

The `latex-lab-mathtagging` code*

Frank Mittelbach, Joseph Wright, L^AT_EX Project

March 9, 2023

Abstract

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

Contents

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	6
3.7	Document commands	11
3.8	<code>\everymath</code> and <code>\everydisplay</code>	12
3.9	Modifying kernel environments	14
3.10	Modifying <code>amsmath</code>	14

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 \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.

- 3.

2.3 Other ToDos

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

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

3 The Implementation

1 `<@@=math>`

2 `<*kernel>`

3.1 File declaration

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

Change description here?

Temp loading ...

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

3.2 Setup

Loading `amsmath` is an absolute requirement: this avoids needing to have conditional definitions and deals with how to define `\[/\]` 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 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 definition for `\math_processor:n`. This function is documented on page 2.)

3.5 Content grabbing

`_math_grab_dollar:w`

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?

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

```

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

```

(End definition for `_math_grab_dollar:w`.)

`_math_grab_dollardollar:w`

And for the classical \TeX display structure.

```

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

```

```

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

```

(End definition for `_math_grab_dollardollar:w`.)

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

```

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

```

(End definition for `_math_grab_inline:w`.)

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

```

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

```

(End definition for `_math_grab_eqn:w`.)

3.6 Marking math environments

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

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

115 `\tl_new:N \l__math_env_name_tl`

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

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

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

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

```

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

```



```

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

```

```

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

```

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

`__math_env_forward:w`

```

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

```

(End definition for `_math_env_forward:w`.)

3.7 Document commands

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

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

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

```

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

```

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

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

```

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

```

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

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

```

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

```

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

why does `ensuremath` need handling at all?

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

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

```

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

```

(End 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.

```

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

```

```

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

```

```

439         \bool_set_true:N \l__math_collected_bool
440         \__math_grab_dollardollar:w
441     }
442 }

```

3.9 Modifying kernel environments

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

```

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

```

`__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!

```

451 \cs_new_eq:NN \__math_m@th: \m@th
452 \cs_gset_protected:Npn \m@th
453 {
454     \bool_set_true:N \l__math_collected_bool
455     \__math_m@th:
456 }

```

(End 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 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`.

```

457
458
459
460 \tl_gput_right:Nn \@kernel@before@begindocument {
461 %
462 \renewenvironment{gather*}{%
463     \start@gather\st@rredtrue
464 }
465 {%
466 % this redirection doesn't work if we alter "gather"!
467 % \endgather
468 % so replace it with its real meaning
469 \math@cr \black@\totwidth@ \egroup
470 $$$\ignorespacesafterend
471 }

```

```

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

```

```

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

```

Also for false?

```

536 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_stop:}
537 %
538 \math_register_halign_env:nn {align}{}
539 \math_register_halign_env:nn {align*}{}
540 \math_register_halign_env:nn {flalign}{}
541 \math_register_halign_env:nn {flalign*}{}
542 \math_register_halign_env:nn {gather}{}
543 \math_register_halign_env:nn {gather*}{}
544 \math_register_halign_env:nn {multline}{}
545 \math_register_halign_env:nn {multline*}{}
546 \math_register_halign_env:nn {xalignat}{}
547 \math_register_halign_env:nn {xalignat*}{}
548 \math_register_halign_env:nn {xxalignat}{}
549 %
550 \@namedef{maketag @ @ @} #1{%
551 % \typeout{--->maketag @ @ @}
552 \ifmeasuring@
553 \hbox{\m@th\normalfont#1}%
554 \else
555 \tagmcend \tagstructbegin{tag=Lbl}%
556 \tagmcbegin{tag=Lbl}%
557 \hbox{\m@th\normalfont#1}%
558 \tagmcend \tagstructend \tagmcbegin{}%
559 \fi
560 }
561 \def\intertext@{%
562 \def\intertext##1{%
563 \ifvmode\else\\\@empty\fi
564 \noalign{%
565 % we have to flip the sign on the skip because we flipped it on the outside
566 \penalty\postdisplaypenalty\vskip-\belowdisplayskip
567 \vbox{

```

Stop tagging when measuring:

```

568 \ifmeasuring@\tag_stop:\fi
569 \normalbaselines
570 \ifdim\linewidth=\columnwidth
571 \else \parshape\@ne \@totalleftmargin \linewidth
572 \fi

```

End the previous mc:

```

573 \tag_mc_end_push:

```

if we use 2 levels of formulas this would need changing

not true any longer

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

```

574         \tagpdfsetup{paratag=P}%
575         \tagpdfpara0n
576         \noindent\ignorespaces##1\par

Restart the MC

577         \tag_mc_begin_pop:n{}}}%
578         \penalty\predisplaypenalty\vskip\abovedisplayskip%
579     }%
580 }
581 }

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

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

611 \def\restore@math@cr{\@namedef{math@cr @ @ @}{
612 %
613     \maybestartnewformulatag
614 %
615     \cr}}
616 \restore@math@cr
617 }

