 \exp_not:N \__math_env_end: {#1}
215       }
216       { \exp_not:N \__math_process:nn {#1} {#####1} }
217       \exp_not:n { \@kernel@math@registered@begin }
218       \bool_set_true:N \exp_not:N \l__math_collected_bool
219     }
220 %     \exp_not:N \tracingall
221     \exp_not:c { __math_env_ #1 _begin: }
222     #####1
223 %     \exp_not:c { __math_env_ #1 _end: }
224 %     \exp_not:N \tracingnone
225   }
226   {
227     \exp_not:c { __math_env_ #1 _end: }
228   }
229 }
230
231 \cs_new_protected:Npn \math_register_odd_env:nn #1#2
232 {
233   \tl_set:Nn \l__math_env_name_tl {#1}
234   \keys_set:nn { __math } {#2}
235   \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
236   \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
237 %
238   \ExpandArgs { nxxx } \RenewDocumentEnvironment {#1} { b }
239   {
240     \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
241     {
242 %       \typeout{==>B1}
243     }
244     {
245 %       \typeout{==>B2}
246       \cs_if_exist:cTF { __math_env #1 _aux: }
247       {
248         \exp_not:c { __math_env #1 _aux: }
249         #####1 \exp_not:N \__math_env_end: {#1}
250       }
251       { \exp_not:N \__math_process:nn {#1} {#####1} }
252       \exp_not:n { \@kernel@math@registered@begin }
253       \bool_set_true:N \exp_not:N \l__math_collected_bool
254     }
255 %     \exp_not:N \tracingall
256     \exp_not:c { __math_env_ #1 _begin: }
257     #####1
258   }
259   {
260     \exp_not:c { __math_env_ #1 _end: }
261 % needed if we don't have $$...$$

```

```

262 % \exp_not:n { \typeout{---> @kernel@math@registered@end }}
263 \exp_not:n { \@kernel@math@registered@end }
264 }
265 }
266
267
268 % FMi: compare with block change!
269 %
270 % \DeclareRobustCommand*\begin[1]{%
271 % \UseHook{env/#1/before}%
272 % \@ifundefined{#1}%
273 % {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
274 % {\def\reserved@a{\def\@currenvir{#1}%
275 % \edef\@currenvline{\on@line}%
276 % \execute@begin@hook{#1}%
277 % \csname #1\endcsname}}%
278 % \@ignorefalse
279 % \begingroup
280 % \@endpefalse % tmp!!! is it ok to drop this here?
281 % \reserved@a}
282
283
284 \cs_new:Npn \@kernel@math@registered@begin {
285 % \ShowTagging{struct-stack}
286 %\typeout{==>A1}\ShowTagging{struct-stack,mc-current}
287 \mode_if_vertical:TF
288 {
289 % \legacy_if:nTF { @endpe }
290 % { \legacy_if_set_false:n { @endpe } }
291 % { \__block_list_beginpar_vmode: }
292 %
293 % \typeout{==>~ at:~ \g__tag_struct_tag_tl}
294 %
295 \exp_args:Noo\str_if_eq:nnF \g__tag_struct_tag_tl { \l__tag_para_main_tag_tl } %
296 {
297 % \typeout{==>A2}
298 \__block_beginpar_vmode:
299 } % needs correction!
300 }
301 {
302 % \typeout{==>A3}
303 \@kernel@close@P
304 % \tagmcend % needs correction!
305 }
306 \@kernel@math@begin
307 \tagpdfparaOff
308 % \typeout{==>MC1}\ShowTagging{mc-current}
309 }
310
311 \cs_new:Npn \@kernel@math@registered@end {
312 % \typeout{==>MC2}\ShowTagging{mc-current}
313 \para_raw_end:
314 \tagpdfparaOn
315 \@kernel@math@end

```

```

316 % \typeout{==>MC3}\ShowTagging{mc-current}
317 \endpetrue
318 }
319
320 \cs_new_protected:Npn \math_register_env:n #1
321 { \math_register_env:nn {#1} { } }
322 \NewDocumentCommand \RegisterMathEnvironment { 0{ } m }
323 { \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`

```

324 \cs_new_protected:Npn \__math_env_forward:w #1 \__math_env_end: #2
325 { \__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
\__math_equation_begin:
\equation*
\__math_equation_star_begin:
\endequation
\__math_equation_end:
\endequation*
\__math_equation_star_end:

```

These environments are not set up by `amsmath` to collect their body, so we do that here. This has to be done *after* we can be sure `amsmath` is loaded.

