

The `eqnlines` Package

Source Code Documentation

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland
`nbeisert@itp.phys.ethz.ch`

2026/01/13, v0.12.2

<https://ctan.org/pkg/eqnlines>
<https://github.com/nbeisert/latex-pkg-nb>

Abstract

`eqnlines` is a $\text{\LaTeX}$ 2 ϵ package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of $\text{\LaTeX}$ and the `amsmath` package with many options for convenient adjustment of the intended layout. In particular, the package adds flexible schemes for numbering, horizontal alignment and semi-automatic punctuation, and it improves upon the horizontal and vertical spacing options. The extensions can be used and adjusted through optional arguments and modifiers to the equation environments as well as global settings.

Contents

1	Implementation	3
2	General Support	3
2.1	Development Messages	3
2.2	Supporting Definitions	4
2.3	Dollardollar Abstraction	4
2.4	Look-Ahead in Alignment	4
2.5	Error Messages	7
2.6	<code>amsmath</code> Integration	8
2.7	PDF Tagging Support	8
2.8	Key-Value Processing	10
3	Parameters and Registers	12
3.1	Parameters	12
3.2	Registers	15
3.3	Hooks	17
4	Features	18
4.1	Punctuation	18
4.2	Math Classes at Alignment	20

4.3	Framed Cells	21
4.4	Single-Line Composition	21
4.5	Alternative Content Description	22
5	Equation Numbering	22
5.1	Supporting Definitions	22
5.2	Schemes	23
5.3	Interface	27
5.4	Integration	32
5.5	Positioning	34
5.6	Component Display	38
5.7	Tag Composition	41
6	Subequation Numbering	44
6.1	Definitions	44
6.2	Environment	45
6.3	Subequation Scheme	46
7	Display Equations Support	47
7.1	Display Breaks	47
7.2	Explicit Vertical Space	48
7.3	Default Vertical Spacing	50
7.4	Entry and Exit	51
7.5	Stack	56
8	Multi-Line Support	57
8.1	Measure Support	57
8.2	Line Breaks	59
8.3	Intertext	60
8.4	Line Marks	62
9	Column Placement	63
9.1	Supporting Definitions	63
9.2	Shape Schemes	64
9.3	Width Data	67
9.4	Best Line Selection	71
9.5	Tag Margin	72
9.6	Single Column	72
9.7	Multiple Columns	74
10	Single Column Arrangement	79
10.1	Supporting Definitions	79
10.2	Arrangement Methods	80
10.3	Central Alignment	81
10.4	Left Alignment	83
10.5	Right Alignment	84
11	Equations Box Environment	85
11.1	Line Breaks	85
11.2	Stacked Mode	86
11.3	Aligned Mode	87
11.4	Main	88
11.5	Environment	90
12	Single-Line Equation	91

12.1 Native Mode	91
12.2 Print	91
13 Multi-Line with Single Column	93
13.1 Measure	93
13.2 Column Placement	94
13.3 Print	95
14 Multi-Line with Multiple Columns	96
14.1 Support	96
14.2 Transpose	97
14.3 Measure	99
14.4 Columns Placement	101
14.5 Print	102
15 Interface	105
15.1 Scanning the Equation Body	105
15.2 Options Processing	108
15.3 Single-Line Main	110
15.4 Multi-Line Main	111
15.5 Equations Environment	113
16 Options	116
16.1 Selection Tools	116
16.2 Options Declarations	116
16.3 Parameter Presets	127
16.4 Component Selection	129
16.5 Global and Package Options	138

1 Implementation

This appendix documents the implementation for the various components of the eqnlines package.

The code for the package is based on the `amsmath` package, see the reference manual for details. It was forked at version v2.17t dated 2024/11/05. Most of the code was substantially redesigned (macros renamed, reshuffled, enhanced), but many of the underlying mechanisms were preserved. The documentation thus contains excerpts from the `amsmath` package documentation explaining some details of the implementation.

Please note that the documentation is completed only for few sections in the present version. Various open issues are remarked.

2 General Support

In the following we describe general purpose supporting routines.

2.1 Development Messages

The package offers a version `eqnlines-dev` for development and debugging purposes. It outputs extra information on the current location within the code in order to track

progress. The extra lines for the development version are indicated as ‘(dev)’ in the implementation documentation:

```

1 (dev)\def\eql@dev#1{\PackageInfo{eqnlines-dev}{#1}}
2 (dev)\def\eql@dev@start#1{\eql@dev{starting \string#1}}
3 (dev)\def\eql@dev@enter#1{\eql@dev{entering \string#1}}
4 (dev)\def\eql@dev@leave#1{\eql@dev{ leaving \string#1}}
5 (dev)\def\eql@dev@enterenv{\eql@dev{entering \@currenvir}}
6 (dev)\def\eql@dev@leaveenv{\eql@dev{ leaving \@currenvir}}
7 (dev)\def\eql@dev@in#1#2{\eql@dev{ \space within \string#1 #2}}
```

2.2 Supporting Definitions

`\eql@false` (*bool*) Rather than the standard L^AT_EX scheme of `\xxxfalse`, `\xxxtrue` and `\ifxxx` for boolean variables *xxx*, we use a scheme where `\xxx` is either undefined or defined (to an empty macro) and is tested against by the ϵ -T_EX conditional `\ifdefined\xxx`. In order to make the scheme more tangible, we define the two expected values for boolean variables:

```

8 \let\eql@false\@undefined
9 \let\eql@true\@empty
```

TODO: describe

```

10 \def\eql@append#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
11 \def\eql@appendexpand#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
12 \def\eql@appendmacro#1#2{\eql@appendexpand#1{\unexpanded\expandafter{#2}}}
13 \def\eql@letcs#1{\expandafter\let\csname#1\endcsname}
```

2.3 Dollardollar Abstraction

`\dollar@begin` As of 2025 L^AT_EX defines `\dollar@begin` and `\dollar@end` to represent (and adjust) the beginning and end of bare T_EX display equations (`‘$’`). For the time being, we make sure to revert to `‘$’` if these macros are not yet available:

```

14 \ifdefined\dollar@begin
15   \def\eql@dollar@begin{\dollar@begin}
16   \def\eql@dollar@end{\dollar@end}
17 \else
18   \def\eql@dollar@begin{$$}
19   \def\eql@dollar@end{$$}
20 \fi
```

2.4 Look-Ahead in Alignment

Scanning for optional arguments [...] or modifiers such as ‘*’ using the L^AT_EX `\@ifnextchar` mechanism has two challenges within aligned equations: a square bracket or star may well be part of the intended mathematical expression and the look-ahead could trip upon an alignment character ‘&’ which inadvertently triggers to enter the next alignment column.

`\eql@ifnextchar@loose` To address the first challenge, we can force the special characters to follow immediately the macro invocation. For clarity, we copy L^AT_EX’s original `\@ifnextchar` in `\kernel@ifnextchar` which skips over spaces as `\eql@ifnextchar@loose`. We replicate the `amsgen` version `\new@ifnextchar` that does not skip over spaces as `\eql@ifnextchar@loose`. The space before #1 allows to look-ahead for spaces as well:

```

21 \let\eq@ifnextchar@loose\kernel@ifnextchar
22 \long\def\eq@ifnextchar@tight#1#2#3{%
23   \let\reserved@d= #1%
24   \def\reserved@a{#2}%
25   \def\reserved@b{#3}%
26   \futurelet\@let@token\eq@ifnch@tight
27 }
28 \def\eq@ifnch@tight{%
29   \ifx\@let@token\reserved@d
30     \let\reserved@b\reserved@a
31   \fi
32   \reserved@b
33 }

```

`\eq@atxi` Capture ‘@’ as a character (catcode 12) rather than a letter (catcode 11) as `\eq@atxii` so `\eq@atxii` that we can look-ahead for ‘@’ with both `\makeatother` and `\makeatletter` modes:

```

34 \let\eq@atxi=@
35 \begingroup
36   \makeatother
37   \let\tmp=@%
38   \makeatletter
39   \global\let\eq@atxii\tmp
40 \endgroup

```

`\eq@ifnextgobble@...` We introduce a collection of look-ahead macros which do or do not skip over spaces. The `\eq@ifstar@...` macros `\eq@ifstar@...` and `\eq@testopt@...` replicate the L^AT_EX counterparts `\eq@testopt@...` `\@ifstar` and `\@testopt`. The macros `\eq@ifnextgobble@...` work like `\@ifnextchar`, `\eq@teststaropt@...` but also gobble the specific character if found; one might define `\eq@ifstar@...` as `\eq@ifnextgobble@...*`. The macros `\eq@teststaropt@...` tests for combinations of ‘*’ and optional arguments [...]:

```

41 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
42 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
43 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
44 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
45 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{#2}}
46 \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}
47 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{#2}}
48 \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}
49 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[#{1}{#1}]{#2}}
50 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[#{1}{#1}]{#2}}
51 \long\def\eq@teststaropt@loose#1#2#3{%
52   \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}
53 \long\def\eq@teststaropt@tight#1#2#3{%
54   \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}
55 \long\def\eq@teststaroropt@loose#1#2#3{%
56   \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}
57 \long\def\eq@teststaroropt@tight#1#2#3{%
58   \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}
59 \long\def\eq@gobbleopt[#1]{#1}{}
60 \long\def\eq@gobbleoptone[#1]#2{#1}

```

TODO: describe

```

61 \def\eq@testopt@default{\eq@testopt@default}

```

TODO: describe

```

62 \def\eql@parseopt#1#2{%
63   \def\eql@parseopt@case{#1}%
64   \def\eql@parseopt@end{#2}%
65   \eql@parseopt@peek
66 }
67 \def\eql@parseopt@peek{%
68   \futurelet\eql@parseopt@token\eql@parseopt@select
69 }
70 \def\eql@parseopt@select{%
71   \let\eql@parseopt@next\eql@parseopt@other
72   \ifx\eql@parseopt@token\@sptoken
73     \let\eql@parseopt@next\eql@parseopt@end
74   \fi
75   \eql@parseopt@case
76   \eql@parseopt@next
77 }
78 \def\eql@parseopt@other{\eql@parseopt@warn\eql@parseopt@end}
79 \let\eql@parseopt@warn\@empty
80 \def\eql@parseopt@gobble#1{\eql@parseopt@peek}

```

`\eql@spbggroup` The second challenge is addressed by enclosing the look-ahead in spurious groups¹ which
`\eql@speggroup` protect against triggering ‘&’. The macros `\eql@spbggroup` and `\eql@speggroup` open and
`\eql@srbgroup` close a spurious group. For some reason, the look-ahead mechanism requires further
`\eql@sregroup` protections by inserting `\relax` at the beginning and by resetting `\@let@token` at the end.
These adjustments are included in the macros `\eql@srbgroup` and `\ers@speggroup`:

```

81 \def\eql@spbggroup{\iffalse{\fi\ifnum0='}\fi}
82 \def\eql@speggroup{\ifnum0='{ \fi\iffalse}\fi}
83 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
84 \def\eql@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

`\eql@ampprotect` The macros `\eql@ampprotect` and `\eql@ampprotecttwo` inject the opening and closing of
`\eql@ampprotecttwo` spurious groups into the look-ahead mechanism:

```

85 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1\eql@sregroup#2}}
86 \long\def\eql@ampprotecttwo#1#2#3{%
87   \eql@srbgroup#1\eql@sregroup#2}{\eql@sregroup#3}}

```

`...@ampsafe` We introduce a collection of ‘&’-safe look-ahead macros:

```

88 \def\eql@ifnextchar@loose@ampsafe#1{%
89   \eql@ampprotecttwo{\eql@ifnextchar@loose#1}}
90 \def\eql@ifnextchar@tight@ampsafe#1{%
91   \eql@ampprotecttwo{\eql@ifnextchar@tight#1}}
92 \def\eql@ifstar@loose@ampsafe{\eql@ampprotecttwo\eql@ifstar@loose}
93 \def\eql@ifstar@tight@ampsafe{\eql@ampprotecttwo\eql@ifstar@tight}
94 \def\eql@testopt@loose@ampsafe{\eql@ampprotect\eql@testopt@loose}
95 \def\eql@testopt@tight@ampsafe{\eql@ampprotect\eql@testopt@tight}
96 \def\eql@teststaropt@loose@ampsafe{\eql@ampprotecttwo\eql@teststaropt@loose}
97 \long\def\eql@teststaropt@tight@ampsafe{%
98   \eql@ampprotecttwo\eql@teststaropt@tight}

```

`\eql@amproof` We may want to replace L^AT_EX’s definitions `\@ifnextchar`, `\@ifstar` and `\@testopt` to
`\eql@amprevert` respect ‘&’ characters within aligned equations. This might make unrelated definitions with

¹See <https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3040>,
<https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=amslatex/1834> and
<https://tex.stackexchange.com/questions/9897/showcase-of-brace-tricks-egroup-iffalse-fi-etc>.

optional arguments and starred variants more robust in this context. The macro `\eql@amproof` overwrites the original definitions, and `\eql@amprevert` reverts the changes:

```

99 \let\eql@ifnextchar@org\ifnextchar
100 \let\eql@ifstar@org\ifstar
101 \let\eql@testopt@org\@testopt
102 \def\eql@amprevert{%
103   \let\@ifnextchar\eql@ifnextchar@org
104   \let\@testopt\eql@testopt@org
105   \let\@ifstar\eql@ifstar@org
106 }
107 \def\eql@amproof{%
108   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
109   \let\@testopt\eql@testopt@loose@ampsafe
110   \let\@ifstar\eql@ifstar@loose@ampsafe
111 }

```

2.5 Error Messages

`\eql@error` Main error and warning message function for the package:
`\eql@warning`

```

112 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
113 \def\eql@warning{\PackageWarning{eqnlines}}

```

`\eql@error@mathmode` Error messages concerning math mode:

```

114 \def\eql@warn@here#1{\eql@warning{\string#1 not allowed outside equations}}
115 \def\eql@error@mathmode#1{\eql@error{#1 allowed only in paragraph mode}}

```

`\eql@warn@label@unused` Warning messages concerning unused and multiply declared labels and tags:

```

\eql@warn@label@multiple
\eql@warn@tag@unused
\eql@warn@tag@multiple
\eql@warn@name@unused
\eql@warn@name@multiple
\eql@warn@ref@unused
\eql@warn@ref@multiple
116 \def\eql@warn@tags@unused#1#2{\eql@warning{Unused equation #1:
117   #2 will be lost}}
118 \def\eql@warn@tags@multiple#1#2#3{\eql@warning{Multiple equation #1:
119   previous #2 will be lost#3}}
120 \def\eql@warn@label@unused{\eql@warn@tags@unused{\string\label}}
121   {label '\eql@tags@label'}}
122 \def\eql@warn@label@multiple#1{\eql@warn@tags@multiple{\string\label's}
123   {label '\eql@tags@label'}}{ and replaced by '#1'}}
124 \def\eql@warn@name@unused{\eql@warn@tags@unused{label name}
125   {name declaration}}
126 \def\eql@warn@name@multiple{\eql@warn@tags@multiple{label names}
127   {name declaration}{}}
128 \def\eql@warn@tag@unused{\eql@warn@tags@unused{\string>tag}
129   {tag declaration}}
130 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string>tag's}
131   {tag declaration will be lost}{}}
132 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}
133   {tag label declaration}}
134 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}
135   {tag label declaration}{}}

136 \def\eql@warn@parseopt{%
137   \eql@warning{Unknown modifier token: starting math content}}
138 \def\eql@warn@parseopt@verbose{%
139   \eql@warning{Unknown modifier token: \meaning\eql@parseopt@token}}

```

2.6 amsmath Integration

`\eql@amsmath@after` We need to overwrite certain macros from `amsmath`. The method `\eql@amsmath@after` executes argument #1 after loading `amsmath` is loaded. It also runs the code if `amsmath` has already been loaded. Furthermore, loading `amsmath` requires certain macros to be undefined. To this end `\eql@amsmath@before` will execute argument #1 before any future loading of `amsmath`. `\eql@amsmath@undefine` undefines a macro in this way and `\eql@amsmath@let` overwrites a macro of `\amsmath/`:

```
140 \def\eql@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
141 \def\eql@amsmath@before#1{%
142   \ifpackageloaded{amsmath}{\AddToHook{package/amsmath/before}{#1}}
143 \def\eql@amsmath@undefine#1{\eql@amsmath@before{\let#1\undefined}}
144 \def\eql@amsmath@let#1#2{\eql@amsmath@undefine#1\let#1#2}
```

TODO: temporary fix for development stages

```
145 \ifpackageloaded{amsmath}{%
146   \DeclareHookRule{package/amsmath/after}
147     {eqnlines}{after}{latex-lab-testphase-math}}
```

2.7 PDF Tagging Support

`\eql@tagging@...` Proper PDF tagging² support requires a L^AT_EX version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L^AT_EX versions around 2020: **TODO:** adjust to further developments

```
148 \let\eql@tagging@on\eql@false
149 \IfFormatAtLeastTF{2025-06-01}{%
150   \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}
151 \ifdefined\eql@tagging@on
152   \def\eql@tagging@mathsave{%
153     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
154   \def\eql@tagging@mathaddlast{%
155     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
156   \def\eql@tagging@tagbegin{%
157     \UseTaggingSocket{math/display/tag/begin}}
158   \def\eql@tagging@tagend{%
159     \UseTaggingSocket{math/display/tag/end}}
160   \def\eql@tagging@tagsave{%
161     \UseTaggingSocket{math/luamml/mtable/tag/save}}
162   \def\eql@tagging@tagaddbox{%
163     \setbox\z@\copy\eql@tagbox%
164     \UseTaggingSocket{math/luamml/mtable/tag/set}}
165   \def\eql@tagging@tablesaveinner{%
166     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
167   \def\eql@tagging@tableaddinner{%
168     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
169   \def\eql@tagging@tablesavelines{%
170     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
171   \def\eql@tagging@tablesavealign{%
172     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
173   \def\eql@tagging@alignleft{%
174     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
175   \def\eql@tagging@aligncenter{%
176     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
```

²see <https://latex3.github.io/tagging-project/>

```

177 \def\eql@tagging@alignright{%
178   \UseTaggingSocket{math/luamml/mtable/aligncol}{right}}

```

We need to get hold of the equation body in all cases so that we can feed it into the tagging mechanism:

```

179 \let\eql@single@doscan\eql@true
180 \let\eql@scan@body\eql@scan@body@rescan

```

`\eql@tagging@start` We need to activate tagging for display equations for environments and for enclosures
`\eql@tagging@end` `\[...]` and `\<...>`. The tagging interface registration macro `\RegisterMathEnvironment` will work only partially for our cases, hence we replicate code from `\math_register_halign_env:nn`. Make sure collection is not yet active (`\l__math_collected_bool`). Then feed collected environment name, options and body into `\__math_process:nn`. Indicate the start of a display equation:

```

181 \def\eql@tagging@start{%
182   \csname bool_if:N\expandafter\endcsname
183   \csname l__math_collected_bool\endcsname{%
184     \edef\eql@tmp{{\@currentenv}{\unexpanded\expandafter{\eql@tagging@opt}}}%
185     \the\eql@scan@reg@}%
186   \csname __math_process:nn\expandafter\endcsname\eql@tmp
187   \@kernel@math@registered@begin
188   \csname bool_set_true:N\expandafter\endcsname
189   \csname l__math_collected_bool\endcsname
190   }%
191 }
192 \def\eql@tagging@end{}
193 \def\eql@tagging@register@luamml#1{%
194   \AddToHook{package/luamml/after}{%
195     \eql@letcs{c__luamml_label_#1_tl}{\@empty}}
196 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}

```

When tagging is deactivated, provide empty definitions:

```

197 \else
198   \let\eql@tagging@mathsave\@empty
199   \let\eql@tagging@mathaddlast\@empty
200   \let\eql@tagging@tagbegin\@empty
201   \let\eql@tagging@tagend\@empty
202   \let\eql@tagging@tagsave\@empty
203   \let\eql@tagging@tagaddbox\@empty
204   \let\eql@tagging@tablesaveinner\@empty
205   \let\eql@tagging@tableaddinner\@empty
206   \let\eql@tagging@tablesavealign\@empty
207   \let\eql@tagging@alignleft\@empty
208   \let\eql@tagging@aligncenter\@empty
209   \let\eql@tagging@alignright\@empty
210   \let\eql@tagging@start\@empty
211   \let\eql@tagging@end\@empty
212   \let\eql@tagging@register@luamml\@gobble
213   \let\eql@tagging@register@env\@gobble
214 \fi
215 \fi

```

2.8 Key-Value Processing

The package uses the `keyval` mechanism to parse key-value pairs to specify adjustments to the behaviour of the equations environments:

```
216 \RequirePackage{keyval}
```

Value Selection.

`\eqld@decide@select` Some parameter values take values in a given set, e.g. `true` vs. `false` or `left` vs. `right`. The macro `\eqld@decide@select` is a general purpose selector. Arguments `#1` and `#2` describe the category and key which are used only towards error messages. Argument `#3` contains the value and argument `#4` is a list of values and corresponding actions in the format

$$\{\{\{val1a, val1b, \dots\}act1\}, \{\{val2a, val2b, \dots\}act2\}, \dots\}.$$

The (single) value `\relax` matches everything (can be used for handling generic values after specific ones). If no corresponding value is found in the list, an error message is invoked. Single expansion is applied to the list of values:

```
217 \def\eqld@decide@relax{\eqld@tmpb:=\relax}
218 \def\eqld@decide@select#1#2#3#4{%
219   \def\eqld@tmpa{#3}%
220   \let\eqld@tmpd\@undefined
221   \@for\eqld@tmpc:=#4\do{%
222     \ifdefined\eqld@tmpd\else
223       \edef\eqld@tmpb{\noexpand\eqld@tmpb:=\expandafter\@firstoftwo\eqld@tmpc}%
224       \ifx\eqld@tmpb\eqld@decide@relax
225         \def\eqld@tmpa{\relax}%
226       \fi
227       \expandafter\@for\eqld@tmpb\do{%
228         \ifx\eqld@tmpa\eqld@tmpb
229           \edef\eqld@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
230             \expandafter\@secondoftwo\eqld@tmpc}}%
231         \fi
232       }%
233     \fi
234   }%
235   \ifdefined\eqld@tmpd
236     \eqld@tmpd
237   \else
238     \eqld@error{undefined value '3' for option '#2' of '#1'}%
239   \fi
240 }
```

Decide between `true` and `false` or related pairs of values:

```
241 \def\eqld@decide@true{on,true,yes,enabled}
242 \def\eqld@decide@false{off,false,no,disabled}
```

`\eqld@decide@if`

```
243 \def\eqld@decide@if#1#2#3#4#5{%
244   \eqld@decide@select{#1}{#2}{#3}{%
245     {\eqld@decide@true{#4}},%
246     {\eqld@decide@false{#5}}}%
247 }
```

`\eqld@decide@bool` Store a boolean value into a conditional register:

```

247 \def\eql@decide@bool#1#2#3#4{%
248   \eql@decide@if{#1}{#2}{#3}{\let#4\eql@true}{\let#4\eql@false}}

```

Key Declaration.

`\eql@define@key` For convenience, we define a wrapper for `keyval`'s `\define@key` which accepts lists of categories and keys. We prepend the prefix `eql@` to all our categories so that we can hide it from the user in error messages:

