

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 $\backslash\sqcup\backslash$), 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 (`\DeclareMathEnviornment`, 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.

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)]`

Temp loading ...

7 `\AddToHook{begindocument/before}{\RequirePackage{latex-lab-testphase-block}}`

8 `\ExplSyntaxOn`

Change description here?

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

storage?

`\g__math_grabbed_env_tl` contains the name of the math environment (`math` in the case of inline math, `\g__math_grabbed_math_tl` the math content).

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

```
\__math_process:Vn
\__math_process_auxi:nn
\__math_process_auxii:nn
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

```

We do not want math tagging in `fakemath` or when measuring, so we imitate the test inside `__math_process:nn` for now, see <https://github.com/latex3/tagging-project/issues/5> TODO: use socket to get more control about typesetting variants (tagged, drop etc)?

```

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 }

```

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

And for the classical \TeX display structure.

```

54
55 \skip_new:N \l__math_tmpa_skip
56
57 \cs_new_protected:Npn \__math_grab_dollardollar:w % $$
58   #1 $$
59   {
60     \tl_if_blank:nF {#1}
61     {
62       \__math_tag_dollardollar_display:nn { equation* }{#1}
63       #1
64       $$
65     }
66   }

```

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

```

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

```

```

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

```

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

```

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

```

(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.

```

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

```

(End of definition for __math_grab_inline:w.)

```

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

(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”

```

124 \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`.

```

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

```

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

```

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

```

```

153 % \typeout{==>B1}
154 }
155 {
156 % \typeout{==>B2}
157 \cs_if_exist:cTF { __math_env #1 _aux: }
158 {
159 \exp_not:c { __math_env #1 _aux: }
160 #####1 \exp_not:N \__math_env_end: {#1}
161 }
162 { \exp_not:N \__math_process:nn {#1} {####1} }
163 \exp_not:n { \@kernel@math@registered@begin }
164 \bool_set_true:N \exp_not:N \l__math_collected_bool
165 }
166 % \exp_not:N \tracingall
167 \exp_not:c { __math_env_ #1 _begin: }
168 #####1
169 \exp_not:c { __math_env_ #1 _end: }
170 % \exp_not:c { __math_env_ #1 _end: }
171 % \exp_not:N \tracingnone
172 % \exp_not:n { \@kernel@math@registered@end }
173 }
174 {
175 }
176 }
177
178 \cs_set_protected:Npn \__cs_tmp:w #1
179 {
180 \group_begin:
181 \exp_args:No \__cs_generate_internal_variant:n
182 { \tl_to_str:n {#1} }
183 \group_end:
184 }
185 \__cs_tmp:w { nnxx }
186
187
188 \cs_new_protected:Npn \math_register_halign_env:nn #1#2
189 {
190 \tl_set:Nn \l__math_env_name_tl {#1}
191 \keys_set:nn { __math } {#2}
192 \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
193 \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
194 %
195 \ExpandArgs { nnxx } \RenewDocumentEnvironment {#1} { b }
196 {
197 % \bool_set_true:N \exp_not:N \l__math_collected_bool
198 % \cs_if_exist:cTF { __math_env #1 _aux: }
199 % {
200 % \exp_not:c { __math_env #1 _aux: }
201 % #####1 \exp_not:N \__math_env_end: {#1}
202 % }
203 % { \exp_not:N \__math_process:nn {#1} {####1} }
204 \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
205 {
206 % \typeout{==>B1}

```



```

207     }
208     {
209 %       \typeout{==>B2}
210       \cs_if_exist:cTF { __math_env #1 _aux: }
211       {
212         \exp_not:c { __math_env #1 _aux: }
213         #####1 \exp_not:N \__math_env_end: {#1}
214       }
215       { \exp_not:N \__math_process:nn {#1} {#####1} }
216       \exp_not:n { \@kernel@math@registered@begin }
217       \bool_set_true:N \exp_not:N \l__math_collected_bool
218     }
219 %     \exp_not:N \tracingall
220     \exp_not:c { __math_env_ #1 _begin: }
221     #####1
222 %     \exp_not:c { __math_env_ #1 _end: }
223 %     \exp_not:N \tracingnone
224   }
225   {
226     \exp_not:c { __math_env_ #1 _end: }
227   }
228 }
229
230 \cs_new_protected:Npn \math_register_odd_env:nn #1#2
231 {
232   \tl_set:Nn \l__math_env_name_tl {#1}
233   \keys_set:nn { __math } {#2}
234   \cs_gset_eq:cc { __math_env_ #1 _begin: } {#1}
235   \cs_gset_eq:cc { __math_env_ #1 _end: } { end #1 }
236 %
237   \ExpandArgs { nxxx } \RenewDocumentEnvironment {#1} { b }
238   {
239     \exp_not:N \bool_if:NTF \exp_not:N \l__math_collected_bool
240     {
241 %       \typeout{==>B1}
242     }
243     {
244 %       \typeout{==>B2}
245       \cs_if_exist:cTF { __math_env #1 _aux: }
246       {
247         \exp_not:c { __math_env #1 _aux: }
248         #####1 \exp_not:N \__math_env_end: {#1}
249       }
250       { \exp_not:N \__math_process:nn {#1} {#####1} }
251       \exp_not:n { \@kernel@math@registered@begin }
252       \bool_set_true:N \exp_not:N \l__math_collected_bool
253     }
254 %     \exp_not:N \tracingall
255     \exp_not:c { __math_env_ #1 _begin: }
256     #####1
257   }
258   {
259     \exp_not:c { __math_env_ #1 _end: }
260 % needed if we don't have $$...$$