Note that with `amsmath` loaded, `equation*` and `equation` are the two basics: they are used to define the other single-row display environments, etc.

```

326 \tl_gput_right:Nn \@kernel@before@begindocument
327 {
328   \math_register_env:n { equation }
329   \math_register_env:n { equation* }
330   % at the moment register_env can only do display math
331   % \math_register_env:n { math }
332   \RenewDocumentEnvironment{math} {b}{\$#1\$}{}
333   % and this one doesn't work either
334   % \math_register_env:n { displaymath }
335   \RenewDocumentEnvironment{displaymath} {b}{\[#1\]}{}
336 }

```

(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.

```

337 \cs_gset_protected:Npn \( % \)
338 {
339   \bool_if:NTF \l__math_collected_bool
340   {
341     \mode_if_math:TF
342     { \@badmath }
343     { $ }

```

```

344     }
345     {
346         \bool_set_true:N \l__math_collected_bool
347         \__math_grab_inline:w
348     }
349 } % \l(
350 \cs_gset_protected:Npn \l(
351 {
352     \mode_if_math:TF
353     { $ }
354     { \@badmath }
355 }

```

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

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

```

356 \tl_gput_right:Nn \@kernel@before@begindocument
357 {
358     \cs_gset_protected:Npn \l[ % \l]
359     {
360         \bool_if:NTF \l__math_collected_bool
361         { \begin { equation* } }
362         { \__math_grab_eqn:w }
363     } % \l[
364 \cs_gset_protected:Npn \l]
365 {
366     \bool_if:NTF \l__math_collected_bool
367     { \end{ equation* } }
368     { \@badmath }
369 }
370 }

```

(End of definition for $\l[$ and $\l]$. 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.

```

371 %\cs_gset_protected:Npn \ensuremath #1
372 % {
373 %     \mode_if_math:TF
374 %     {#1}
375 %     {
376 %         \bool_if:NTF \l__math_collected_bool
377 %         { \@ensuredmath {#1} }
378 %         {
379 %             \bool_set_true:N \l__math_collected_bool
380 %             \__math_process:nn { math } {#1}
381 %             \@ensuredmath {#1}
382 %             \bool_set_false:N \l__math_collected_bool
383 %         }
384 %     }
385 % }

```

(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” TeX display. Most display math mode is actually handled elsewhere, as we have macro control.

```

386
387 \tl_new:N\tmpmathcontent
388
389
390 \def\@kernel@math@begin {
391 % \typeout{==>-math-begin}
392 % needs different handling if we support nesting
393 \tl_gset:Nx\tmpmathcontent
394 {
395 LaTeX~ formula~ starts~
396 \exp_not:N\begin{\g__math_grabbed_env_tl}
397 \space
398 \exp_not:V\g__math_grabbed_math_tl
399 \space
400 \exp_not:N\end{\g__math_grabbed_env_tl}
401 \space LaTeX~ formula~ ends~
402 }
403 \tagstructbegin{tag=Formula,
404 AFinline-o=\tmpmathcontent,
405 title-o=\g__math_grabbed_env_tl,
406 actualtext=\tmpmathcontent
407 % alt=\tmpmathcontent
408 }
409 % inner formula if multiple parts (not really implemented yet)
410 \grabaformulapartandstart
411 % the above does:
412 % \tagstructbegin{tag=Formula}\tagmcbegin{}
413 % or just
414 % \tagmcbegin{}
415 }
416 \def\@kernel@math@end {
417 % \typeout{==>-math-end}
418 % \ShowTagging{struct-stack}
419 \tagmcend
420 \if@subformulas
421 \tagstructend
422 \else
423 \fi
424 \tagstructend
425 % \ShowTagging{struct-stack}
426 }
427
428 \exp_args:No \tex_everymath:D
429 {
430 \tex_the:D \tex_everymath:D
431 \bool_if:NF \l__math_collected_bool
432 {
433 \bool_set_true:N \l__math_collected_bool
434 \__math_grab_dollar:w
435 }

```

```

436 }
437
438 \exp_args:No \tex_everydisplay:D
439 {
440   \tex_the:D \tex_everydisplay:D
441   \iftrue % this may have to be a settable flag!
442   % {
443   %   \typeout{==>~ in~ everydisplay}

```

flipping the `\belowdisplay` values is done so that we get (assumption) a negative skip and not make the page bigger then we take that out, then we add the tagging code (in `__math_tag_dollardollar_display_end`) and then we put a real `\postdisplaypenalty` in and the right skip (of which we don't know if it is short or a normal `\belowdisplayskip`). This might need some refinement if that skip is actually negative from the start (not sure it ever is and is worth bothering about)

```

444   \skip_set:Nn \belowdisplayskip {-\belowdisplayskip}
445   \skip_set:Nn \belowdisplayshortskip {-\belowdisplayshortskip}
446   \int_set:Nn \postdisplaypenalty {10000}
447 %
448   \group_insert_after:N \__math_tag_dollardollar_display_end:
449 % }
450 \fi
451 \bool_if:NF \l__math_collected_bool
452 {
453   \bool_set_true:N \l__math_collected_bool
454   \__math_grab_dollardollar:w
455 }
456 }