```

249 \def\eql@define@key#1#2{%
250   \eql@ifnextchar@loose[%
251     {\eql@definekey@opt{#1}{#2}}%
252     {\eql@definekey@noopt{#1}{#2}}%
253 }
254 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{#3}}
255 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{[#3]{#4}}}
256 \def\eql@definekey@for#1#2#3{%
257   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}}%
258   \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
259   \expandafter\@for\eql@for@vara\do{%
260     \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
261     \expandafter\@for\eql@for@varb\do{%
262       \edef\eql@for@call##1{%
263         \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
264       \eql@for@call{##1}%
265     }%
266   }%
267 }

```

`\eql@setkeys` Our wrapper of `keyval`'s `\setkeys` prepends the prefix `eql@` to the category, and it expands the list argument once:

```

268 \def\eql@setkeys#1#2{%
269   \def\eql@tmp{\setkeys{eql@#1}}%
270   \expandafter\eql@tmp\expandafter{#2}%
271 }

```

Options and Control Interface.

`\eql@nextopt` It can be convenient to add arguments to the following equations environment, e.g.
`\eql@nextopt@process` towards defining modifier macros:

```

272 \let\eql@nextopt\@empty
273 \def\eql@nextopt@process#1{%
274   (dev)\eql@dev@start\eql@nextopt@process
275   \eql@setkeys{#1}\eql@nextopt
276   \let\eql@tagging@opt\eql@nextopt
277   \global\let\eql@nextopt\@empty
278 }

```

`\eqnaddopt`

```

279 \newcommand{\eqnaddopt}[1]{%
280   \ifx\eql@nextopt\@empty
281     \eql@append\eql@nextopt{#1}%
282   \else
283     \eql@append\eql@nextopt{, #1}%

```

```

284 \fi
285 }

```

`\eqnlineset` Process global configuration options including the package options:

```

286 \newcommand{\eqnlineset}[1]{%
287 (dev)\eql@dev@start\eqnlineset
288 \eql@setkeys{setup}{#1}%
289 \ignorespaces
290 }

```

`\eql@control@default`

```

291 \protected\def\eql@control@default{%
292 \eql@warn@here\eqncontrol
293 \@gobble
294 }
295 \let\eqncontrol\eql@control@default

```

`\eqncontrol` Macro for general-purpose control within equations using key-value pairs:

```

296 \newcommand{\eql@control}[1]{%
297 \relax
298 \eql@setkeys{control}{#1}%
299 \ignorespaces
300 }

```

3 Parameters and Registers

In the following, we collect parameter and register definitions.

3.1 Parameters

TODO: describe

TODO: maybe sort parameters into sections **TODO:** or sort parameters in sections here

`\eql@tagsleft` (*bool*) The boolean parameter `\eql@tagsleft` specifies whether the tags are placed at the left or right margin:

```

301 \let\eql@tagsleft\eql@false

```

`\eql@layoutleft` (*bool*) The boolean parameter `\eql@layoutleft` specifies whether to use left or central alignment layout:

```

302 \let\eql@layoutleft\eql@false

```

`\eql@layoutleftmargin` The default width of the left margin in left alignment layout is specified by `\eql@layoutleftmargin`. It may be pushed down to `\eql@layoutleftmarginmin` and up to `\eql@layoutleftmarginmax`:

```

303 \def\eql@layoutleftmargin{\leftmargini}
304 \def\eql@layoutleftmarginmax{.5\maxdimen}
305 \def\eql@layoutleftmarginmin{\z@}

```

`\eql@tagmargin@` (*dimen*) The intended margin width for tags in central alignment layout is stored in `\eql@tagmargin@` which is sourced by `\eql@tagmargin@val`. An undefined `\eql@tagmargin@val` will compute the margin width as the maximum width of tags (without separation). `\eql@tagmargin@ratio@` describes the maximum ratio of lines with tags to total number of lines for which `\eql@tagmargin@` is set to zero: **TODO:** threshold

```
306 \newdimen\eql@tagmargin@
307 \let\eql@tagmargin@val\@undefined
308 \newdimen\eql@tagmargin@ratio@
309 \eql@tagmargin@ratio@\p@
310 \def\eql@tagmargin@threshold{0.5}
```

`\eql@indent@` (*dimen*) The currently selected indentation width is specified by `\eql@indent@`. This dimension register is set to the macro `\eql@indent@val` when entering the equation environments:

```
311 \newdimen\eql@indent@
312 \def\eql@indent@val{2em}
```

`\eql@paddingleft@` (*dimen*) The padding of an equation (column) is specified by `\eql@paddingleft@` and `\eql@paddingright@`. These dimension registers are set to the macros `\eql@paddingleft@val` and `\eql@paddingright@val`, respectively, when entering the equation environments:

```
313 \newdimen\eql@paddingleft@
314 \newdimen\eql@paddingright@
315 \let\eql@paddingleft@val\@undefined
316 \let\eql@paddingright@val\@undefined
```

`\eql@display@linewidth` **TODO:** describe

```
\eql@display@marginleft
\eql@display@marginright
317 \let\eql@display@linewidth\@undefined
318 \let\eql@display@marginleft\@undefined
319 \let\eql@display@marginright\@undefined
```

`\eql@box@colsep` The macro `\eql@box@colsep` specifies the intercolumn separation for equation boxes:

```
320 \def\eql@box@colsep{2em}
```

`\eql@break@line@sep` **TODO:** describe

```
\eql@break@line@shortsep
\eql@break@col@sep
\eql@break@col@shortsep
321 \def\eql@break@line@sep{2em minus 1em}
322 \def\eql@break@line@shortsep{1em}
323 \def\eql@break@col@sep{2em minus 1em}
324 \def\eql@break@col@shortsep{1em}
```

`\eql@spread@val` The extra spread of equation lines is specified by `\eql@spread@val`:

```
325 \def\eql@spread@val{\jot}
326 \newdimen\eql@spread@
```

`\eql@tagfuzz@` (*dimen*) The value `\eql@tagfuzz@` specifies the margin of error for comparing whether a tag fits a given equation line. We should not expect rounding errors in the fixed point arithmetic of $\TeX$, nevertheless: **TODO:** probably do not need this due to fixed point arithmetic.

```
327 \newdimen\eql@tagfuzz@
328 \eql@tagfuzz@16sp\relax
```

`\eqldisplay@height` An equation will appear to the surrounding text with a fixed apparent height and depth
`\eqldisplay@depth` specified by `\eqldisplay@height` and `\eqldisplay@depth`, respectively:

```
329 \def\eqldisplay@height{\@undefined
330 \def\eqldisplay@depth{\@undefined
```

`\eqlskip@mode@short` The setting `\eqlskip@mode@short` specifies when a reduced amount of glue should be used around equations in case the text line above the equation fits in the space that is left available in the first equation line. Value 0 turns this feature off, value 1 reduces the glue above the equation, value 2 furthermore reduces the glue below a single equation line and value 3 also reduces the glue below multi-line equations:

```
331 \def\eqlskip@mode@short{2}

332 \def\eqlskip@mode@cont@above{2}
333 \def\eqlskip@mode@cont@below{0}

334 \def\eqlskip@mode@par@above{3}
335 \def\eqlskip@mode@par@below{0}

336 \def\eqlskip@mode@top@above{4}
337 \def\eqlskip@mode@top@below{0}

338 \newcount\eqlskip@mode@leave@
339 \let\eqlskip@force@leave@\@undefined
```

`\eqlskip@force@above` 0: short, 1: long, 2: cont, 3: par, 4: top, 5: no, 6: med, 7: custom

`\eqlskip@force@below`
`\mode@above@ (counter)`
`\mode@below@ (counter)`

```
340 \newcount\eqlskip@mode@above@
341 \newcount\eqlskip@mode@below@
342 \let\eqlskip@force@above@\@undefined
343 \let\eqlskip@force@below@\@undefined
344 \let\eqlskip@custom@above@\@undefined
345 \let\eqlskip@custom@below@\@undefined
```

`\eqlskip@cont@above` The glue when an equation is at the top of a horizontal list is specified by `\eqlskip@cont@above`:

`\eqlskip@top@above` The glue when an equation is at the top of a vertical list is specified by

`\eqlskip@top@below` `\eqlskip@top@above` and `\eqlskip@top@below`:

`\eqlskip@par@above` The glue when an equation starts a paragraph is specified by `\eqlskip@par@above`:

`\eqlskip@med@above` The surrounding glue for an equation with reduced spacing is given by

`\eqlskip@med@below` `\eqlskip@med@above` and `\eqlskip@med@below`:

```
346 \def\eqlskip@long@above{\abovedisplayskip}
347 \def\eqlskip@long@below{\belowdisplayskip}
348 \def\eqlskip@short@above{\abovedisplayshortskip}
349 \def\eqlskip@short@below{\belowdisplayshortskip}
350 \def\eqlskip@cont@above{\eqlskip@short@above}
351 \def\eqlskip@cont@below{\eqlskip@short@below}
352 \def\eqlskip@par@above{\eqlskip@long@above}
353 \def\eqlskip@par@below{\eqlskip@long@below}
354 \def\eqlskip@top@above{\eqlskip@long@above}
355 \def\eqlskip@top@below{\eqlskip@long@below}
356 \def\eqlskip@med@above{\abovedisplayskip/2}
357 \def\eqlskip@med@below{\belowdisplayskip/2}
358 \def\eqlskip@tag@above{\z@skip}
359 \def\eqlskip@tag@below{\z@skip}
```

`\eq@colsepmin@` (*dimen*) The minimum intercolumn separation is specified by `\eq@colsepmin@`. This dimension register is set to `\eq@colsepmin@val` when entering the equation environments to allow font-dependent values. Furthermore, `\eq@colsepmax@val` specifies the maximum intercolumn separation:

```
360 \newdimen\eq@colsepmin@
361 \def\eq@colsepmin@val{1em}
362 \def\eq@colsepmax@val{.5\maxdimen}
```

`\eq@tagwidthmin@` (*dimen*) The minimum tag width is specified by `\eq@tagwidthmin@`:

```
363 \newdimen\eq@tagwidthmin@
364 \eq@tagwidthmin@{\z@}
```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. $\TeX$'s built-in value is half a quad³ in font number 2. As the tag is processed in text mode, we use 0.5em instead.

```
365 \newdimen\eq@tagsepmin@
366 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}
```

`\eq@equations@sqr@opt` Store the default arguments for `\[...]` and `\<...>`, respectively:

```
\eq@equations@ang@opt
\eq@box@ang@opt
367 \def\eq@equations@sqr@opt{equation,nonumber}
368 \def\eq@equations@ang@opt{align,nonumber}
369 \def\eq@box@ang@opt{align}
```

Multi-Line Align Mode.

```
370 \let\eq@columns@fulllength\eq@false
```

3.2 Registers

TODO: describe

General. **TODO:** describe

```
371 \newcount\eq@count@
372 \newdimen\eq@dimen@
373 \newskip\eq@skip@
```

TODO: describe

```
374 \let\eq@display@container\@empty
```

`\eq@cellbox@` (*box*) The box `\eq@cellbox@` holds the present alignment component and `\eq@tagbox@` the tag for the present line. The corresponding dimensions `\eq@cellwidth@` and

`\eq@tagwidth@` (*dimen*) `\eq@tagwidth@` hold their widths. `\eq@prevwidth@` holds the width of the previous

`\eq@prevwidth@` (*dimen*) alignment component: **TODO:** adjust

`\eq@tagwidth@` (*dimen*)

`\eq@prevdepth@` (*dimen*)

`\eq@prevgraf@` (*counter*)

```
375 \newbox\eq@cellbox@
376 \newbox\eq@tagbox@
377 \newdimen\eq@cellwidth@
378 \newdimen\eq@prevwidth@
379 \newdimen\eq@tagwidth@
```

³another half of a quad is left empty at the other end of the line.