```

```

261 % \exp_not:n { \typeout{---> @kernel@math@registered@end }}
262 \exp_not:n { \@kernel@math@registered@end }
263 }
264 }
265
266
267 % FMi: compare with block change!
268 %
269 % \DeclareRobustCommand*\begin[1]{%
270 % \UseHook{env/#1/before}%
271 % \ifundefined{#1}%
272 % {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
273 % {\def\reserved@a{\def\@currenvir{#1}%
274 % \edef\@currenvline{\on@line}%
275 % \execute@begin@hook{#1}%
276 % \csname #1\endcsname}}%
277 % \@ignorefalse
278 % \begingroup
279 % \@endpefalse % tmp!!! is it ok to drop this here?
280 % \reserved@a}
281
282
283 \cs_new:Npn \@kernel@math@registered@begin {
284 % \ShowTagging{struct-stack}
285 %\typeout{==>A1}\ShowTagging{struct-stack,mc-current}
286 \mode_if_vertical:TF
287 {
288 % \legacy_if:nTF { @endpe }
289 % { \legacy_if_set_false:n { @endpe } }
290 % { \__block_list_beginpar_vmode: }
291 %
292 % \typeout{==>~ at:~ \g__tag_struct_tag_tl}
293 %
294 \exp_args:Noo\str_if_eq:nnF \g__tag_struct_tag_tl { \l__tag_para_main_tag_tl } %
295 {
296 % \typeout{==>A2}
297 \__block_beginpar_vmode:
298 } % needs correction!
299 }
300 {
301 % \typeout{==>A3}
302 \@kernel@close@P
303 % \tagmcend % needs correction!
304 }
305 \@kernel@math@begin
306 \tagpdfparaOff
307 % \typeout{==>MC1}\ShowTagging{mc-current}
308 }
309
310 \cs_new:Npn \@kernel@math@registered@end {
311 % \typeout{==>MC2}\ShowTagging{mc-current}
312 \para_raw_end:
313 \tagpdfparaOn
314 \@kernel@math@end

```

```

315 % \typeout{==>MC3}\ShowTagging{mc-current}
316 \endpetrue
317 }
318
319 \cs_new_protected:Npn \math_register_env:n #1
320 { \math_register_env:nn {#1} { } }
321 \NewDocumentCommand \RegisterMathEnvironment { 0{ } m }
322 { \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`

```

323 \cs_new_protected:Npn \__math_env_forward:w #1 \__math_env_end: #2
324 { \__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.

```

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

```

(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.

```

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

```

```

343     }
344     {
345         \__math_grab_inline:w
346     }
347 } % \ (
348 \cs_gset_protected:Npn \ (
349 {
350     \mode_if_math:TF
351     { $ }
352     { \@badmath }
353 }

```

(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.

```

354 \tl_gput_right:Nn \@kernel@before@begindocument
355 {
356     \cs_gset_protected:Npn \[ % \]
357     {
358         \bool_if:NTF \l__math_collected_bool
359         { \begin { equation* } }
360         { \__math_grab_eqn:w }
361     } % \[
362     \cs_gset_protected:Npn \]
363     {
364         \bool_if:NTF \l__math_collected_bool
365         { \end{ equation* } }
366         { \@badmath }
367     }
368 }

```

(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.

```

369 %\cs_gset_protected:Npn \ensuremath #1
370 % {
371 %     \mode_if_math:TF
372 %     {#1}
373 %     {
374 %         \bool_if:NTF \l__math_collected_bool
375 %         { \@ensuredmath {#1} }
376 %         {
377 %             \bool_set_true:N \l__math_collected_bool
378 %             \__math_process:nn { math } {#1}
379 %             \@ensuredmath {#1}
380 %             \bool_set_false:N \l__math_collected_bool
381 %         }
382 %     }
383 % }