```

3.9 Modifying kernel environments

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

```

457 \math_register_env:n { eqnarray }
458 \math_register_env:n { eqnarray* }
    Places where math mode is (ab)used.
459 \clist_map_inline:nn
460 { tabular }
461 {
462   \AddToHook{ env / #1 / begin }
463   { \bool_set_true:N \l__math_collected_bool }
464 }

```

`__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!

```

465 \cs_new_eq:NN \__math_m@th: \m@th
466 \cs_gset_protected:Npn \m@th
467 {
468   \bool_set_true:N \l__math_collected_bool
469   \__math_m@th:
470 }

```

(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`.

```

471
472
473
474 \tl_gput_right:Nn \@kernel@before@begindocument {
475 %
476 \renewenvironment{gather*}{%
477 \start@gather\st@rredtrue
478 }
479 {%
480 % this redirection doesn't work if we alter "gather"!
481 % \endgather
482 % so replace it with its real meaning
483 \math@cr \black@\totwidth@ \egroup
484 $$\ignorespacesafterend
485 }

486 \def\common@align@ending {
487 \math@cr \black@\totwidth@
488 \egroup
489 \ifingather@
490 \restorealignstate@
491 \egroup
492 \nonumber
493 \ifnum0='{ \fi \iffalse } \fi
494 \else
495 $$%
496 \fi
497 \ignorespacesafterend
498 }
499 \renewenvironment{alignat}{%
500 \start@align\z@\st@rredfalse
501 }{%
502 \common@align@ending
503 }
504 \renewenvironment{alignat*}{%
505 \start@align\z@\st@rredtrue
506 }{%
507 \common@align@ending
508 }
509 \renewenvironment{xalignat}{%
510 \start@align\@ne\st@rredfalse
511 }{%
512 \common@align@ending
513 }
514 \renewenvironment{xalignat*}{%
515 \start@align\@ne\st@rredtrue

```

```

516 }{%
517   \common@align@ending
518 }
519 \renewenvironment{xxalignat}{%
520   \start@align\tw@\st@rredtrue
521 }{%
522   \common@align@ending
523 }
524 \renewenvironment{align}{%
525   \start@align\@ne\st@rredfalse\m@ne
526 }{%
527   \common@align@ending
528 }
529 \renewenvironment{align*}{%
530   \start@align\@ne\st@rredtrue\m@ne
531 }{%
532   \common@align@ending
533 }
534 \renewenvironment{flalign}{%
535   \start@align\tw@\st@rredfalse\m@ne
536 }{%
537   \common@align@ending
538 }
539 \renewenvironment{flalign*}{%
540   \start@align\tw@\st@rredtrue\m@ne
541 }{%
542   \common@align@ending
543 }
544 %
545 \renewenvironment{multline*}{\start@multline\st@rredtrue}
546 {%
547   \iftagsleft@ \exp\lendmultline@ \else \exp\rendmultline@ \fi
548   \ignorespacesafterend
549 }

```

Also for false?

```

550 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_stop:}

551 %
552 \math_register_halign_env:nn {align}{}
553 \math_register_halign_env:nn {align*}{}
554 \math_register_halign_env:nn {flalign}{}
555 \math_register_halign_env:nn {flalign*}{}
556 \math_register_halign_env:nn {gather}{}
557 \math_register_halign_env:nn {gather*}{}
558 \math_register_halign_env:nn {multline}{}
559 \math_register_halign_env:nn {multline*}{}
560 \math_register_halign_env:nn {xalignat}{}
561 \math_register_halign_env:nn {xalignat*}{}
562 \math_register_halign_env:nn {xxalignat}{}
563 %
564 \@namedef{maketag @ @ @} #1{%
565   % \typeout{--->maketag @ @ @}
566   \ifmeasuring@
567     \hbox{\m@th\normalfont#1}%

```



```

568 \else
569 \tagmcend \tagstructbegin{tag=Lbl}%
570 \tagmcbegin{tag=Lbl}%
571 \hbox{\m@th\normalfont#1}%
572 \tagmcend \tagstructend \tagmcbegin{}}%
573 \fi
574 }

575 \def\intertext@{%
576 \def\intertext##1{%
577 \ifvmode\else\\@empty\fi
578 \noalign{%
579 % we have to flip the sign on the skip because we flipped it on the outside
580 \penalty\postdisplaypenalty\vskip-\belowdisplayskip
581 \vbox{

```

Stop tagging when measuring:

```

582 \ifmeasuring@\tag_stop:\fi
583 \normalbaselines
584 \ifdim\linewidth=\columnwidth
585 \else \parshape\@ne \@totalleftmargin \linewidth
586 \fi

```

End the previous mc:

```

587 \tag_mc_end_push:

```

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

```

588 \tagpdfsetup{paratag=P}%
589 \tagpdfpara0n
590 \noindent\ignorespaces##1\par