```

380 \newdimen\eql@prevdepth@
381 \newcount\eql@prevgraf@

\eql@totalwidth@ (dimen)
\eql@tagwidth@max@ (dimen)
\eql@totalheight@ (dimen)
382 \newdimen\eql@totalwidth@
383 \newdimen\eql@tagwidth@max@
384 \newdimen\eql@totalheight@
385 \newdimen\eql@topheight@
386 \newdimen\eql@bottomdepth@

```

`\eql@line@height@` (dimen) The dimension registers `\eql@line@height@` and `\eql@line@depth@` keep track of the height and depth of the present line in an alignment:

```

387 \newdimen\eql@line@height@
388 \newdimen\eql@line@depth@

\eql@line@width@ (dimen)
\eql@line@avail@ (dimen)
\eql@line@pos@ (dimen)
\eql@line@widthsep@ (counter)
\eql@line@availsep@ (counter)
\eql@line@possep@ (counter)
\eql@line@offset@ (dimen)
\eql@line@prevdepth@ (dimen)
\eql@line@interline@ (dimen)
389 \newdimen\eql@line@width@
390 \newdimen\eql@line@avail@
391 \newdimen\eql@line@pos@
392 \newcount\eql@line@availsep@
393 \newcount\eql@line@widthsep@
394 \newcount\eql@line@possep@
395 \newdimen\eql@line@offset@
396 \newdimen\eql@line@prevdepth@
397 \newdimen\eql@line@interline@

```

Rows and Columns.

`\eql@row@` (counter) **TODO:** `tagrows \eql@row@` counts the present row (1-based) and `\eql@totalrows@` holds the total number of rows:

```

\eql@tagrows@ (counter)
398 \newcount\eql@row@
399 \newcount\eql@totalrows@
400 \newcount\eql@tagrows@

```

```

\eql@column@
\eql@totalcolumns@
401 \newcount\eql@column@
402 \newcount\eql@totalcolumns@

```

`\eql@colsep@` (dimen) The dimension of the intercolumn separation for align environments is stored in `\eql@colsep@`:

```

403 \newdimen\eql@colsep@

\eql@intercolumns@ (counter)
404 \newcount\eql@intercolumns@

```

Vertical Spacing Adjustments.

`\firstavail@ (dimen)` The unused space on the first line of an alignment is stored in `\eql@display@firstavail@` for comparison against `\predisplaysize` and determining short skip mode of display equations. It is convenient to set it via `\eql@display@firstavail@set` provided that we are on the first line:

```
405 \newdimen\eql@display@firstavail@
406 \def\eql@display@firstavail@set#1{%
407   \ifnum\eql@row@=\@ne
408     \global\eql@appendexpand\eql@display@container{%
409       \eql@display@firstavail@the#1\relax}%
410   \fi
411 }
```

The counter stores whether the tag one first/last line is raised/lowered as 1/2 (or 3 for both). This implies a different `vskip` corresponding to the mostly empty line: **TODO:** adjust

```
412 \newdimen\eql@display@aboveextend@
413 \newdimen\eql@display@belowextend@
```

Shared Registers.

`\ifmeasuring@ (bool)` All display environments get typeset twice – once during a “measuring” phase and then again during a “production” phase. We reuse the original `amsmath` definition `\ifmeasuring@` to determine which case we’re in, so we and other packages may take appropriate action. It does not hurt to define this conditional in any case. We should tell `hyperref` about measuring processes as we’re not `amsmath` and not being catered for:

```
414 \ifdefined\measuring@true\else
415   \expandafter\newif\csname ifmeasuring@\endcsname
416 \fi
417 \AddToHook{package/hyperref/after}{
418   \ifdefined\Hy@ifnotmeasuring
419     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
420   \fi
421 }
```

`\if@display (bool)` `amsmath` defines the conditional `\if@display` to test whether we’re in a display equation including the inner math parts of equation blocks. We provide it in case `amsmath` is absent, and initialise it:

```
422 \ifdefined\@displaytrue\else
423   \expandafter\newif\csname if@display\endcsname
424   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
425 \fi
```

3.3 Hooks

`\eql@hook@...` For what it’s worth, we define a couple of entry points where one might hook into the equations typesetting framework. The L^AT_EX hook framework would be more versatile, but as the purpose of these hooks is rather unclear at the moment, we make this as efficient as it could get: **TODO:** may add a few more hooks

```
426 \let\eql@hook@blockbefore\@empty
427 \let\eql@hook@blockafter\@empty
428 \let\eql@hook@blockin\@empty
429 \let\eql@hook@blockout\@empty
```

```

430 \let\eql@hook@linein\@empty
431 \let\eql@hook@lineout\@empty
432 \let\eql@hook@colin\@empty
433 \let\eql@hook@colout\@empty
434 \let\eql@hook@eqin\@empty
435 \let\eql@hook@eqout\@empty
436 \let\eql@hook@innerleft\@empty
437 \let\eql@hook@innerright\@empty
438 \let\eql@hook@innerlead\@empty

```

4 Features

4.1 Punctuation

The equations environments supply an automatic punctuation scheme which allows to define a default punctuation at the end of each column, line and equation block.

`\eql@punct@col` These macros store the punctuation character for columns, lines and blocks. A value
`\eql@punct@line` `\relax` indicates that the punctuation should be handed down to the next lower level:
`\eql@punct@block` **TODO:** update

```

439 \let\eql@punct@col\@empty
440 \let\eql@punct@line\relax
441 \let\eql@punct@block\relax
442 \let\eql@punct@main\relax

```

`\eql@punct@sep` This macro stores the separation to be applied before the punctuation (unless it is empty):

```

443 \let\eql@punct@sep\relax

```

`\eql@punct@set` **TODO:** describe

```

444 \def\eql@punct@relax{\relax}
445 \def\eql@punct@tilde{~}
446 \def\eql@punct@set#1#2{%
447   \def#1{#2}%
448   \ifx#1\eql@punct@relax
449     \let#1\relax
450   \fi
451   \ifx#1\eql@punct@tilde
452     \let#1\@empty
453   \fi
454 }

```

`\eqnpunct` Set the punctuon for columns, lines and blocks. Note that the macro `\eqnpunct` sets the punctuation for the following equation block or for the current cell. Starred versions clear the punctuation for the respectively levels:

```

455 \def\eqnpunct{%
456   \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
457 \def\eql@punct@next@set#1{%
458   \ifmmode
459     \eql@punct@set\eql@punct@col{#1}%
460     \eql@punct@set\eql@punct@line{#1}%
461     \eql@punct@set\eql@punct@block{#1}%
462   \else

```

```

463 \eqnaddopt{punct={#1}}%
464 \fi
465 \ignorespaces}
466 \def\eql@punct@next@setrelax{%
467 \ifmmode
468 \let\eql@punct@block\relax
469 \else
470 \eqnaddopt{punct*}%
471 \fi
472 \ignorespaces}

```

`\eql@punct@print@col` Output the punctuation for the present column. If non-empty, prepend some separation:

```

473 \def\eql@punct@print@col{%
474 \ifx\eql@punct@col\@empty\else
475 \eql@punct@sep
476 \eql@punct@col
477 \fi
478 }

```

`\eql@punct@apply@col` Output the punctuation for the present column. Clear the punctuation so that no further column punctuation is output within the current group:

```

479 \def\eql@punct@apply@col{%
480 \eql@punct@print@col
481 \let\eql@punct@col\@empty
482 }

```

Output the punctuation currently set for lines unless disabled:

`\eql@punct@print@line`

```

483 \def\eql@punct@print@line{%
484 \ifx\eql@punct@line\relax
485 \eql@punct@print@col
486 \else
487 \ifx\eql@punct@line\@empty\else
488 \eql@punct@sep
489 \eql@punct@line
490 \fi
491 \fi
492 }

```

Output the punctuation currently set for lines unless disabled. Alike `\eql@punct@apply@col` prevent further output of punctuations for lines and columns within the current group:

`\eql@punct@apply@line`

```

493 \def\eql@punct@apply@line{%
494 \ifx\eql@punct@line\relax
495 % \TODO hand down immediately?
496 \else
497 \ifx\eql@punct@line\@empty\else
498 \eql@punct@sep
499 \eql@punct@line
500 \fi
501 \let\eql@punct@line\relax
502 \let\eql@punct@col\@empty
503 \fi

```

504 }

`\eql@punct@apply@block` Outputs the punctuation for the current equation block unless disabled in analogy to
`\eqnpunctapply \eql@punct@apply@line:`

```

505 \def\eql@punct@apply@block{%
506   \ifx\eql@punct@block\relax
507   % \TODO hand down immediately?
508   \else
509     \ifx\eql@punct@block\@empty\else
510       \eql@punct@sep
511       \eql@punct@block
512     \fi
513     \let\eql@punct@block\relax
514     \let\eql@punct@line\relax
515     \let\eql@punct@col\@empty
516   \fi
517 }

518 \let\eqnpunctapply\eql@punct@apply@block

```

4.2 Math Classes at Alignment

The following describes the adjustment of math classes surrounding the alignment marker.

`\class@innerright@sel@` Select between `\eql@class@innerlead` and `\eql@class@innerright` depending on whether the left part of the aligned column is empty:

```

519 \def\eql@class@innerright@sel{%
520   \ifdim\eql@prevwidth@=\z@
521     \eql@class@innerlead
522   \else
523     \eql@class@innerright
524   \fi
525 }

```

`\class@innerleft@set` Set the left, right and leading math classes. Setting the right math class disables the
`\class@innerright@set` leading math class, so the leading math class must be specified after the right one:
`\class@innerlead@set`

```

526 \def\eql@class@innerleft@set#1{%
527   \def\eql@class@innerleft{#1}%
528 }
529 \def\eql@class@innerright@set#1{%
530   \def\eql@class@innerright{#1}%
531   \let\eql@class@innerright@sel\eql@class@innerright
532 }
533 \def\eql@class@innerlead@set#1{%
534   \def\eql@class@innerlead{#1}%
535   \let\eql@class@innerright@sel\eql@class@innerright@sel@
536 }

```

`\eql@class@ampeq` We define two standard combinations of math classes intended to be used with ‘&=’
`\eql@class@eqamp` (ampeq) or ‘&’ (eqamp). The default setting is ‘&=’ (ampeq):

```

537 \def\eql@class@ampeq{%
538   \eql@class@innerleft@set{}%
539   \eql@class@innerright@set{}}%
540 }

```

```

541 \def\eql@class@eqamp{%
542   \eql@class@innerleft@set{\mathrel{}}}%
543   \eql@class@innerright@set{\mathrel{}}}%
544   \eql@class@innerlead@set{}}}%
545 }
546 \eql@class@ampeq

```

4.3 Framed Cells

TODO: describe **TODO:** warn if issued in even cells

```

547 \let\eql@frame@cmd@\undefined
548 \newdimen\eql@frame@margin@
549 \def\eql@frame@set[#1]{%
550   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
551   \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbbox}
552   \def\eql@frame@measure{%
553     \setbox\z@\hbox{\eql@frame@cmd{}}%
554     \eql@frame@margin@.5\wd\z@
555   }
556   \def\eql@frame@print{%
557     \setbox\eql@cellbox@\hbox{%
558       \eql@frame@cmd{\unhbox\eql@cellbox@}%
559     }%
560   }
561   \def\eql@frame@adjust{%
562     \setbox\eql@cellbox@\hbox{%
563       \eql@frame@cmd{%
564         \unhbox\eql@cellbox@
565         \unkern
566         \unskip
567       }%
568       \hfil
569       \kern\z@
570     }%
571   }

```

4.4 Single-Line Composition

TODO: describe

`\eql@break@line`