```

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

```

618 \cs_new:Npn \_math_split_at_nl_first:w #1 \ \ #2 \ \ #3 \s_stop

```

```

619 {
620   \quark_if_nil:nTF {#2}
621   { {#1} { } }
622   {
623     \__math_split_chk_if_begin:ww #1 \begin \q_nil \s_mark
624     #2 \ \ #3 \s_stop
625   }
626 }
627 \cs_new:Npn \__math_split_chk_if_begin:ww #1 \begin #2 #3 \s_mark
628 #4 \ \ \q_nil \ \ \s_stop
629 {
630   \quark_if_nil:nTF {#2}
631   { {#1} {#4} }
632   {
633     \exp_after:wN \__math_split_collect_one_end:w
634     \__math_split_cleanup_begin_q_nil:w #1 \begin{#2} #3 \ \ #4 \s_stop
635     { } { 1 }
636   }
637 }
638 \cs_new:Npn \__math_split_cleanup_begin_q_nil:w #1 \begin \q_nil {#1}
639 \cs_new:Npn \__math_split_collect_one_end:w #1 \end #2 #3 \s_stop #4 #5
640 {
641   \exp_args:Nf \__math_split_check_count_begins:nnnn
642   { \__math_split_count_begins:n { #4 #1 } } {#5}
643   { #4 #1 \end{#2} } {#3}
644 }
645 \cs_new:Npn \__math_split_count_begins:n #1
646 { \int_eval:n { 0 \__math_split_count_begins:w #1 \begin \q_nil } }
647 \cs_new:Npn \__math_split_count_begins:w #1 \begin #2
648 { \quark_if_nil:nF {#2} { +1 \__math_split_count_begins:w } }
649 \cs_new:Npn \__math_split_check_count_begins:nnnn #1 #2 #3 #4
650 {
651   \int_compare:nNnTF {#1} = {#2}
652   {
653     \exp_last_unbraced:Nf \__math_split_final_cleanup:nn
654     { \split:n { \__math_split_guard:n {#3} #4 } }
655   }
656   {
657     \exp_args:No \use_ii_i:nn
658     { \exp_after:wN { \int_value:w \int_eval:n { #2 + 1 } } }
659     { \__math_split_collect_one_end:w #4 \s_stop {#3} }
660   }
661 }
662 \cs_new:Npn \__math_split_final_cleanup:nn #1 #2
663 {
664   \exp:w \__math_split_final_cleanup:w #1
665   \__math_split_guard:n \q_nil \s_mark { }
666   {#2}
667 }
668 \cs_new:Npn \__math_split_final_cleanup:w #1 \__math_split_guard:n #2 #3 \s_mark #4
669 {
670   \quark_if_nil:nTF {#2}
671   { \exp_end: { #4 #1 } }
672   { \__math_split_final_cleanup:w #3 \s_mark { #4 #1 #2 } }

```

```

673 }
674 \NewDocumentCommand \splitnl { mm +m }
675 {
676   \tl_set:Nf \l_tmpa_tl { \split:n {#3} }
677   \show \l_tmpa_tl
678   \exp_after:wN \__splitnl_aux:nnNN \l_tmpa_tl #1 #2
679 }
680
681
682 \cs_new:Npn \split:n #1 {
683   \__math_split_at_nl_first:w #1 \q_nil \s_stop }
684
685 \cs_new:Npn \__math_split_at_nl:NN #1#2 {
686   \tl_set:Nf \l_tmpa_tl {
687     \exp_after:wN \__math_split_at_nl_first:w #1 \q_nil \s_stop }
688   \exp_after:wN \__math_split_at_nl_aux:nnNN \l_tmpa_tl #1 #2
689 }
690
691 \cs_new_protected:Npn \__math_split_at_nl_aux:nnNN #1 #2 #3 #4
692 {
693   \tl_gset:Nn #4 {#1}
694   \tl_gset:Nn #3 {#2}
695 }
696

```

(End definition for .)