```

Restart the MC

```

591 \tag_mc_begin_pop:n{}}%
592 \penalty\predisplaypenalty\vskip\abovedisplayskip%
593 }%
594 }
595 }

```

```

596 \@namedef{math@cr @ @ @ gather}{%
597 \ifst@rred\nonumber\fi
598 &\relax
599 \make@display@tag
600 %
601 \maybestartnewformulatag
602 %
603 \ifst@rred\else\global\@eqnswtrue\fi
604 \global\advance\row@\@ne
605 \cr
606 }

```

```

607 \@namedef{math@cr @ @ @ align}{%
608 \ifst@rred\nonumber\fi
609 \if@eqnsw \global\tag@true \fi
610 \global\advance\row@\@ne
611 \add@amps\maxfields@
612 \omit
613 \kern-\alignsep@

```

if we use 2 levels of formulas this would need changing

not true any longer

```

614 \iftag@
615 \setboxz@h{\@align\strut@{\make@display@tag}}%
616 \place@tag
617 \fi
618 %
619 \maybestartnewformulatag
620 %
621 \ifst@rred\else\global\@eqnswtrue\fi
622 \global\lineht@z@
623 \cr
624 }
625 \def\restore@math@cr{\@namedef{math@cr @ @ @}{
626 %
627 \maybestartnewformulatag
628 %
629 \cr}}
630 \restore@math@cr
631 }

```

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

```

632 \cs_new:Npn \_math\_split\_at\_nl\_first:w #1 \ \ #2 \ \ #3 \s\_stop
633 {
634 \quark\_if\_nil:nTF {#2}
635 { {#1} { } }
636 {
637 \_math\_split\_chk\_if\_begin:ww #1 \begin \q\_nil \s\_mark
638 #2 \ \ #3 \s\_stop
639 }
640 }
641 \cs_new:Npn \_math\_split\_chk\_if\_begin:ww #1 \begin #2 #3 \s\_mark
642 #4 \ \ \q\_nil \ \ \s\_stop
643 {
644 \quark\_if\_nil:nTF {#2}
645 { {#1} {#4} }
646 {
647 \exp\_after:wN \_math\_split\_collect\_one\_end:w
648 \_math\_split\_cleanup\_begin\_q\_nil:w #1 \begin{#2} #3 \ \ #4 \s\_stop
649 { } { 1 }
650 }
651 }
652 \cs_new:Npn \_math\_split\_cleanup\_begin\_q\_nil:w #1 \begin \q\_nil {#1}
653 \cs_new:Npn \_math\_split\_collect\_one\_end:w #1 \end #2 #3 \s\_stop #4 #5
654 {
655 \exp\_args:Nf \_math\_split\_check\_count\_begins:nnnn
656 { \_math\_split\_count\_begins:n { #4 #1 } } {#5}
657 { #4 #1 \end{#2} } {#3}
658 }
659 \cs_new:Npn \_math\_split\_count\_begins:n #1
660 { \int\_eval:n { 0 \_math\_split\_count\_begins:w #1 \begin \q\_nil } }
661 \cs_new:Npn \_math\_split\_count\_begins:w #1 \begin #2
662 { \quark\_if\_nil:nF {#2} { +1 \_math\_split\_count\_begins:w } }

```

```

663 \cs_new:Npn \__math_split_check_count_begins:nnnn #1 #2 #3 #4
664 {
665   \int_compare:nNnTF {#1} = {#2}
666   {
667     \exp_last_unbraced:Nf \__math_split_final_cleanup:nn
668     { \split:n { \__math_split_guard:n {#3} #4 } }
669   }
670   {
671     \exp_args:No \use_ii_i:nn
672     { \exp_after:wN { \int_value:w \int_eval:n { #2 + 1 } } }
673     { \__math_split_collect_one_end:w #4 \s_stop {#3} }
674   }
675 }
676 \cs_new:Npn \__math_split_final_cleanup:nn #1 #2
677 {
678   \exp:w \__math_split_final_cleanup:w #1
679   \__math_split_guard:n \q_nil \s_mark { }
680   {#2}
681 }
682 \cs_new:Npn \__math_split_final_cleanup:w #1 \__math_split_guard:n #2 #3 \s_mark #4
683 {
684   \quark_if_nil:nTF {#2}
685   { \exp_end: { #4 #1 } }
686   { \__math_split_final_cleanup:w #3 \s_mark { #4 #1 #2 } }
687 }
688 \NewDocumentCommand \splitnl { mm +m }
689 {
690   \tl_set:Nf \l_tmpa_tl { \split:n {#3} }
691   \show \l_tmpa_tl
692   \exp_after:wN \__splitnl_aux:nnNN \l_tmpa_tl #1 #2
693 }
694
695
696 \cs_new:Npn \split:n #1 {
697   \__math_split_at_nl_first:w #1 \q_nil \s_stop }
698
699 \cs_new:Npn \__math_split_at_nl:NN #1#2 {
700   \tl_set:Nf \l_tmpa_tl {
701     \exp_after:wN \__math_split_at_nl_first:w #1 \q_nil \s_stop }
702   \exp_after:wN \__math_split_at_nl_aux:nnNN \l_tmpa_tl #1 #2
703 }
704
705 \cs_new_protected:Npn \__math_split_at_nl_aux:nnNN #1 #2 #3 #4
706 {
707   \tl_gset:Nn #4 {#1}
708   \tl_gset:Nn #3 {#2}
709 }
710

```

(End of definition for .)