```

572 \def\eql@break@line{\eql@srbgroup
573   \eql@ifnextgobble@tight~\eql@break@line@star
574   {\eql@punct@print@line\eql@break@line@star}}
575 \def\eql@break@line@star{%
576   \eql@ifstar@tight
577   {\eql@break@opt[\eql@break@line@shortsep]}%
578   {\eql@testopt@tight\eql@break@opt\eql@break@line@sep}}

```

`\eql@break@col`

```

579 \def\eql@break@col{\eql@srbgroup
580   \eql@ifnextgobble@tight~\eql@break@col@star
581   {\eql@punct@print@col\eql@break@col@star}}
582 \def\eql@break@col@star{%

```

```

583 \eq@ifstar@tight
584   {\eq@break@opt[\eq@break@col@shortsep]}%
585   {\eq@testopt@tight\eq@break@opt\eq@break@col@sep}}

\eq@break@opt

586 \def\eq@break@opt[#1]{\eq@sregroup\hspace{\glueexpr#1\relax}}

\eq@break@join

587 \def\eq@break@join{\eq@srbgroup
588   \eq@ifstar@tight
589   {\eq@break@join@opt[\eq@break@col@shortsep]}%
590   {\eq@testopt@tight\eq@break@join@opt\eq@break@col@sep}}
591 \def\eq@break@join@opt[#1]#2{\eq@sregroup%
592   \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

\eqnsep TODO: expand to lines and columns mode
\eqnbreak
\eqnjoin 593 \def\eqnsep{\eq@break@col}
594 \def\eqnbreak{\eq@break@line}
595 \def\eqnjoin{\eq@break@join}

```

4.5 Alternative Content Description

TODO: describe **TODO:** would be nice to provide as environments as well **TODO:** implement for PDF tagging

```
596 \DeclareRobustCommand{\eqnalt}[2][{}]{}
```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```

597 \let\eq@tags@autolabel\eq@false
598 \let\eq@tags@autotag\eq@true
599 \let\eq@tags@warn\eq@true

600 \def\eq@tags@name@generic{[equation]}

601 \let\eq@tagpos@doconvert\eq@false

602 \def\eq@tagpos@snap{4pt}

```

Registers.

```

603 \let\eq@numbering@mode\@undefined

604 \let\eq@numbering@active\eq@true
605 \let\eq@numbering@multi\eq@true

```

```

606 \let\eql@tags@container\@undefined
607 \def\eql@tags@container@clear{%
608   \let\eql@tags@label\@undefined
609   \let\eql@tags@name\@undefined
610   \let\eql@tags@tag\@undefined
611   \let\eql@tags@ref\@undefined
612   \let\eql@tags@anchor\@empty
613   \eql@tagpos@shift@z@
614   \eql@tagpos@smashup@z@
615   \eql@tagpos@smashdown@z@
616   \let\eql@tagpos@reserve\eql@true
617 }

618 \let\eql@tags@label\@undefined
619 \let\eql@tags@name\@undefined
620 \let\eql@tags@tag\@undefined
621 \let\eql@tags@ref\@undefined
622 \let\eql@tags@frame@cmd\@firstofone

```

tags@glabel@ (*counter*)

```

623 \newcount\eql@tags@glabel@
624 \eql@tags@glabel@z@
625 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
626 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\@ne}

627 \let\eql@tagpos@continuous\eql@false

628 \newcount\eql@tagpos@row@
629 \newcount\eql@tagpos@prevrow@
630 \newdimen\eql@tagpos@shift@
631 \newdimen\eql@tagpos@smashup@
632 \newdimen\eql@tagpos@smashdown@
633 \newdimen\eql@tagpos@current@
634 \newdimen\eql@tagpos@plain@
635 \newdimen\eql@tagpos@raised@
636 \newdimen\eql@tagpos@target@
637 \newdimen\eql@tagpos@headroom@
638 \newdimen\eql@tagpos@footroom@

```

5.2 Schemes

TODO: describe

Table.

```

639 \def\eql@numbering@tab@sub{sub}
640 \def\eql@numbering@tab@all{all}
641 \def\eql@numbering@tab@first{first}
642 \def\eql@numbering@tab@last{last}
643 \def\eql@numbering@tab@in{in}
644 \def\eql@numbering@tab@out{out}
645 \def\eql@numbering@tab@middle{middle}
646 \def\eql@numbering@tab@best{best}
647 \def\eql@numbering@tab@here{here}
648 \def\eql@numbering@tab@top{top}
649 \def\eql@numbering@tab@bottom{bottom}

```

```

650 \def\eql@numbering@tab@center{center}
651 \def\eql@numbering@tab@centerone{centerone}
652 \def\eql@numbering@tab@median{median}
653 \def\eql@numbering@tab@baseline{baseline}

654 \let\eql@numbering@mode\eql@numbering@tab@all
655 \let\eql@numbering@mode@multi\eql@numbering@tab@all
656 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

657 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
658 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
659 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
660 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
661 \let\eql@numbering@tab@outside\eql@numbering@tab@out
662 \let\eql@numbering@tab@inside\eql@numbering@tab@in
663 \let\eql@numbering@tab@within\eql@numbering@tab@in
664 \let\eql@numbering@tab@opt\eql@numbering@tab@best
665 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
666 \let\eql@numbering@tab@pick\eql@numbering@tab@here
667 \let\eql@numbering@tab@med\eql@numbering@tab@median
668 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
669 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone

```

TODO: describe

```

670 \let\eql@numbering@tab@a\eql@numbering@tab@all
671 \let\eql@numbering@tab@s\eql@numbering@tab@sub
672 \let\eql@numbering@tab@f\eql@numbering@tab@first
673 \let\eql@numbering@tab@l\eql@numbering@tab@last
674 \let\eql@numbering@tab@o\eql@numbering@tab@out
675 \let\eql@numbering@tab@i\eql@numbering@tab@in
676 \let\eql@numbering@tab@h\eql@numbering@tab@here
677 \let\eql@numbering@tab@t\eql@numbering@tab@top
678 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
679 \let\eql@numbering@tab@c\eql@numbering@tab@center
680 \let\eql@numbering@tab@m\eql@numbering@tab@median
681 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
682 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
683 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
684 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

685 \def\eql@numbering@init@all{\let\eql@numbering@mode\eql@numbering@tab@all}

```

TODO: describe

```

686 \def\eql@numbering@init@sub{%
687   \let\eql@numbering@mode\eql@numbering@tab@all
688   \ifdefined\eql@subequations@active
689     \let\eql@numbering@mode\eql@numbering@tab@all
690   \else
691     \let\eql@numbering@mode\eql@numbering@tab@sub
692   \fi
693 }

694 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}}
695 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}}
696 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}}

```

TODO: describe

```
697 \def\eq@numbering@init@in{%
698   \ifdefined\eq@tagsleft
699     \eq@numbering@init@last
700   \else
701     \eq@numbering@init@first
702   \fi
703 }
```

TODO: describe

```
704 \def\eq@numbering@init@out{%
705   \ifdefined\eq@tagsleft
706     \eq@numbering@init@first
707   \else
708     \eq@numbering@init@last
709   \fi
710 }
```

TODO: describe

```
711 \def\eq@tagpos@eval@middle{%
712   \ifnum\eq@tagpos@row@=\z@
713     \eq@tagpos@row@=\numexpr(\eq@totalrows@
714       +\ifdefined\eq@tagsleft\z@\else\@ne\fi)/\tw@\relax
715   \fi
716 }
```

TODO: describe

```
717 \def\eq@tagpos@eval@best{%
718   \ifnum\eq@tagpos@row@=\z@
719     \let\eq@numbering@best@use\eq@true
720     \eq@numbering@init@out
721   \fi
722 }
```

TODO: describe

```
723 \def\eq@numbering@init@continuous{\let\eq@tagpos@continuous\eq@true}
```

TODO: describe

```
724 \let\eq@numbering@init@top\eq@numbering@init@continuous
725 \def\eq@tagpos@eval@top{%
726   \eq@tagpos@current@\z@
727 }
```

TODO: describe

```
728 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
729 \def\eq@tagpos@eval@bottom{%
730   \eq@tagpos@current@\dimexpr\eq@totalheight@
731     -\eq@tagheight@block@-\eq@tagdepth@block@\relax
732 }
```

TODO: describe

```
733 \let\eq@numbering@init@center\eq@numbering@init@continuous
734 \def\eq@tagpos@eval@center{%
735   \ifnum\eq@totalrows@=\@ne
736     \eq@tagpos@row@\@ne
```

```

737 \fi
738 \eq\tagpos@current@\dimexpr(\eq\totalheight@
739   -\eq\tagheight@block@-\eq\tagdepth@block@)/\tw@\relax
740 }

```

TODO: describe

```

741 \let\eq\numbering@init@centerone\eq\numbering@init@continuous
742 \def\eq\tagpos@eval@centerone{%
743   \eq\tagpos@current@\dimexpr(\eq\totalheight@
744     -\eq\tagheight@block@-\eq\tagdepth@block@)/\tw@\relax
745 }

```

TODO: describe

```

746 \let\eq\numbering@init@baseline\eq\numbering@init@continuous
747 \def\eq\tagpos@eval@baseline{%
748   \eq\tagpos@current@\dimexpr(\eq\totalheight@
749     +\eq\topheight@-\eq\bottomdepth@)/\tw@-\eq\tagheight@block@\relax
750 }

```

TODO: describe

```

751 \let\eq\numbering@init@median\eq\numbering@init@continuous
752 \def\eq\tagpos@eval@median{%
753   \ifnum\eq\tagpos@row@=\z@
754     \ifodd\eq\totalrows@
755       \eq\tagpos@row@\numexpr(\eq\totalrows@+\@ne)/\tw@\relax
756     \else
757       \eq\tagpos@row@\numexpr(\eq\totalrows@+\tw@)/\tw@\relax
758       \eq\dimensions@get\eq\tagpos@row@
759       \advance\eq\tagpos@shift@\dimexpr\eq\line@height@
760         +(\eq\line@interline@-\eq\tagheight@block@
761           +\eq\tagdepth@block@)/\tw@\relax
762     \fi
763   \ifnum\eq\totalrows@=\@ne
764     \eq\tagpos@row@\@ne
765   \else
766     \eq\tagpos@adjust@eval@convert
767     \eq\tagpos@row@\z@
768   \fi
769 \fi
770 }

```

Selection.

```

771 \def\eq\numbering@set#1{%
772   \ifcsname eq\numbering@tab@#1\endcsname
773     \expandafter\let\expandafter\eq\numbering@mode
774       \csname eq\numbering@tab@#1\endcsname
775     \ifx\eq\numbering@mode\eq\numbering@tab@all
776       \let\eq\numbering@mode@multi\eq\numbering@mode
777     \else\ifx\eq\numbering@mode\eq\numbering@tab@sub
778       \let\eq\numbering@mode@multi\eq\numbering@mode
779     \else
780       \let\eq\numbering@mode@single\eq\numbering@mode
781     \fi\fi
782   \else
783     \eq\error{numbering mode '#1' unknown: setting mode to 'all'}%
784     \let\eq\numbering@mode\eq\numbering@tab@all

```

```

785 \fi
786 }

```

TODO: describe

```

787 \def\eql@numbering@init{%
788   \let\eql@numbering@multi\eql@false
789   \let\eql@tagpos@continuous\eql@false
790   \let\eql@numbering@subeq@use\eql@false
791   \let\eql@numbering@best@use\eql@false
792   \eql@tagpos@row@\z@
793   \csname eql@numbering@init@\eql@numbering@mode\endcsname
794   \ifdefined\eql@numbering@active
795     \let\eql@numbering@eqnswinit\@eqnswtrue
796   \else
797     \let\eql@numbering@eqnswinit\@eqnswfalse
798   \fi
799   \let\eql@numbering@active\eql@false
800 }