\maybestartnewformulatag

```

697
698 \newif\if@subformulas
699 \tl_new:N \result
700
701 \cs_new_protected:Npn \grabaformulapartandstart {
702   \__math_split_at_nl:NN \g__math_grabbed_math_tl \result
703   \typeout{====>first-result=\meaning\result}
704   \typeout{====>first-tmpmathcontent=\meaning\g__math_grabbed_math_tl}
705   \tl_if_empty:NTF \g__math_grabbed_math_tl
706   {
707     \typeout{====>formula~ has~ no~ subparts}
708     \global\@subformulasfalse
709   }
710   {
711     \typeout{====>formula~ has~ subparts}
712     \global\@subformulastrue
713     \edef\resulttitle{\g__math_grabbed_env_tl\space (part)}
714     \tagstructbegin{tag=Formula,
715 %           alt=\result,
716 %           title-o=\resulttitle
717     }
718   }
719   \tagmcbegin{}
720 }
721

```

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

```

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

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

740 \def\maybestartnewformulatag {
741 \if@subformulas
742 \ifmeasuring@\else
743 %
744 \tl_if_empty:NF \g__math_grabbed_math_tl
745 {
746   \tagmccend
747   \tagstructend
748   \grabaformulapartandmaybe restart
749 }
750 \fi
751 \fi
752 }

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

753 %\AddToHook{package/breqn/after}{
754 % \typeout{===>~ in~ hook}
755 % \math_register_halign_env:nn {dmath}{-}
756 % \math_register_halign_env:nn {dgroup*}{-}
757 %}

758 \ExplSyntaxOff
759 <@@=>
760 %
761 </kernel>

```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
$\backslash($	99
$\backslash)$	<u>327</u>
$\backslash[$	100
$\backslash]$	<u>327</u>
$\backslash[$	109, 325
$\backslash[$	4, 6, 11, 12, <u>346</u>
$\backslash\backslash$	563, 618, 624, 628, 634, 683, 687
$\backslash_$	65
$\backslash]$	110, 325
$\backslash]$	4, 6, 11, 12, <u>346</u>
A	
$\backslashabovedisplayskip$	578
\backslashAddToHook	7, 448, 753
\backslashadvance	590, 596
B	
\backslashbegin	113, 260, 351, 384, 623, 627, 634, 638, 646, 647
\backslashbegingroup	269
$\backslashbelowdisplayshortskip$	78, 431
$\backslashbelowdisplayskip$	77, 430, 566
block internal commands:	
$\backslash_block_beginpar_vmode$:	288
$\backslash_block_list_beginpar_vmode$: ..	281
bool commands:	
$\backslashbool_if:N\TF$	63, 112, 142, 195, 230, 329, 350, 356, 366, 418, 437
$\backslashbool_new:N$	11
$\backslashbool_set_false:N$	107, 372
$\backslashbool_set_true:N$...	135, 155, 188, 208, 243, 336, 369, 420, 439, 449, 454
bool internal commands:	
$\backslashl_math_collected_bool$ 4, 11, 107,	
112, 135, 142, 155, 188, 195, 208,	
230, 243, 329, 336, 350, 356, 366,	
369, 372, 418, 420, 437, 439, 449, 454	
C	
clist commands:	
$\backslashclist_map_inline:nn$	445
color commands:	
$\backslashcolor_select:n$	65
\backslashcolumnwidth	570
\backslashcr	591, 609, 615
cs commands:	
$\backslashcs_generate_variant:Nn$	22
$\backslashcs_gset_eq:NN$	130, 131, 183, 184, 225, 226
$\backslashcs_gset_protected:Npn$	327, 340, 348, 354, 361, 452
$\backslashcs_if_exist:N\TF$ 136, 148, 189, 201, 236	
$\backslashcs_new:Npn$ 274, 301, 618, 627, 638,	
639, 645, 647, 649, 662, 668, 682, 685	
$\backslashcs_new_eq:NN$	451
$\backslashcs_new_protected:Npn$...	14, 23, 29, 30, 32, 49, 60, 73, 84, 99, 109,
126, 179, 221, 310, 314, 691, 701, 722	
$\backslashcs_set_protected:Npn$	31, 169
cs internal commands:	
$\backslash_cs_generate_internal_variant:n$	172
$\backslash_cs_tmp:w$	169, 176
\backslashcsname	267
D	
$\backslashDeclareDocumentCommand$	120
$\backslashDeclareMathEnvironment$	3
$\backslashDeclareRobustCommand$	260
\backslashdef	263, 264, 380, 403, 472, 536, 561, 562, 611, 740
\backslashdisplaylines	3
E	
\backslashedef	265, 713, 732
\backslashegroup	469, 474, 477
\backslashelse	409, 480, 533, 554, 563, 571, 589, 607, 742
\backslashend	95, 113, 357, 388, 639, 643
\backslashendcsname	267
\backslashendequation	<u>316</u>
$\backslashendequation*$	<u>316</u>
\backslashendgather	467
\backslashensuremath	3, <u>361</u>
\backslasheqalign	3
\backslashequation	<u>316</u>
$\backslashequation*$	<u>316</u>
\backslasheverydisplay	12
\backslasheverymath	12
exp commands:	
$\backslashexp:w$	664
$\backslashexp_after:wN$.	633, 658, 678, 687, 688
$\backslashexp_args:Nf$	641
$\backslashexp_args:No$	172, 415, 424, 657
$\backslashexp_args:Noo$	285