\maybestartnewformulatag

```

711
712 \newif\if@subformulas

```

```

713 \tl_new:N \result
714
715 \cs_new_protected:Npn\grabaformulapartandstart {
716   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
717   \typeout{====>first-result=\meaning\result}
718   \typeout{====>first-tmpmathcontent=\meaning\g__math_grabbed_math_tl}
719   \tl_if_empty:NTF \g__math_grabbed_math_tl
720   {
721     \typeout{====>formula~ has~ no~ subparts}
722     \global\@subformulasfalse
723   }
724   {
725     \typeout{====>formula~ has~ subparts}
726     \global\@subformulastrue
727     \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
728     \tagstructbegin{tag=Formula,

```

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

```

729 %       alt=\result,
730 %       title-o=\resulttitle
731 %     }
732 %   }
733 %   \tagmcbegin{}
734 % }
735
736 \cs_new_protected:Npn\grabaformulapartandmayberestart {
737   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
738   \typeout{====>result=\meaning\result}
739   \typeout{====>tmpmathcontent=\meaning\g__math_grabbed_math_tl}
740 %   \tl_if_empty:NTF \g__math_grabbed_math_tl
741 %   {
742 %     \typeout{====>tmpmathcontent=empty}
743 %   }
744 %   {
745 %     \typeout{====>tmpmathcontent=not-empty}
746 %     \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
747 %     \tagstructbegin{tag=Formula,
748 %       alt=\result,
749 %       title-o=\resulttitle
750 %     }
751 %   }
752 %   \tagmcbegin{}
753 % }

```

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

```

754 \def\maybestartnewformulatag {
755   \if@subformulas
756   \ifmeasuring@\else
757   %
758   \tl_if_empty:NF \g__math_grabbed_math_tl
759   {
760     \tagmcend
761     \tagstructend
762     \grabaformulapartandmayberestart