```

5.3 Interface

Activation. **TODO:** note `\nonumber` already defined, modifications by `amsmath`

```

801 \eql@amsmath@after{
802   \let\eql@print@eqnum@default\print@eqnum
803   \let\eql@incr@eqnum@default\incr@eqnum
804 }

```

TODO: describe

```

805 \protected\def\donumber{%
806   \if@eqnsw\else
807     \global\@eqnswtrue
808     \ifx\print@eqn\@empty
809       \global\let\print@eqn\eql@print@eqnum@default
810     \fi
811     \ifx\incr@eqn\@empty
812       \global\let\incr@eqn\eql@incr@eqnum@default
813     \fi
814   \fi
815 }

```

TODO: reconsider operation

`\numberhere`

```

816 \protected\def\eql@numberhere{%
817   \ifdefined\eql@numbering@multi
818     \global\@eqnswtrue
819   \else
820     \global\eql@tagpos@row@\eql@row@
821   \fi
822 }

```

TODO: describe

`\numbernext`

```

823 \protected\def\eql@numbernext{%

```

```

824 \ifdefined\eql@numbering@multi
825   \global\@eqnswfalse
826 \else
827   \ifdefined\eql@tagpos@continuous\else
828     \ifnum\eql@tagpos@row@=\eql@row@
829       \global\advance\eql@tagpos@row@\@ne
830     \fi
831   \fi
832 \fi
833 }

```

Activation Trigger.

```

834 \def\eql@tags@autoenable{%
835   \global\@eqnswtrue
836   \ifnum\eql@tagpos@row@=\m@ne
837     \numberhere
838   \fi
839 }

```

Labels. **TODO:** describe

\eql@label@org

```

840 \let\eql@label@org\label

```

TODO: describe

```

841 \def\eql@label@gobble{\eql@ampprotect\eql@testopt@tight\eql@gobbleoptone{}}

```

TODO: describe

```

842 \protected\def\eql@label{%
843   \eql@ampprotect\eql@testopt@tight\eql@tags@add@labelname\eql@testopt@default
844 }

```

TODO: describe

```

845 \def\eql@tags@add@labelname[#1]#2{%
846   \def\eql@tmp{#1}%
847   \ifx\eql@tmp\eql@testopt@default\else
848     \eql@tags@add@name{#1}%
849   \fi
850   \eql@tags@add@label{#2}%
851 }

```

TODO: describe

```

852 \def\eql@tags@set@label#1{%
853   \ifdefined\eql@tags@warn
854     \ifdefined\eql@tags@label
855       \eql@warn@label@multiple{#1}%
856     \fi
857   \fi
858   \def\eql@tags@label{#1}%
859 }

```

TODO: describe

```

860 \def\eql@tags@set@name#1{%

```

```

861 \ifdefined\eql@tags@warn
862   \ifdefined\eql@tags@name
863     \eql@warn@name@multiple
864   \fi
865 \fi
866 \def\eql@tags@name{#1}%
867 }

```

TODO: describe

```

868 \def\eql@tags@add@label#1{%
869   \ifdefined\eql@tags@autolabel
870     \eql@tags@autoenable
871   \fi
872   \global\eql@appendexpand\eql@tags@container{%
873     \noexpand\eql@tags@set@label{#1}}%
874 }

```

TODO: describe

```

875 \def\eql@tags@add@name#1{%
876   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
877   \global\eql@appendmacro\eql@tags@container\eql@tmp
878 }

```

TODO: describe

```

879 \def\eql@tags@addblock@label#1{%
880   \eql@appendexpand\eql@tags@container@block{%
881     \noexpand\eql@tags@set@label{#1}}%
882 }

```

TODO: describe

```

883 \def\eql@tags@addblock@name#1{%
884   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
885   \eql@appendmacro\eql@tags@container@block\eql@tmp
886 }

```

Tags. **TODO:** describe

`\eql@tag@default`

```

887 \protected\def\eql@tag@default{%
888   \eql@warn@here\tag
889   \eql@tag@gobble
890 }
891 \let\tag\eql@tag@default

```

`\eql@tag@gobble`

```

892 \def\eql@tag@gobble{%
893   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}

```

TODO: describe

```

894 \protected\def\eql@tag{%
895   \eql@ampprotecttwo\eql@teststaropt@tight
896   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
897   \eql@testopt@default
898 }

```

\eql@tags@add@tagref

```
899 \def\eql@tags@add@tagref[#1]#2{%
900   \def\eql@tmp{#1}%
901   \ifx\eql@tmp\eql@testopt@default\else
902     \eql@tags@add@ref{#1}%
903   \fi
904   \eql@tags@add@tag{#2}%
905 }
```

TODO: describe

```
906 \def\eql@tags@set@tag#1{%
907   \ifdefined\eql@tags@warn
908     \ifdefined\eql@tags@tag
909       \eql@warn@tag@multiple
910     \fi
911   \fi
912   \def\eql@tags@tag{#1}%
913 }
```

TODO: describe

```
914 \def\eql@tags@set@ref#1{%
915   \ifdefined\eql@tags@warn
916     \ifdefined\eql@tags@ref
917       \eql@warn@ref@multiple
918     \fi
919   \fi
920   \def\eql@tags@ref{#1}%
921 }
```

TODO: describe

```
922 \def\eql@tags@add@tag#1{%
923   \ifdefined\eql@tags@autotag
924     \eql@tags@autoenable
925   \fi
926   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
927   \global\eql@appendmacro\eql@tags@container\eql@tmp
928 }
```

TODO: describe

```
929 \def\eql@tags@add@ref#1{%
930   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
931   \global\eql@appendmacro\eql@tags@container\eql@tmp
932 }
```

tags@add@tagform@off

```
933 \def\eql@tags@add@tagform@off{%
934   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
935 }
```

TODO: describe

```
936 \def\eql@tags@addblock@tag#1{%
937   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
938   \eql@appendmacro\eql@tags@container@block\eql@tmp
939 }
```

TODO: describe

```
940 \def\eql@tags@addblock@ref#1{%
941   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
942   \eql@appendmacro\eql@tags@container@block\eql@tmp
943 }
```

TODO: describe

```
944 \def\eql@tags@addblock@tagform@off{%
945   \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
946 }
```

Raise Tags.

`\raisetag`

```
947 \def\eql@raisetag@default{%
948   \eql@warn@here\raisetag
949   \eql@raisetag@gobble
950 }

951 \def\eql@raisetag@gobble{%
952   \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
953 }
```

TODO: describe

```
954 \eql@amsmath@let\raisetag\eql@raisetag@default

955 \def\eql@raisetag{%
956   \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
957 }

958 \def\eql@raisetag@test#1{%
959   \def\eql@tmpa{#1}\def\eql@tmpb{!}%
960   \ifx\eql@tmpa\eql@tmpb
961     \eql@tags@add@forceraise
962   \else
963     \eql@tags@add@raisesmash{#1}%
964   \fi
965 }

966 \def\eql@tags@add@raiseshift#1{%
967   \global\eql@appendexpand\eql@tags@container{%
968     \advance\eql@tagpos@shift@the\glueexpr#1\relax\relax}%
969 }

970 \def\eql@tags@add@raisesmash#1{%
971   \dimen@0\glueexpr#1\relax
972   \ifdim\dimen@<\z@
973     \global\eql@appendexpand\eql@tags@container{%
974       \advance\eql@tagpos@smashdown@the\dimen@\relax}%
975   \else
976     \global\eql@appendexpand\eql@tags@container{%
977       \advance\eql@tagpos@smashup@the\dimen@\relax}%
978   \fi
979 }

980 \def\eql@tags@add@forceraise{%
981   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
982 }
```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```
983 \def\eql@numbering@settools{%
984   \let\label\eql@label
985   \let\tag\eql@tag
986   \let\raisetag\eql@raisetag
987   \let\numberhere\eql@numberhere
988   \let\numbernext\eql@numbernext
989 }
```

TODO: not necessary anymore

```
990 \def\eql@numbering@settools@gobble{%
991   \let\label\eql@label@gobble
992   \let\tag\eql@tag@gobble
993   \let\raisetag\eql@raisetag@gobble
994   \let\numberhere\relax
995   \let\numbernext\relax
996 }

997 \def\eql@numbering@autoblock{%
998   \begingroup
999     \let\eql@tags@warn\eql@false
1000     \eql@tags@container@block
1001     \ifdefined\eql@tags@autolabel
1002       \ifdefined\eql@tags@label
1003         \global\@eqnswtrue
1004       \fi
1005     \fi
1006     \ifdefined\eql@tags@autotag
1007       \ifdefined\eql@tags@tag
1008         \global\@eqnswtrue
1009       \fi
1010     \fi
1011   \endgroup
1012 }

1013 \def\eql@numbering@warnunused{%
1014   \ifdefined\eql@tags@label
1015     \eql@warn@label@unused
1016   \fi
1017   \ifdefined\eql@tags@name
1018     \eql@warn@name@unused
1019   \fi
1020   \ifdefined\eql@tags@tag
1021     \eql@warn@tag@unused
1022   \fi
1023   \ifdefined\eql@tags@erf
1024     \eql@warn@ref@unused
1025   \fi
1026 }
```

Single Line. **TODO:** describe

```

1027 \def\eql@numbering@single@init{%
1028   \let\eql@numbering@multi\eql@false
1029   \eql@numbering@settools
1030   \eql@numbering@eqnswinit
1031   \eql@numbering@autoblock
1032   \global\let\eql@tags@container\eql@tags@container@block
1033   \let\eql@tags@warn\eql@true
1034 }

1035 \def\eql@numbering@single@eval{%
1036   \ifnum\eql@tagpos@row@=\m@ne
1037     \@eqnswfalse
1038   \fi
1039 }

```

Multi-Line Measuring Pass. **TODO:** describe

```

1040 \def\eql@numbering@measure@init{%
1041   \eql@numbering@settools
1042   \ifdefined\eql@numbering@multi\else
1043     \eql@numbering@eqnswinit
1044     \eql@numbering@autoblock
1045   \fi
1046   \global\let\eql@tags@container\eql@tags@container@block
1047   \let\eql@tags@warn\eql@true
1048 }

```

TODO: might select only relevant routines in init **TODO:** describe

```

1049 \def\eql@numbering@measure@line@begin{%
1050   \ifdefined\eql@numbering@multi
1051     \global\eql@numbering@eqnswinit
1052   \fi
1053 }

```

TODO: describe

```

1054 \def\eql@numbering@measure@blocktag{%
1055   \ifdefined\eql@numbering@multi
1056     \@eqnswfalse
1057   \else
1058     \ifnum\eql@tagpos@row@=\m@ne
1059       \@eqnswfalse
1060     \fi
1061     \ifnum\eql@totalrows@=\z@
1062       \@eqnswfalse
1063     \fi
1064   \fi
1065 }

```

Multi-Line Print Pass. **TODO:** describe

TODO: can we be absolutely sure about all values being preserved global might pick up a value from a higher level block and restore it globally!