```

(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.

```

384
385 \tl_new:N\tmpmathcontent
386
387
388 \def\@kernel@math@begin {
389 % \typeout{==>-math-begin}
390 % needs different handling if we support nesting
391 \tl_gset:Nx\tmpmathcontent
392 {
393 LaTeX~ formula~ starts~
394 \exp_not:N\begin{\g__math_grabbed_env_tl}
395 \space
396 \exp_not:V\g__math_grabbed_math_tl
397 \space
398 \exp_not:N\end{\g__math_grabbed_env_tl}
399 \space LaTeX~ formula~ ends~
400 }
401 \tagstructbegin{tag=Formula,
402 AInline-o=\tmpmathcontent,
403 title-o=\g__math_grabbed_env_tl,
404 actualtext=\tmpmathcontent
405 % alt=\tmpmathcontent
406 }
407 % inner formula if multiple parts (not really implemented yet)
408 \grabaformulapartandstart
409 % the above does:
410 % \tagstructbegin{tag=Formula}\tagmcbegin{}
411 % or just
412 % \tagmcbegin{}
413 }
414 \def\@kernel@math@end {
415 % \typeout{==>-math-end}
416 % \ShowTagging{struct-stack}
417 \tagmcend
418 \if@subformulas
419 \tagstructend
420 \else
421 \fi
422 \tagstructend
423 % \ShowTagging{struct-stack}
424 }
425
426 \exp_args:No \tex_everymath:D
427 {
428 \tex_the:D \tex_everymath:D
429 \bool_if:NF \l__math_collected_bool
430 {
431 \bool_set_true:N \l__math_collected_bool
432 \__math_grab_dollar:w
433 }

```

```

434 }
435
436 \exp_args:No \tex_everydisplay:D
437 {
438   \tex_the:D \tex_everydisplay:D
439   \iftrue % this may have to be a settable flag!
440   % {
441   %   \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)

```

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

```

3.9 Modifying kernel environments

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

```

455 \math_register_env:n { eqnarray }
456 \math_register_env:n { eqnarray* }
457
458   Places where math mode is (ab)used.
459
460 \clist_map_inline:nn
461 { tabular }
462 {
463   \AddToHook{ env / #1 / begin }
464   { \bool_set_true:N \l__math_collected_bool }
465 }

```

`__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!

```

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

```

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

3.10 Modifying amsmath

```

\__math_amsmath_align@:nn
\__math_amsmath_gather@:n
\__math_amsmath_multline@:n
    \align@
    \gather@
    \multline@

```

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

```

469
470
471
472 \tl_gput_right:Nn \@kernel@before@begindocument {
473 %
474 \renewenvironment{gather*}{%
475 \start@gather\st@rredtrue
476 }
477 {%
478 % this redirection doesn't work if we alter "gather"!
479 % \endgather
480 % so replace it with its real meaning
481 \math@cr \black@\totwidth@ \egroup
482 $$\ignorespacesafterend
483 }
484
485 \def\common@align@ending {
486 \math@cr \black@\totwidth@
487 \egroup
488 \ifingather@
489 \restorealignstate@
490 \egroup
491 \nonumber
492 \ifnum0='{ \fi \iffalse } \fi
493 $$%
494 \fi
495 \ignorespacesafterend
496 }
497 \renewenvironment{alignat}{%
498 \start@align\z@\st@rredfalse
499 }{%
500 \common@align@ending
501 }
502 \renewenvironment{alignat*}{%
503 \start@align\z@\st@rredtrue
504 }{%
505 \common@align@ending
506 }
507 \renewenvironment{xalignat}{%
508 \start@align\@ne\st@rredfalse
509 }{%
510 \common@align@ending
511 }
512 \renewenvironment{xalignat*}{%
513 \start@align\@ne\st@rredtrue

```

```

514 }{%
515   \common@align@ending
516 }
517 \renewenvironment{xxalignat}{%
518   \start@align\tw@\st@rredtrue
519 }{%
520   \common@align@ending
521 }
522 \renewenvironment{align}{%
523   \start@align\@ne\st@rredfalse\m@ne
524 }{%
525   \common@align@ending
526 }
527 \renewenvironment{align*}{%
528   \start@align\@ne\st@rredtrue\m@ne
529 }{%
530   \common@align@ending
531 }
532 \renewenvironment{flalign}{%
533   \start@align\tw@\st@rredfalse\m@ne
534 }{%
535   \common@align@ending
536 }
537 \renewenvironment{flalign*}{%
538   \start@align\tw@\st@rredtrue\m@ne
539 }{%
540   \common@align@ending
541 }
542 %
543 \renewenvironment{multline*}{\start@multline\st@rredtrue}
544 {%
545   \iftagsleft@ \exp\lendmultline@ \else \exp\rendmultline@ \fi
546   \ignorespacesafterend
547 }

```

Also for false?