```

```

763     }
764   \fi
765 \fi
766 }

```

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

```

767 %\AddToHook{package/breqn/after}{
768 % \typeout{==>~ in~ hook}
769 % \math_register_halign_env:nn {dmath}{\}
770 % \math_register_halign_env:nn {dgroup*}{\}
771 %}

772 \ExplSyntaxOff

773 <@@=>

774 %

775 </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		bool commands:	
\(109	\bool_if:NTF	73, 122, 152, 205, 240, 339, 360, 366, 376, 431, 451
\(<u>337</u>	\bool_new:N	11
\)	110	\bool_set_false:N	117, 382
\)	<u>337</u>	\bool_set_true:N	145, 165, 198, 218, 253, 346, 379, 433, 453, 463, 468
\[119, 335	bool internal commands:	
\[4, 7, 11, 12, <u>356</u>	\l__math_collected_bool	4, 11, 117, 122, 145, 152, 165, 198, 205, 218, 240, 253, 339, 346, 360, 366, 376, 379, 382, 431, 433, 451, 453, 463, 468
\]	577, 632, 638, 642, 648, 697, 701		
_	75		
\]	120, 335		
\]	4, 7, 11, 12, <u>356</u>		
A		C	
\abovedisplayskip	592	clist commands:	
\AddToHook	7, 462, 767	\clist_map_inline:nn	459
\advance	604, 610	color commands:	
		\color_select:n	75
		\columnwidth	584
		\cr	605, 623, 629
		cs commands:	
		\cs_generate_variant:Nn	22
		\cs_gset_eq:NN	
			140, 141, 193, 194, 235, 236
		\cs_gset_protected:Npn	
			337, 350, 358, 364, 371, 466
		\cs_if_exist:NTF	146, 158, 199, 211, 246
		\cs_new:Npn	284, 311, 632, 641, 652, 653, 659, 661, 663, 676, 682, 696, 699
B			
\begin	123, 270, 361, 396, 637, 641, 648, 652, 660, 661		
\begingroup	279		
\belowdisplay	14		
\belowdisplayshortskip	89, 445		
\belowdisplayskip	14, 88, 444, 580		
block internal commands:			
__block_beginpar_vmode:	298		
__block_list_beginpar_vmode:	291		

<code>\cs_new_eq:NN</code>	465	<code>\ExpandArgs</code>	130, 143, 196, 238
<code>\cs_new_protected:Npn</code> ...	14, 23, 29, 30, 32, 58, 68, 84, 95, 109, 119, 136, 189, 231, 320, 324, 705, 715, 736	<code>\ExplSyntaxOff</code>	772
<code>\cs_set_protected:Npn</code>	31, 179	<code>\ExplSyntaxOn</code>	8
cs internal commands:		F	
<code>__cs_generate_internal_variant:n</code>	182	<code>\fi</code>	423, 450, 493, 496, 547, 573, 577, 582, 586, 597, 603, 608, 609, 617, 621, 764, 765
<code>__cs_tmp:w</code>	179, 186	G	
<code>\csname</code>	277	<code>\global</code> 603, 604, 609, 610, 621, 622, 722, 726	
D		<code>\grabaformulapartandmayberestart</code> ...	736, 762
<code>\DeclareDocumentCommand</code>	130	<code>\grabaformulapartandstart</code>	410, 715
<code>\DeclareMathEnviornment</code>	3	group commands:	
<code>\DeclareRobustCommand</code>	270	<code>\group_begin:</code>	181
<code>\def</code>	273, 274, 390, 416, 486, 550, 575, 576, 625, 754	<code>\group_end:</code>	184
<code>\displaylines</code>	3	<code>\group_insert_after:N</code>	92, 448
E		H	
<code>\edef</code>	275, 727, 746	<code>\hbox</code>	567, 571
<code>\egroup</code>	483, 488, 491	I	
<code>\else</code>	422, 494, 547, 568, 577, 585, 603, 621, 756	<code>\ifdim</code>	584
<code>\end</code>	106, 123, 367, 400, 653, 657	<code>\iffalse</code>	493
end internal commands:		<code>\ifnum</code>	493
<code>__math_tag_dollardollar_-display_end</code>	14	<code>\iftrue</code>	441, 550
<code>\endcsname</code>	277	<code>\ifvmode</code>	577
<code>\endequation</code>	326	<code>\ignorespaces</code>	590
<code>\endequation*</code>	326	<code>\ignorespacesafterend</code>	484, 497, 548
<code>\endgather</code>	481	int commands:	
<code>\ensuremath</code>	3, 371	<code>\int_compare:nNnTF</code>	665
<code>\equalign</code>	3	<code>\int_eval:n</code>	660, 672
<code>\equation</code>	326	<code>\int_gincr:N</code>	72
<code>\equation*</code>	326	<code>\int_set:Nn</code>	90, 446
<code>\everydisplay</code>	13	<code>\int_use:N</code>	75
<code>\everymath</code>	12, 13	<code>\int_value:w</code>	672
exp commands:		<code>\intertext</code>	576
<code>\exp:w</code>	678	K	
<code>\exp_after:wN</code> .	647, 672, 692, 701, 702	<code>\kern</code>	613
<code>\exp_args:Nf</code>	655	keys commands:	
<code>\exp_args:No</code>	182, 428, 438, 671	<code>\keys_define:nn</code>	126
<code>\exp_args:Noo</code>	295	<code>\keys_set:nn</code>	139, 192, 234
<code>\exp_end:</code>	685	L	