```

1066 \def\eql@numbering@print@init{%
1067   \let\eql@tags@warn\eql@false
1068   \ifdefined\eql@numbering@multi
1069     \eql@numbering@settools

```

```

1070 \global\let\eq\eq\tags@container\eq\tags@container@block
1071 \else
1072 \let\eq\eq\tags@container@block\eq\tags@container
1073 \eq\@numbering@settools@gobble
1074 \fi
1075 }

```

TODO: might select only relevant routines in init **TODO:** describe

```

1076 \def\eq\@numbering@print@block@begin{%
1077 \ifdefined\eq\@numbering@multi\else
1078 \ifnum\eq\@tagpos@row@>\z@
1079 \eq\@tags@makeblockanchor
1080 \global\eq\@appendexpand\eq\@tags@container@block{%
1081 \def\noexpand\eq\@tags@anchor{%
1082 \unexpanded\expandafter{\eq\@tags@anchor}}}%
1083 \fi
1084 \fi
1085 \ifdefined\eq\@numbering@subeq@use
1086 \eq\@tags@printsubeqlabel
1087 \fi
1088 }

```

TODO: describe

```

1089 \def\eq\@numbering@print@line@begin{%
1090 \ifdefined\eq\@numbering@multi
1091 \global\eq\@numbering@eqnswinit
1092 \fi
1093 }

```

TODO: describe

```

1094 \def\eq\@numbering@print@line@eval{%
1095 \ifdefined\eq\@numbering@multi
1096 \if@eqnsw
1097 \eq\@tags@container
1098 \fi
1099 \else
1100 \ifnum\eq\@tagpos@row@=\eq\@row@
1101 \@eqnswtrue
1102 \eq\@tags@container@block
1103 \else
1104 \@eqnswfalse
1105 \fi
1106 \fi
1107 }

```

5.5 Positioning

TODO: describe

```

1108 \def\eq\@tagpos@single@eval{%
1109 \if@eqnsw
1110 \csname eq\@tagpos@eval@\eq\@numbering@mode\endcsname
1111 \ifnum\eq\@tagpos@row@>\@one
1112 \eq\@tagpos@row@\@one
1113 \fi
1114 \ifdefined\eq\@tagpos@doconvert
1115 \let\eq\@tagpos@continuous\eq\@true

```

```

1116 \fi
1117 \ifdefined\eql@tagpos@continuous
1118 \eql@tagpos@single@eval@continuous
1119 \fi
1120 \else
1121 \eql@tagpos@row@z@
1122 \fi
1123 \eql@tagpos@prevrow@z@
1124 \eql@tagpos@headroom@z@
1125 \eql@tagpos@footroom@z@
1126 }

```

TODO: describe

```

1127 \def\eql@tagpos@single@eval@continuous{%
1128 \ifnum\eql@tagpos@row@>z@
1129 \eql@tagpos@target@eql@tagpos@shift@
1130 \else
1131 \eql@tagpos@target@dimexpr\eql@line@height@
1132 -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@relax
1133 \fi
1134 \eql@tagpos@row@one
1135 \ifdim\ifdim\eql@tagpos@target@<z@-\fi
1136 \eql@tagpos@target@<\glueexpr\eql@tagpos@snaprelax
1137 \eql@tagpos@target@z@
1138 \fi
1139 }

```

TODO: describe

```

1140 \def\eql@tagpos@adjust@eval{%
1141 \if@eqnsw
1142 \csname eql@tagpos@eval@eql@numbering@mode\endcsname
1143 \ifnum\eql@tagpos@row@>\eql@totalrows@
1144 \eql@tagpos@row@\eql@totalrows@
1145 \fi
1146 \ifdefined\eql@tagpos@doconvert
1147 \let\eql@tagpos@continuous\eql@true
1148 \fi
1149 \ifdefined\eql@tagpos@continuous
1150 \ifnum\eql@tagpos@row@>z@
1151 \eql@tagpos@adjust@eval@convert
1152 \fi
1153 \eql@tagpos@adjust@eval@continuous
1154 \fi
1155 \else
1156 \eql@tagpos@row@z@
1157 \eql@tagpos@prevrow@z@
1158 \fi
1159 }

```

TODO: describe

```

1160 \def\eql@tagpos@adjust@eval@convert{%
1161 \eql@tagpos@current@z@
1162 \eql@dimensions@for{%
1163 \ifnum\eql@row@<\eql@tagpos@row@
1164 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1165 +\eql@line@height@+\eql@line@depth@relax
1166 \fi
1167 \ifnum\eql@row@=\eql@tagpos@row@

```

```

1168     \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1169     +\eql@line@height@-\eql@tagheight@block@\relax
1170   \fi
1171 }%
1172 }

```

TODO: describe

```

1173 \def\eql@tagpos@adjust@eval@continuous{%
1174   \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1175   \eql@tagpos@row@\eql@totalrows@
1176   \eql@tagpos@prevrow@\z@
1177   \eql@tagpos@headroom@\z@
1178   \eql@tagpos@footroom@\z@
1179   \eql@dimensions@for{%
1180     \ifnum\eql@tagpos@row@=\eql@totalrows@
1181       \eql@tagpos@headroom@\eql@line@interline@
1182       \eql@tagpos@target@\dimexpr\eql@line@interline@
1183       +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1184       \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1185       \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1186       \advance\dimen@\eql@tagpos@target@
1187       \eql@tagpos@target@\z@
1188     \fi
1189     \ifdim\eql@tagpos@target@>%
1190       \ifdefined\eql@tagleft-1sp\relax\else\z@\fi
1191       \eql@tagpos@row@\eql@row@
1192       \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1193     \fi
1194     \advance\dimen@-\dimexpr\eql@line@interline@
1195     +\eql@line@depth@+\eql@line@height@\relax
1196   \fi
1197   \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1198     \eql@tagpos@footroom@\eql@line@interline@
1199   \fi
1200 }%
1201 }

```

TODO: describe

```

1202 \def\eql@tagpos@print@line@eval{%
1203   \ifdefined\eql@tagpos@continuous
1204     \eql@tagpos@print@line@eval@continuous
1205   \else
1206     \eql@tagpos@print@line@eval@discrete
1207   \fi
1208 }

```

TODO: describe

```

1209 \def\eql@tagpos@print@line@eval@continuous{%
1210   \if@eqnsw
1211     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1212     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1213     \eql@tagpos@plain@\eql@tagpos@target@
1214     \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1215     -\ht\eql@tagbox@\relax
1216     \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1217     +\dp\eql@tagbox@\relax
1218     \ifnum\eql@row@=\@ne
1219       \@tempdima.5\maxdimen

```

```

1220 \fi
1221 \ifnum\eq@row@=\eq@totalrows@
1222 \@tempdimb-.5\maxdimen
1223 \fi
1224 \ifdim\eq@tagpos@plain@>\@tempdima
1225 \ifdim\eq@tagpos@plain@>\@tempdimb
1226 \ifdim\@tempdima>\@tempdimb
1227 \eq@tagpos@plain@\@tempdima
1228 \else
1229 \eq@tagpos@plain@\@tempdimb
1230 \fi
1231 \fi
1232 \else
1233 \ifdim\eq@tagpos@plain@<\@tempdimb
1234 \ifdim\@tempdima>\@tempdimb
1235 \eq@tagpos@plain@\@tempdimb
1236 \else
1237 \eq@tagpos@plain@\@tempdima
1238 \fi
1239 \fi
1240 \fi
1241 \ifnum\eq@tagpos@prevrow@> \z@
1242 \eq@tagpos@raised@\dimexpr\eq@line@height@+\dp\eq@tagbox@\relax
1243 \ifdim\eq@tagpos@raised@>\eq@tagpos@plain@\else
1244 \eq@tagpos@raised@\eq@tagpos@plain@
1245 \let\eq@tagpos@reserve\eq@false
1246 \fi
1247 \else
1248 \ifdim\eq@tagpos@target@>%
1249 \ifdefined\eq@tagsleft-1sp\relax\else\z@\fi
1250 \eq@tagpos@raised@\dimexpr\eq@line@height@+\dp\eq@tagbox@\relax
1251 \ifdim\eq@tagpos@raised@>\eq@tagpos@plain@\else
1252 \eq@tagpos@raised@\eq@tagpos@plain@
1253 \let\eq@tagpos@reserve\eq@false
1254 \fi
1255 \else
1256 \eq@tagpos@raised@\dimexpr-\eq@line@depth@
1257 -\ht\eq@tagbox@\relax
1258 \ifdim\eq@tagpos@raised@<\eq@tagpos@plain@\else
1259 \eq@tagpos@raised@\eq@tagpos@plain@
1260 \let\eq@tagpos@reserve\eq@false
1261 \fi
1262 \fi
1263 \fi
1264 \else
1265 \ifnum\eq@tagpos@prevrow@=\eq@row@
1266 \eq@tagwidth@\eq@tagwidth@block@
1267 \else
1268 \let\eq@tagpos@reserve\eq@false
1269 \fi
1270 \fi
1271 }

```

TODO: describe

```

1272 \def\eq@tagpos@print@line@eval@discrete{%
1273 \if@eqnsw
1274 \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@\relax
1275 \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@\relax

```

```

1276 \eql@tagpos@plain@\eql@tagpos@shift@
1277 \eql@tagpos@headroom@\z@
1278 \eql@tagpos@footroom@\z@
1279 \ifdim\eql@tagpos@shift@>%
1280 \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1281 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1282 \else
1283 \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1284 \fi
1285 \else
1286 \let\eql@tagpos@reserve\eql@false
1287 \fi
1288 }

```

TODO: describe

```

1289 \def\eql@tagpos@print@line@end{%
1290 \ifdefined\eql@tagpos@continuous
1291 \ifnum\eql@tagpos@prevrow@=\eql@row@
1292 \ifdefined\eql@tagpos@reserve
1293 \global\eql@appendexpand\eql@tags@container@block{%
1294 \advance\eql@tagpos@headroom@\the\dimexpr\eql@line@height@
1295 +\eql@line@depth@\relax\relax}%
1296 \eql@displaybreak@star\@M
1297 \fi
1298 \fi
1299 \fi
1300 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1301 \let\eql@SK@loaded\eql@false
1302 \let\eql@SK@label\@gobble
1303 \let\eql@SK@clearlabel\@empty
1304 \let\eql@SK@setlabel\@gobble
1305 \let\eql@SK@printlabel@right\@empty
1306 \let\eql@SK@printlabel@left\@empty
1307 \let\eql@SK@printlabel@line\@empty
1308 \def\eql@label@clean{\eql@label@org}
1309 \AddToHook{package/showkeys/after}{
1310 \let\eql@SK@loaded\eql@true
1311 \def\eql@SK@label#1{\SK@\SK@@label#1}
1312 \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1313 \eql@SK@clearlabel
1314 \def\eql@SK@@label#1>#2\SK@{%
1315 \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1316 }
1317 \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1318 \def\eql@SK@printlabel@right{%
1319 \ifx\eql@SK@lab\relax\else
1320 \rlap{\kern\marginparsep\eql@SK@lab}%
1321 \eql@SK@clearlabel
1322 \fi
1323 }
1324 \def\eql@SK@printlabel@left{%
1325 \ifx\eql@SK@lab\relax\else

```

```

1326     \llap{\eq@SK@lab\kern\marginparsep}%
1327     \eq@SK@clearlabel
1328   \fi
1329 }
1330 \def\eq@SK@printlabel@line{%
1331   \ifx\eq@SK@lab\relax\else
1332     \dimen@\prevdepth
1333     \nointerlineskip
1334     \ifdefined\eq@tagsleft
1335       \llap{%
1336         \eq@SK@lab
1337         \kern\marginparsep
1338       }%
1339     \eq@SK@clearlabel
1340   \else
1341     \rlap{%
1342       \dimen@\displaywidth
1343       \advance\dimen@\marginparsep
1344       \kern\dimen@
1345       \eq@SK@lab
1346     }%
1347   \fi
1348   \eq@SK@clearlabel
1349   \prevdepth\dimen@
1350 \fi
1351 }
1352 \let\eq@label@org\label
1353 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
1354 }

```

Labels.

`\eq@composetag@label` **TODO:** describe