```

548 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_stop:}
549 %
550 \math_register_halign_env:nn {align}{\}
551 \math_register_halign_env:nn {align*}{\}
552 \math_register_halign_env:nn {flalign}{\}
553 \math_register_halign_env:nn {flalign*}{\}
554 \math_register_halign_env:nn {gather}{\}
555 \math_register_halign_env:nn {gather*}{\}
556 \math_register_halign_env:nn {multline}{\}
557 \math_register_halign_env:nn {multline*}{\}
558 \math_register_halign_env:nn {xalignat}{\}
559 \math_register_halign_env:nn {xalignat*}{\}
560 \math_register_halign_env:nn {xxalignat}{\}
561 %
562 \@namedef{maketag @ @ @} #1{%
563   % \typeout{--->maketag @ @ @}
564   \ifmeasuring@
565     \hbox{\m@th\normalfont#1}%

```



```

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

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

```

Stop tagging when measuring:

```

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

```

End the previous mc:

```

585 \tag_mc_end_push:

```

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

```

586 \tagpdfsetup{paratag=P}%
587 \tagpdfpara0n
588 \noindent\ignorespaces##1\par

```

Restart the MC

```

589 \tag_mc_begin_pop:n{}}%
590 \penalty\predisplaypenalty\vskip\abovedisplayskip%
591 }%
592 }
593 }

```

```

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

```

```

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

```

if we use 2 levels of formulas this would need changing

not true any longer

```

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

```

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

```

630 \cs_new:Npn \_math\_split\_at\_nl\_first:w #1 \ \ #2 \ \ #3 \s\_stop
631 {
632 \quark\_if\_nil:nTF {#2}
633 { {#1} { } }
634 {
635 \_math\_split\_chk\_if\_begin:ww #1 \begin \q\_nil \s\_mark
636 #2 \ \ #3 \s\_stop
637 }
638 }
639 \cs_new:Npn \_math\_split\_chk\_if\_begin:ww #1 \begin #2 #3 \s\_mark
640 #4 \ \ \q\_nil \ \ \s\_stop
641 {
642 \quark\_if\_nil:nTF {#2}
643 { {#1} {#4} }
644 {
645 \exp\_after:wN \_math\_split\_collect\_one\_end:w
646 \_math\_split\_cleanup\_begin\_q\_nil:w #1 \begin{#2} #3 \ \ #4 \s\_stop
647 { } { 1 }
648 }
649 }
650 \cs_new:Npn \_math\_split\_cleanup\_begin\_q\_nil:w #1 \begin \q\_nil {#1}
651 \cs_new:Npn \_math\_split\_collect\_one\_end:w #1 \end #2 #3 \s\_stop #4 #5
652 {
653 \exp\_args:Nf \_math\_split\_check\_count\_begins:nnnn
654 { \_math\_split\_count\_begins:n { #4 #1 } } {#5}
655 { #4 #1 \end{#2} } {#3}
656 }
657 \cs_new:Npn \_math\_split\_count\_begins:n #1
658 { \int\_eval:n { 0 \_math\_split\_count\_begins:w #1 \begin \q\_nil } }
659 \cs_new:Npn \_math\_split\_count\_begins:w #1 \begin #2
660 { \quark\_if\_nil:nF {#2} { +1 \_math\_split\_count\_begins:w } }

```

```

661 \cs_new:Npn \__math_split_check_count_begins:nnnn #1 #2 #3 #4
662 {
663   \int_compare:nNnTF {#1} = {#2}
664   {
665     \exp_last_unbraced:Nf \__math_split_final_cleanup:nn
666     { \split:n { \__math_split_guard:n {#3} #4 } }
667   }
668   {
669     \exp_args:No \use_ii_i:nn
670     { \exp_after:wN { \int_value:w \int_eval:n { #2 + 1 } } }
671     { \__math_split_collect_one_end:w #4 \s_stop {#3} }
672   }
673 }
674 \cs_new:Npn \__math_split_final_cleanup:nn #1 #2
675 {
676   \exp:w \__math_split_final_cleanup:w #1
677   \__math_split_guard:n \q_nil \s_mark { }
678   {#2}
679 }
680 \cs_new:Npn \__math_split_final_cleanup:w #1 \__math_split_guard:n #2 #3 \s_mark #4
681 {
682   \quark_if_nil:nTF {#2}
683   { \exp_end: { #4 #1 } }
684   { \__math_split_final_cleanup:w #3 \s_mark { #4 #1 #2 } }
685 }
686 \NewDocumentCommand \splitn { mm +m }
687 {
688   \tl_set:Nf \l_tmpa_tl { \split:n {#3} }
689   \show \l_tmpa_tl
690   \exp_after:wN \__splitn_aux:nnNN \l_tmpa_tl #1 #2
691 }
692
693
694 \cs_new:Npn \split:n #1 {
695   \__math_split_at_nl_first:w #1 \q_nil \s_stop }
696
697 \cs_new:Npn \__math_split_at_nl:NN #1#2 {
698   \tl_set:Nf \l_tmpa_tl {
699     \exp_after:wN \__math_split_at_nl_first:w #1 \q_nil \s_stop }
700   \exp_after:wN \__math_split_at_nl_aux:nnNN \l_tmpa_tl #1 #2
701 }
702
703 \cs_new_protected:Npn \__math_split_at_nl_aux:nnNN #1 #2 #3 #4
704 {
705   \tl_gset:Nn #4 {#1}
706   \tl_gset:Nn #3 {#2}
707 }
708

```

(End of definition for .)

\maybestartnewformulatag