<code>\exp_last_unbraced:Nf</code>	667	<code>\lastskip</code>	101
<code>\exp_not:N</code>	145, 148, 149, 151, 152, 160, 161, 163, 165, 167, 168, 170, 171, 172, 198, 201, 202, 204, 205, 213, 214, 216, 218, 220, 221, 223, 224, 227, 240, 248, 249, 251, 253, 255, 256, 260, 396, 400	legacy commands:	
<code>\exp_not:n</code>	164, 173, 217, 252, 262, 263, 398	<code>\legacy_if:nTF</code>	16, 39, 289
		<code>\legacy_if_set_false:n</code>	290
		<code>\let</code>	550
		<code>\linewidth</code>	584, 585
		<code>\ltlabmathdate</code>	4
		<code>\ltlabmathversion</code>	5

M	
math commands:	
\math_processor:n	2, 30, 30
\math_register_env:n	2,
136, 320, 328, 329, 331, 334, 457, 458	
\math_register_env:nn	
2, 136, 136, 321, 323	
\math_register_halign_env:nn	
189, 552, 553, 554, 555, 556,	
557, 558, 559, 560, 561, 562, 769, 770	
\math_register_odd_env:nn	231
math internal commands:	
__math_amsmath_align@:nn	471
__math_amsmath_gather@:n	471
__math_amsmath_multline@:n	471
__math_env_end:	
149, 161, 202, 214, 249, 324	
__math_env_forward:w	133, 324, 324
__math_equation_begin:	326
__math_equation_end:	326
__math_equation_star_begin:	326
__math_equation_star_end:	326
__math_grab_dollar:w	32, 32, 434
__math_grab_dollardollar:w	
55, 58, 454	
__math_grab_eqn:w	119, 119, 362
__math_grab_inline:w	109, 109, 347
__math_m@th:	465, 465, 469
__math_process:nn	
5, 14, 14, 22, 37, 85,	
114, 151, 163, 204, 216, 251, 325, 380	
__math_process_auxi:nn	14, 19, 23
__math_process_auxii:nn	14, 27, 29, 31
__math_split_at_nl:NN	699, 716, 737
__math_split_at_nl_aux:nnNN	702, 705
__math_split_at_nl_first:w	
632, 697, 701	
__math_split_check_count_-	
begins:nnnn	655, 663
__math_split_chk_if_begin:ww	
637, 641	
__math_split_cleanup_begin_q_-	
nil:w	648, 652
__math_split_collect_one_end:w	
647, 653, 673	
__math_split_count_begins:n	656, 659
__math_split_count_begins:w	
660, 661, 662	
__math_split_final_cleanup:nn	
667, 676	
__math_split_final_cleanup:w	
678, 682, 686	
__math_split_guard:n	668, 679, 682
__math_tag_dollardollar_-	
display:nn	63, 84
__math_tag_dollardollar_-	
display_end:	92, 95, 448
\maybe_start_new_formula_tag	
601, 619, 627, 711, 754	
\MaybeStop	3
\meaning	717, 718, 738, 739
mode commands:	
\mode_if_math:TF	341, 352, 373
\mode_if_vertical:TF	287
N	
\NewDocumentCommand	322, 688
\newif	712
\noalign	578
\noindent	590
\nonumber	492, 597, 608
\normalbaselines	583
\normalfont	567, 571
O	
\omit	612
P	
\par	590
para commands:	
\para_raw_end:	99, 313
\parshape	585
\penalty	103, 580, 592
\postdisplaypenalty	14, 90, 103, 446, 580
\predisplaypenalty	592
\ProvidesFile	3
Q	
quark commands:	
\q_nil	637, 642, 652, 660, 679, 697, 701
\quark_if_nil:nTF	634, 644, 662, 684
R	
\RegisterMathEnvironment	2, 3, 136
\relax	598
\RenewDocumentEnvironment	
143, 196, 238, 332, 335	
\renewenvironment	476, 499, 504,
509, 514, 519, 524, 529, 534, 539, 545	
\RequirePackage	7, 10
\result	713, 716, 717, 729, 737, 738, 748
\resulttitle	727, 730, 746, 749
\rlap	75
S	
scan commands:	
\s_mark	637, 641, 679, 682, 686

<code>\s_stop</code>	632, 638, 642, 648, 653, 673, 697, 701
<code>\show</code>	691
<code>\ShowTagging</code>	97, 285, 286, 308, 312, 316, 418, 425
skip commands:	
<code>\skip_new:N</code>	56
<code>\skip_set:Nn</code>	88, 89, 444, 445
<code>\skip_vertical:n</code>	104
skip internal commands:	
<code>\l_math_tmpa_skip</code>	56, 101, 104
<code>\space</code>	4, 5, 397, 399, 401, 727, 746
split commands:	
<code>\split:n</code>	668, 690, 696
<code>\splitnl</code>	688
splitnl internal commands:	
<code>_splitnl_aux:nnNN</code>	692
str commands:	
<code>\str_if_eq:nnTF</code>	295
T	
tag commands:	
<code>\tag_if_active:TF</code>	69
<code>\tag_mc_begin:n</code>	74
<code>\tag_mc_begin_pop:n</code>	591
<code>\tag_mc_end:</code>	76
<code>\tag_mc_end_push:</code>	45, 71, 587
<code>\tag_stop:</code>	550, 582
<code>\tag_struct_end:</code>	78
tag internal commands:	
<code>\g__tag_para_end_int</code>	72, 75
<code>\l__tag_para_main_tag_tl</code>	295
<code>\l__tag_para_show_bool</code>	73
<code>\g__tag_struct_tag_tl</code>	293, 295
<code>\tagmcbegin</code>	49, 412, 414, 570, 572, 733, 752