<code>\exp_end:</code>	671	L	
<code>\exp_last_unbraced:Nf</code>	653	<code>\lastskip</code>	90
<code>\exp_not:N</code>	135, 138, 139, 141, 142, 150, 151, 153, 155, 157, 158, 160, 161, 162, 188, 191, 192, 194, 195, 203, 204, 206, 208, 210, 211, 213, 214, 217, 230, 238, 239, 241, 243, 245, 246, 250, 384, 388	legacy commands:	
<code>\exp_not:n</code>		<code>\legacy_if:nTF</code>	16, 279
....	154, 163, 207, 242, 252, 253, 386	<code>\legacy_if_set_false:n</code>	280
<code>\ExpandArgs</code>	120, 133, 186, 228	<code>\let</code>	536
<code>\ExplSyntaxOff</code>	758	<code>\linewidth</code>	570, 571
<code>\ExplSyntaxOn</code>	8	<code>\ltmathtaggingdate</code>	4
		<code>\ltmathtaggingversion</code>	5
F		M	
<code>\fi</code>	410, 436, 479, 482, 533, 559, 563, 568, 572, 583, 589, 594, 595, 603, 607, 750, 751	math commands:	
		<code>\math_processor:n</code>	2, 30, 30
		<code>\math_register_env:n</code>	2, 126, 310, 318, 319, 321, 324, 443, 444
		<code>\math_register_env:nn</code>	
		2, 126, 126, 311, 313
		<code>\math_register_halign_env:nn</code> ...	
		179, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 755, 756
		<code>\math_register_odd_env:nn</code>	221
G		math internal commands:	
<code>\global</code> 589, 590, 595, 596, 607, 608, 708, 712		<code>__math_amsmath_align@:nn</code>	457
<code>\grabaformulapartandmayberestart</code> ...		<code>__math_amsmath_gather@:n</code>	457
.....	722, 748	<code>__math_amsmath_multline@:n</code> ...	457
<code>\grabaformulapartandstart</code>	397, 701	<code>__math_env_end:</code>	
group commands:		139, 151, 192, 204, 239, 314
<code>\group_begin:</code>	171	<code>__math_env_forward:w</code> .	123, 314, 314
<code>\group_end:</code>	174	<code>__math_equation_begin:</code>	316
<code>\group_insert_after:N</code>	81, 434	<code>__math_equation_end:</code>	316
		<code>__math_equation_star_begin:</code> ..	316
H		<code>__math_equation_star_end:</code>	316
<code>\hbox</code>	553, 557	<code>__math_grab_dollar:w</code> ...	32, 32, 421
		<code>__math_grab_dollardollar:w</code>	
		46, 49, 440
I		<code>__math_grab_eqn:w</code> ...	109, 109, 352
<code>\ifdim</code>	570	<code>__math_grab_inline:w</code> ...	99, 99, 337
<code>\iffalse</code>	479	<code>__math_m@th:</code>	451, 451, 455
<code>\ifnum</code>	479	<code>__math_process:nn</code> 14, 14, 22, 37, 74, 104, 141, 153, 194, 206, 241, 315, 370	
<code>\iftrue</code>	427, 536	<code>__math_process_auxi:nn</code> ...	14, 19, 23
<code>\ifvmode</code>	563	<code>__math_process_auxii:nn</code> 14, 27, 29, 31	
<code>\ignorespaces</code>	576	<code>__math_split_at_nl:NN</code> .	685, 702, 723
<code>\ignorespacesafterend</code>	470, 483, 534	<code>__math_split_at_nl_aux:nnNN</code>	688, 691
int commands:		<code>__math_split_at_nl_first:w</code>	
<code>\int_compare:nNnTF</code>	651	618, 683, 687
<code>\int_eval:n</code>	646, 658	<code>__math_split_check_count_-</code>	
<code>\int_gincr:N</code>	62	<code>begins:nnnn</code>	641, 649
<code>\int_set:Nn</code>	79, 432	<code>__math_split_chk_if_begin:ww</code> ...	
<code>\int_use:N</code>	65	623, 627
<code>\int_value:w</code>	658	<code>__math_split_cleanup_begin_q_-</code>	
<code>\intertext</code>	562	<code>nil:w</code>	634, 638
		<code>__math_split_collect_one_end:w</code> .	
		633, 639, 659
K			
<code>\kern</code>	599		
keys commands:			
<code>\keys_define:nn</code>	116		
<code>\keys_set:nn</code>	129, 182, 224		