```

709
710 \newif\if@subformulas

```

```

711 \tl_new:N \result
712
713 \cs_new_protected:Npn\grabaformulapartandstart {
714   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
715   \typeout{====>first-result=\meaning\result}
716   \typeout{====>first-tmpmathcontent=\meaning\g__math_grabbed_math_tl}
717   \tl_if_empty:NTF \g__math_grabbed_math_tl
718   {
719     \typeout{====>formula~ has~ no~ subparts}
720     \global\@subformulasfalse
721   }
722   {
723     \typeout{====>formula~ has~ subparts}
724     \global\@subformulastrue
725     \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
726     \tagstructbegin{tag=Formula,

```

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

```

727 %       alt=\result,
728 %       title-o=\resulttitle
729 %     }
730 %   }
731 %   \tagmcbegin{}
732 % }
733
734 \cs_new_protected:Npn\grabaformulapartandmayberestart {
735   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
736   \typeout{====>result=\meaning\result}
737   \typeout{====>tmpmathcontent=\meaning\g__math_grabbed_math_tl}
738 %   \tl_if_empty:NTF \g__math_grabbed_math_tl
739 %   {
740 %     \typeout{====>tmpmathcontent=empty}
741 %   }
742 %   {
743 %     \typeout{====>tmpmathcontent=not-empty}
744 %     \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
745 %     \tagstructbegin{tag=Formula,
746 %       alt=\result,
747 %       title-o=\resulttitle
748 %     }
749 %   }
750 %   \tagmcbegin{}
751 % }

```

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

```

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

```

```

761     }
762   \fi
763 \fi
764 }

```

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

```

765 %\AddToHook{package/breqn/after}{
766 % \typeout{==>~ in~ hook}
767 % \math_register_halign_env:nn {dmath}{\}
768 % \math_register_halign_env:nn {dgroup*}{\}
769 %}

770 \ExplSyntaxOff

771 <@@=>

772 %

773 </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:	
\(108	\bool_if:NTF	72, 121, 151, 204, 239, 338, 358, 364, 374, 429, 449
\(<u>336</u>	\bool_new:N	11
\)	109	\bool_set_false:N	116, 380
\)	<u>336</u>	\bool_set_true:N	144, 164, 197, 217, 252, 377, 431, 451, 461, 466
\[118, 334	bool internal commands:	
\[4, 7, 11, 12, <u>354</u>	\l__math_collected_bool	4, 11, 116, 121, 144, 151, 164, 197, 204, 217, 239, 252, 338, 358, 364, 374, 377, 380, 429, 431, 449, 451, 461, 466
\]	575, 630, 636, 640, 646, 695, 699		
\]	74		
\]	119, 334		
\]	4, 7, 11, 12, <u>354</u>		
A		C	
\abovedisplayskip	590	clist commands:	
\AddToHook	7, 460, 765	\clist_map_inline:nn	457
\advance	602, 608	color commands:	
		\color_select:n	74
		\columnwidth	582
		\cr	603, 621, 627
		cs commands:	
		\cs_generate_variant:Nn	22
		\cs_gset_eq:NN	
			139, 140, 192, 193, 234, 235
		\cs_gset_protected:Npn	
			336, 348, 356, 362, 369, 464
		\cs_if_exist:NTF	145, 157, 198, 210, 245
		\cs_new:Npn	283, 310, 630, 639, 650, 651, 657, 659, 661, 674, 680, 694, 697
B			
\begin	122, 269, 359, 394, 635, 639, 646, 650, 658, 659		
\begingroup	278		
\belowdisplay	14		
\belowdisplayshortskip	88, 443		
\belowdisplayskip	14, 87, 442, 578		
block internal commands:			
__block_beginpar_vmode:	297		
__block_list_beginpar_vmode:	290		