<code>\tagmcentd</code>	45, 71, 304, 419, 569, 572, 760
<code>\tagpdfparaOff</code>	98, 307
<code>\tagpdfparaOn</code>	100, 314, 589
<code>\tagpdfsetup</code>	588
<code>\tagstructbegin</code>	403, 412, 569, 728, 747
<code>\tagstructend</code>	421, 424, 572, 761
TeX and L ^A T _E X 2 _ε commands:	
<code>\@badmath</code>	342, 354, 368
<code>\@currenvir</code>	274
<code>\@currenvline</code>	275
<code>\@doendpe</code>	106
<code>\@eha</code>	273
<code>\@empty</code>	577
<code>\@endpefalse</code>	280
<code>\@endpetrue</code>	317
<code>\@ensuredmath</code>	5, 377, 381
<code>\@eqnswtrue</code>	603, 621
<code>\@execute@begin@hook</code>	276
<code>\@ifundefined</code>	272
<code>\@ignorefalse</code>	278
<code>\@kernel@before@begindocument</code>	9, 326, 356, 474
<code>\@kernel@close@P</code>	68, 86, 303
<code>\@kernel@math@begin</code>	46, 87, 306, 390
<code>\@kernel@math@end</code>	48, 102, 315, 416
<code>\@kernel@math@registered@begin</code>	164, 217, 252, 284
<code>\@kernel@math@registered@end</code>	173, 263, 311
<code>\@latex@error</code>	273
<code>\@align</code>	615
<code>\@namedef</code>	564, 596, 607, 625
<code>\@one</code>	510, 515, 525, 530, 585, 604, 610
<code>\@subformulasfalse</code>	722
<code>\@subformulastrue</code>	726
<code>\@totalleftmargin</code>	585
<code>\@exp</code>	547
<code>\add@amps</code>	611
<code>\align@</code>	471
<code>\alignsep@</code>	613
<code>\black@</code>	483, 487
<code>\common@align@ending</code>	486, 502, 507, 512, 517, 522, 527, 532, 537, 542
<code>\gather@</code>	471
<code>\if@eqnsw</code>	609
<code>\if@subformulas</code>	420, 712, 755
<code>\ifingather@</code>	489
<code>\ifmeasuring@</code>	550, 566, 582, 756
<code>\ifst@rred</code>	597, 603, 608, 621
<code>\iftag@</code>	614
<code>\iftagsleft@</code>	547
<code>\intertext@</code>	575
<code>\lendmultline@</code>	547
<code>\lineht@</code>	622
<code>\m@ne</code>	525, 530, 535, 540
<code>\m@th</code>	3, 5, 14, 18, 42, 465, 567, 571
<code>\make@display@tag</code>	599, 615
<code>\math@cr</code>	483, 487
<code>\maxfields@</code>	611
<code>\measuring@true</code>	550
<code>\multline@</code>	471
<code>\on@line</code>	275
<code>\place@tag</code>	616
<code>\rendmultline@</code>	547
<code>\reserved@a</code>	273, 274, 281
<code>\restore@math@cr</code>	625, 630
<code>\restorealignstate@</code>	490
<code>\row@</code>	604, 610
<code>\setboxz@h</code>	615
<code>\split@tag</code>	15
<code>\st@rredfalse</code>	500, 510, 525, 535
<code>\st@rredtrue</code>	477, 505, 515, 520, 530, 540, 545

<code>\start@align</code>	500,	<code>\l_tmpa_tl</code>	690, 691, 692, 700, 702
	505, 510, 515, 520, 525, 530, 535, 540	tl internal commands:	
<code>\start@gather</code>	477	<code>\l__math_env_name_tl</code>	
<code>\start@multline</code>	545		7, 125, 131, 138, 191, 233
<code>\strut@</code>	615	<code>\g__math_grabbed_env_tl</code>	
<code>\tag@true</code>	609		4, 12, 25, 396, 400, 405, 727, 746
<code>\totwidth@</code>	483, 487	<code>\g__math_grabbed_math_tl</code> 4, 13, 26,	
<code>\tw@</code>	520, 535, 540		398, 716, 718, 719, 737, 739, 740, 758
<code>\z@</code>	500, 505, 622	<code>\tmpmathcontent</code> . .	387, 393, 404, 406, 407
tex commands:		<code>\tracingall</code>	167, 220, 255
<code>\tex_everydisplay:D</code>	438, 440	<code>\tracingnone</code>	172, 224
<code>\tex_everymath:D</code>	428, 430	<code>\typeout</code> 96, 122, 154, 157, 207, 210, 242,	
<code>\tex_the:D</code>	430, 440		245, 262, 286, 293, 297, 302, 308,
<code>\tiny</code>	75		312, 316, 391, 417, 443, 565, 717,
tl commands:			718, 721, 725, 738, 739, 742, 745, 768
<code>\tl_gput_right:Nn</code> . . .	9, 326, 356, 474	U	
<code>\tl_gset:Nn</code>	25, 26, 393, 707, 708	use commands:	
<code>\tl_if_blank:nTF</code>	35, 61, 112	<code>\use_ii_i:nn</code>	671
<code>\tl_if_empty:NTF</code>	719, 740, 758	<code>\UseHook</code>	271
<code>\tl_if_in:nnTF</code>	18, 42	V	
<code>\tl_new:N</code>	12, 13, 125, 387, 713	<code>\vbox</code>	581
<code>\tl_set:Nn</code>	138, 191, 233, 690, 700	<code>\vskip</code>	580, 592
<code>\tl_to_str:n</code>	183		
<code>\tl_trim_spaces_apply:nN</code>	19		