<code>\cs_new_eq:NN</code>	463	<code>\ExpandArgs</code>	129, 142, 195, 237
<code>\cs_new_protected:Npn</code> . . .	14, 23, 29, 30, 32, 57, 67, 83, 94, 108, 118, 135, 188, 230, 319, 323, 703, 713, 734	<code>\ExplSyntaxOff</code>	770
<code>\cs_set_protected:Npn</code>	31, 178	<code>\ExplSyntaxOn</code>	8
cs internal commands:		F	
<code>__cs_generate_internal_variant:n</code>	181	<code>\fi</code>	421, 448, 491, 494, 545, 571, 575, 580, 584, 595, 601, 606, 607, 615, 619, 762, 763
<code>__cs_tmp:w</code>	178, 185	G	
<code>\csname</code>	276	<code>\global</code> 601, 602, 607, 608, 619, 620, 720, 724	
D		<code>\grabaformulapartandmayberestart</code> . . .	734, 760
<code>\DeclareDocumentCommand</code>	129	<code>\grabaformulapartandstart</code>	408, 713
<code>\DeclareMathEnviornment</code>	3	group commands:	
<code>\DeclareRobustCommand</code>	269	<code>\group_begin:</code>	180
<code>\def</code>	272, 273, 388, 414, 484, 548, 573, 574, 623, 752	<code>\group_end:</code>	183
<code>\displaylines</code>	3	<code>\group_insert_after:N</code>	91, 446
E		H	
<code>\edef</code>	274, 725, 744	<code>\hbox</code>	565, 569
<code>\egroup</code>	481, 486, 489	I	
<code>\else</code>	420, 492, 545, 566, 575, 583, 601, 619, 754	<code>\ifdim</code>	582
<code>\end</code>	105, 122, 365, 398, 651, 655	<code>\iffalse</code>	491
end internal commands:		<code>\ifnum</code>	491
<code>_math_tag_dollardollar_-</code> <code>display_end</code>	14	<code>\iftrue</code>	439, 548
<code>\endcsname</code>	276	<code>\ifvmode</code>	575
<code>\endequation</code>	325	<code>\ignorespaces</code>	588
<code>\endequation*</code>	325	<code>\ignorespacesafterend</code>	482, 495, 546
<code>\endgather</code>	479	int commands:	
<code>\ensuremath</code>	3, 369	<code>\int_compare:nNnTF</code>	663
<code>\eqalign</code>	3	<code>\int_eval:n</code>	658, 670
<code>\equation</code>	325	<code>\int_gincr:N</code>	71
<code>\equation*</code>	325	<code>\int_set:Nn</code>	89, 444
<code>\everydisplay</code>	13	<code>\int_use:N</code>	74
<code>\everymath</code>	12, 13	<code>\int_value:w</code>	670
exp commands:		<code>\intertext</code>	574
<code>\exp:w</code>	676	K	
<code>\exp_after:wN</code>	645, 670, 690, 699, 700	<code>\kern</code>	611
<code>\exp_args:Nf</code>	653	keys commands:	
<code>\exp_args:No</code>	181, 426, 436, 669	<code>\keys_define:nn</code>	125
<code>\exp_args:Noo</code>	294	<code>\keys_set:nn</code>	138, 191, 233
<code>\exp_end:</code>	683	L	
<code>\exp_last_unbraced:Nf</code>	665	<code>\lastskip</code>	100
<code>\exp_not:N</code>	144, 147, 148, 150, 151, 159, 160, 162, 164, 166, 167, 169, 170, 171, 197, 200, 201, 203, 204, 212, 213, 215, 217, 219, 220, 222, 223, 226, 239, 247, 248, 250, 252, 254, 255, 259, 394, 398	legacy commands:	
<code>\exp_not:n</code>	163, 172, 216, 251, 261, 262, 396	<code>\legacy_if:nTF</code>	16, 39, 288
		<code>\legacy_if_set_false:n</code>	289
		<code>\let</code>	548
		<code>\linewidth</code>	582, 583
		<code>\ltxlabmathdate</code>	4
		<code>\ltxlabmathversion</code>	5

M

math commands:

`\math_processor:n` 2, 30, 30
`\math_register_env:n` 2,
 135, 319, 327, 328, 330, 333, 455, 456
`\math_register_env:nn`
 2, 135, 135, 320, 322
`\math_register_halign_env:nn` ...
 188, 550, 551, 552, 553, 554,
 555, 556, 557, 558, 559, 560, 767, 768
`\math_register_odd_env:nn` 230

math internal commands:

`__math_amsmath_align@:nn` 469
`__math_amsmath_gather@:n` 469
`__math_amsmath_multline@:n` ... 469
`__math_env_end:`
 148, 160, 201, 213, 248, 323
`__math_env_forward:w` . 132, 323, 323
`__math_equation_begin:` 325
`__math_equation_end:` 325
`__math_equation_star_begin:` .. 325
`__math_equation_star_end:` ... 325
`__math_grab_dollar:w` ... 32, 32, 432
`__math_grab_dollardollar:w`
 54, 57, 452
`__math_grab_eqn:w` 118, 118, 360
`__math_grab_inline:w` . 108, 108, 345
`__math_m@th:` 463, 463, 467
`__math_process:nn`
 5, 14, 14, 22, 37, 84,
 113, 150, 162, 203, 215, 250, 324, 378
`__math_process_auxi:nn` ... 14, 19, 23
`__math_process_auxii:nn` 14, 27, 29, 31
`__math_split_at_nl:NN` . 697, 714, 735
`__math_split_at_nl_aux:nnNN` 700, 703
`__math_split_at_nl_first:w`
 630, 695, 699
`__math_split_check_count_-`
`begins:nnnn` 653, 661
`__math_split_chk_if_begin:ww` ...
 635, 639
`__math_split_cleanup_begin_q_-`
`nil:w` 646, 650
`__math_split_collect_one_end:w` .
 645, 651, 671
`__math_split_count_begins:n` 654, 657
`__math_split_count_begins:w` ...
 658, 659, 660
`__math_split_final_cleanup:nn` ..
 665, 674
`__math_split_final_cleanup:w` ...
 676, 680, 684
`__math_split_guard:n` . 666, 677, 680

`__math_tag_dollardollar_-`

`display:nn` 62, 83

`__math_tag_dollardollar_-`

`display_end:` 91, 94, 446

`\maybe_start_new_formula_tag`
 599, 617, 625, 709, 752

`\MaybeStop` 3

`\meaning` 715, 716, 736, 737

mode commands:

`\mode_if_math:TF` 340, 350, 371

`\mode_if_vertical:TF` 286

N

`\NewDocumentCommand` 321, 686

`\newif` 710

`\noalign` 576

`\noindent` 588

`\nonumber` 490, 595, 606

`\normalbaselines` 581

`\normalfont` 565, 569

O

`\omit` 610

P

`\par` 588

para commands:

`\para_raw_end:` 98, 312

`\parshape` 583

`\penalty` 102, 578, 590

`\postdisplaypenalty` . 14, 89, 102, 444, 578

`\predisdisplaypenalty` 590

`\ProvidesFile` 3

Q

quark commands:

`\q_nil` 635, 640, 650, 658, 677, 695, 699

`\quark_if_nil:nTF` .. 632, 642, 660, 682

R

`\RegisterMathEnvironment` 2, 3, 135

`\relax` 596

`\RenewDocumentEnvironment`
 142, 195, 237, 331, 334

`\renewenvironment` 474, 497, 502,
 507, 512, 517, 522, 527, 532, 537, 543

`\RequirePackage` 7, 10

`\result` .. 711, 714, 715, 727, 735, 736, 746

`\resulttitle` 725, 728, 744, 747

`\rlap` 74

S

scan commands:

`\s_mark` 635, 639, 677, 680, 684

<code>\s_stop</code>	630, 636, 640, 646, 651, 671, 695, 699
<code>\show</code>	689
<code>\ShowTagging</code>	96, 284, 285, 307, 311, 315, 416, 423
skip commands:	
<code>\skip_new:N</code>	55
<code>\skip_set:Nn</code>	87, 88, 442, 443
<code>\skip_vertical:n</code>	103
skip internal commands:	
<code>\l_math_tmpa_skip</code>	55, 100, 103
<code>\space</code>	4, 5, 395, 397, 399, 725, 744
split commands:	
<code>\split:n</code>	666, 688, 694
<code>\splitnl</code>	686
splitnl internal commands:	
<code>_splitnl_aux:nnNN</code>	690
str commands:	
<code>\str_if_eq:nnTF</code>	294
T	
tag commands:	
<code>\tag_if_active:TF</code>	68
<code>\tag_mc_begin:n</code>	73
<code>\tag_mc_begin_pop:n</code>	589
<code>\tag_mc_end:</code>	75
<code>\tag_mc_end_push:</code>	45, 70, 585
<code>\tag_stop:</code>	548, 580
<code>\tag_struct_end:</code>	77
tag internal commands:	
<code>\g__tag_para_end_int</code>	71, 74
<code>\l__tag_para_main_tag_tl</code>	294
<code>\l__tag_para_show_bool</code>	72
<code>\g__tag_struct_tag_tl</code>	292, 294
<code>\tagmcbegin</code>	49, 410, 412, 568, 570, 731, 750
<code>\tagmcend</code>	45, 70, 303, 417, 567, 570, 758
<code>\tagpdfparaOff</code>	97, 306
<code>\tagpdfparaOn</code>	99, 313, 587
<code>\tagpdfsetup</code>	586
<code>\tagstructbegin</code>	401, 410, 567, 726, 745
<code>\tagstructend</code>	419, 422, 570, 759
TeX and L ^A T _E X 2 _ε commands:	
<code>\badmath</code>	341, 352, 366
<code>\@currenvir</code>	273
<code>\@currenvline</code>	274
<code>\@doendpe</code>	105
<code>\@eha</code>	272
<code>\@empty</code>	575
<code>\@endpefalse</code>	279
<code>\@endpetrue</code>	316
<code>\@ensuredmath</code>	5, 375, 379
<code>\@eqnswtrue</code>	601, 619
<code>\@execute@begin@hook</code>	275
<code>\@ifundefined</code>	271
<code>\@ignorefalse</code>	277
<code>\@kernel@before@begindocument</code>	9, 325, 354, 472
<code>\@kernel@close@P</code>	67, 85, 302
<code>\@kernel@math@begin</code>	46, 86, 305, 388
<code>\@kernel@math@end</code>	48, 101, 314, 414
<code>\@kernel@math@registered@begin</code>	163, 216, 251, 283
<code>\@kernel@math@registered@end</code>	172, 262, 310
<code>\@latex@error</code>	272
<code>\@align</code>	613
<code>\@namedef</code>	562, 594, 605, 623
<code>\@one</code>	508, 513, 523, 528, 583, 602, 608
<code>\@subformulasfalse</code>	720
<code>\@subformulastrue</code>	724
<code>\@totalleftmargin</code>	583
<code>\@exp</code>	545
<code>\add@amps</code>	609
<code>\align@</code>	469
<code>\alignsep@</code>	611
<code>\black@</code>	481, 485
<code>\common@align@ending</code>	484, 500, 505, 510, 515, 520, 525, 530, 535, 540
<code>\gather@</code>	469
<code>\if@eqnsw</code>	607
<code>\if@subformulas</code>	418, 710, 753
<code>\ifingather@</code>	487
<code>\ifmeasuring@</code>	548, 564, 580, 754
<code>\ifst@rred</code>	595, 601, 606, 619
<code>\iftag@</code>	612
<code>\iftagsleft@</code>	545
<code>\intertext@</code>	573
<code>\lendmultline@</code>	545
<code>\lineht@</code>	620
<code>\m@ne</code>	523, 528, 533, 538
<code>\m@th</code>	3, 5, 14, 18, 42, 463, 565, 569
<code>\make@display@tag</code>	597, 613
<code>\math@cr</code>	481, 485
<code>\maxfields@</code>	609
<code>\measuring@true</code>	548
<code>\multline@</code>	469
<code>\on@line</code>	274
<code>\place@tag</code>	614
<code>\rendmultline@</code>	545
<code>\reserved@a</code>	272, 273, 280
<code>\restore@math@cr</code>	623, 628
<code>\restorealignstate@</code>	488
<code>\row@</code>	602, 608
<code>\setboxz@h</code>	613
<code>\split@tag</code>	15
<code>\st@rredfalse</code>	498, 508, 523, 533
<code>\st@rredtrue</code>	475, 503, 513, 518, 528, 538, 543

<code>\start@align</code>	498,	<code>\l_tmpa_tl</code>	688, 689, 690, 698, 700
	503, 508, 513, 518, 523, 528, 533, 538	tl internal commands:	
<code>\start@gather</code>	475	<code>\l__math_env_name_tl</code>	
<code>\start@multline</code>	543		7, 124, 130, 137, 190, 232
<code>\strut@</code>	613	<code>\g__math_grabbed_env_tl</code>	
<code>\tag@true</code>	607		4, 12, 25, 394, 398, 403, 725, 744
<code>\totwidth@</code>	481, 485	<code>\g__math_grabbed_math_tl</code> 4, 13, 26,	
<code>\tw@</code>	518, 533, 538		396, 714, 716, 717, 735, 737, 738, 756
<code>\z@</code>	498, 503, 620	<code>\tmpmathcontent</code> . .	385, 391, 402, 404, 405
tex commands:		<code>\tracingall</code>	166, 219, 254
<code>\tex_everydisplay:D</code>	436, 438	<code>\tracingnone</code>	171, 223
<code>\tex_everymath:D</code>	426, 428	<code>\typeout</code> 95, 121, 153, 156, 206, 209, 241,	
<code>\tex_the:D</code>	428, 438		244, 261, 285, 292, 296, 301, 307,
<code>\tiny</code>	74		311, 315, 389, 415, 441, 563, 715,
tl commands:			716, 719, 723, 736, 737, 740, 743, 766
<code>\tl_gput_right:Nn</code> . . .	9, 325, 354, 472	U	
<code>\tl_gset:Nn</code>	25, 26, 391, 705, 706	use commands:	
<code>\tl_if_blank:nTF</code>	35, 60, 111	<code>\use_ii_i:nn</code>	669
<code>\tl_if_empty:NTF</code>	717, 738, 756	<code>\UseHook</code>	270
<code>\tl_if_in:nnTF</code>	18, 42	V	
<code>\tl_new:N</code>	12, 13, 124, 385, 711	<code>\vbox</code>	579
<code>\tl_set:Nn</code>	137, 190, 232, 688, 698	<code>\vskip</code>	578, 590
<code>\tl_to_str:n</code>	182		
<code>\tl_trim_spaces_apply:nN</code>	19		