```

1355 \def\eq@composetag@label{%
1356   \eq@SK@clearlabel
1357   \ifdefined\eq@tags@label
1358     \eq@SK@setlabel\eq@tags@label
1359   \ifdefined\eq@tags@name
1360     \let\@currentlabelname\eq@tags@name
1361   \else
1362     \let\@currentlabelname\eq@tags@name@generic
1363   \fi
1364   \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1365 \fi
1366 }

```

TODO: describe

```

1367 \def\eq@tags@printsubeqlabel{%
1368   \eq@tags@container@parent
1369   \ifdefined\eq@tags@label
1370     \eq@tags@makeblockanchor
1371     \eq@SK@setlabel\eq@tags@label
1372   \begingroup
1373     \def\@currentcounter{equation}%
1374     \eq@tags@anchor
1375     \let\@currentlabelname\eq@tags@name@generic
1376     \protected@edef\@currentlabel{\p@equation\theparentequation}%

```

```

1377     \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1378     \endgroup
1379     \eql@SK@printlabel@line
1380 \fi
1381 }

```

Hyperref Anchors. **TODO:** describe

```

1382 \let\eql@Hy@anchor\@gobble
1383 \AddToHook{package/hyperref/after}{
1384   \def\eql@Hy@anchor#1{%
1385     \Hy@raisedlink{\hyper@anchor{#1}}%
1386   }%
1387 }

```

TODO: describe

```

1388 \def\eql@tags@makeblockanchor{%
1389   \eql@tags@glabel@step
1390   \eql@Hy@anchor\eql@tags@glabel
1391   \edef\eql@tags@anchor{%
1392     \def\noexpand\thepage{\thepage}%
1393     \def\noexpand\@currentHref{\eql@tags@glabel}%
1394   }%
1395 }

```

TODO: describe

ql@composetag@anchor

```

1396 \def\eql@composetag@anchor{%
1397   \ifdefined\eql@tags@tag
1398     \def\@currentcounter{equation}%
1399     \ifdefined\eql@tags@ref
1400       \let\@currentlabel\eql@tags@ref
1401     \else
1402       \protected@edef\@currentlabel{\p@equation\eql@tags@tag}%
1403     \fi
1404     \eql@tags@glabel@step
1405     \edef\@currentHref{\eql@tags@glabel}%
1406     \eql@Hy@anchor\@currentHref
1407   \else
1408     \refstepcounter{equation}%
1409     \protected@edef\eql@tags@tag{\theequation}%
1410   \fi
1411   \eql@tags@anchor
1412 }

```

Tag Layout. **TODO:** describe

```

1413 \def\eql@tags@taglayout@set@direct#1{%
1414   \def\eql@tags@taglayout##1{#1}%
1415 }
1416 \def\eql@tags@taglayout@set#1{%
1417   \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1418 }

```

TODO: describe

```

1419 \def\eql@tags@tagform@set@direct#1{%
1420   \def\eql@tags@tagform##1{#1}%
1421 }
1422 \def\eql@tags@tagform@set#1#2#3{%
1423   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1424 }

1425 \eql@tags@taglayout@set{#1}
1426 \eql@tags@tagform@set({#1})
1427 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1428 \protected\def\tagform{\eql@tags@tagform}
1429 \protected\def\tagbox{\eql@tags@taglayout}
1430 \protected\def\tagboxed{\eql@tags@tagcompose}

```

`\eqref` `amsmath` defines the macro `\eqref` to refer to equation labels in a proper format. We provide it for completeness:

```

1431 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

`\eql@composetag@tag` **TODO:** describe

```

1432 \def\eql@composetag@tag{%
1433   \eql@tagging@tagbegin
1434   \eql@tags@frame@cmd{%
1435     \eql@tags@taglayout{%
1436       \eql@tags@tagform\eql@tags@tag
1437       \eql@tagging@tagsave
1438     }%
1439   }%
1440   \eql@tagging@tagend
1441 }

```

5.7 Tag Composition

TODO: describe

```

1442 \def\eql@composetag@measure{%
1443   \ifdefined\eql@tags@tag\else
1444     \stepcounter{equation}%
1445     \let\eql@tags@tag\theequation
1446   \fi
1447   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1448   \ifdefined\eql@numbering@multi
1449     \global\let\eql@tags@container\eql@tags@container@clear
1450   \fi
1451 }

```

TODO: describe

```

1452 \def\eql@composetag@print{%
1453   \eql@composetag@anchor
1454   \eql@composetag@label
1455   \ifdefined\eql@tags@left
1456     \eql@SK@printlabel@left
1457     \eql@composetag@tag
1458   \else
1459     \eql@composetag@tag
1460     \eql@SK@printlabel@right

```

```

1461 \fi
1462 \global\let\eql@tags@container\eql@tags@container@clear
1463 }

```

TODO: describe

TODO: one might still compare width to zero and pretend the tag is absent??

```

1464 \def\eql@tagbox@make#1{%
1465   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
1466   \eql@tagwidth@\wd\eql@tagbox@
1467   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1468     \eql@tagwidth@\eql@tagwidthmin@
1469   \fi
1470   \advance\eql@tagwidth@\eql@tagsepmin@
1471 }

```

TODO: describe

```

1472 \def\eql@tagbox@print@adjustheadroom{%
1473   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1474   \ifdim\dimen@>\z@
1475     \ifdim\dimen@>\eql@tagpos@headroom@
1476       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1477     \else
1478       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1479     \fi
1480   \fi
1481 }

```

TODO: describe

```

1482 \def\eql@tagbox@print@adjustfootroom{%
1483   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1484   \ifdim\dimen@>\z@
1485     \ifdim\dimen@>\eql@tagpos@footroom@
1486       \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1487     \else
1488       \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1489     \fi
1490   \fi
1491 }

```

TODO: describe

```

1492 \def\eql@tagbox@print@extendabove{%
1493   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1494   \ifdim\dimen@>\z@
1495     \global\eql@appendexpand\eql@display@container{%
1496       \eql@display@aboveextend@the\dimen@\relax}%
1497   \fi
1498 }

```

TODO: describe

```

1499 \def\eql@tagbox@print@extendbelow{%
1500   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1501   \ifdim\dimen@>\z@
1502     \global\eql@appendexpand\eql@display@container{%
1503       \eql@display@belowextend@the\dimexpr\dimen@\relax}%
1504   \fi
1505 }

```

TODO: describe

```
1506 \def\eql@tagbox@print@prepare{%
1507   \ifdefined\eql@tagpos@reserve
1508     \eql@tagpos@current@\eql@tagpos@plain@
1509   \else
1510     \eql@tagpos@current@\eql@tagpos@raised@
1511   \fi
1512   \ifdim\eql@tagpos@headroom@>\z@
1513     \eql@tagbox@print@adjustheadroom
1514   \fi
1515   \ifdim\eql@tagpos@footroom@>\z@
1516     \eql@tagbox@print@adjustfootroom
1517   \fi
1518   \ifnum\eql@row@=\@ne
1519     \eql@tagbox@print@extendabove
1520   \fi
1521   \ifnum\eql@row@=\eql@totalrows@
1522     \eql@tagbox@print@extendbelow
1523   \fi
1524 }
```

TODO: describe

```
1525 \def\eql@tagbox@print@tagsright{%
1526   \eql@tagbox@print@prepare
1527   \kern-\wd\eql@tagbox@
1528   \raise\eql@tagpos@current@\box\eql@tagbox@
1529 }
```

TODO: describe

```
1530 \def\eql@tagbox@print@tagsleft{%
1531   \eql@display@firstavail@set\z@
1532   \eql@tagbox@print@prepare
1533   \wd\eql@tagbox@\z@
1534   \raise\eql@tagpos@current@\box\eql@tagbox@
1535 }
```

$\eql@tagbox@print@cell$

```
1536 \def\eql@tagbox@print@cell{%
1537   \eql@tagging@tagaddbox
1538   \ifdefined\eql@tagsleft
1539     \ifdefined\eql@tagpos@reserve
1540       \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1541         \let\eql@tagpos@reserve\eql@false
1542       \fi
1543     \fi
1544     \if@eqnsw
1545       \eql@tagbox@print@tagsleft
1546     \fi
1547     \kern\displaywidth
1548   \else
1549     \kern\displaywidth
1550     \ifdefined\eql@tagpos@reserve
1551       \ifdim\eql@tagwidth@>%
1552         \dimexpr\displaywidth-\eql@line@width@+\eql@tagfuzz@\relax
1553       \let\eql@tagpos@reserve\eql@false
1554     \fi
1555   \fi
1556 }
```

```

1555 \fi
1556 \if@eqnsw
1557 \eq@tagbox@print@tagsright
1558 \fi
1559 \fi
1560 }

```

6 Subequation Numbering

We replicate the `amsmath` functionality to number a block of equations with a common number and a sub-numbering.

6.1 Definitions

`parentequation (counter)` We define a counter to store the main equation number while in subequation mode. It makes sense to share this definition with `amsmath` as `parentequation`, and we need to undefine it when `amsmath` is loaded at a later stage:

```

1561 \eq@amsmath@undefine\c@parentequation
1562 \eq@amsmath@undefine\theparentequation
1563 \ifdefined\c@parentequation\else
1564 \newcounter{parentequation}
1565 \fi

```

`subequations@template` We store a template which will be installed as `\theequation` in subequations mode: **TODO:** need to protect something?!

```

1566 \def\eq@subequations@template{\theparentequation\alph{equation}}

```

`@subequations@active` A boolean register which tells whether subequations are in use and thus must not be invoked again:

```

1567 \let\eq@subequations@active\eq@false

```

`eq@subequations@init` Low-level initialise the subequations mode. Store the equation counter in `\eq@subequations@restorecounter` for the case that no equation numbers will be used. Step the equation counter, copy it to `parentequation` and initialise `\theparentequation` (and its `hyperref` counterpart) with the expanded current value of `\theequation`; fill with tag data instead if a tag has been specified. Reset the equation counter and use the template for `\theequation`:

```

1568 \def\eq@subequations@init{%
1569 \edef\eq@subequations@restorecounter{%
1570 \global\c@equation\the\c@equation\relax}%
1571 \eq@tags@container@block
1572 \ifdefined\eq@tags@tag
1573 \eq@tags@glabel@step
1574 \protected@edef\theHparentequation{\eq@tags@glabel}%
1575 \protected@edef\theparentequation{\eq@tags@tag}%
1576 \else
1577 \advance\c@equation\@ne
1578 \protected@edef\theparentequation{\theequation}%
1579 \ifdefined\theHequation
1580 \protected@edef\theHparentequation{\theHequation}%
1581 \fi

```

```

1582 \fi
1583 \global\c@parentequation\c@equation
1584 \global\c@equation\z@
1585 \let\theequation\eq@subequations@template
1586 \def\theHequation{\theHparentequation.\arabic{equation}}%
1587 }

```

1@subequations@close Low-level close the subequations mode. If no number has been used, return to the original equation counter, otherwise use the value stored in `parentequation`. Note that we cannot use `\setcounter` here because the `calc` version would involve actions which are not allowed after `\halign` within a display equation:

```

1588 \def\eq@subequations@close{%
1589 \ifnum\c@equation=\z@
1590 \eq@subequations@restorecounter
1591 \else
1592 \global\c@equation\c@parentequation
1593 \fi
1594 }

```

6.2 Environment

1@subequations@start Start the subequations environment with optional parameters in #1. Enter subequations mode and set an anchor for subsequent `\label`'s. Manually print the `showkeys` tag:

TODO: join with other similar anchor routines `\eq@tags@printsbeqlabel`

```

1595 \def\eq@subequations@start{%
1596 \let\eq@tags@container@block\eq@tags@container@clear
1597 \eq@nextopt@process{subequations}%
1598 \eq@subequations@init
1599 \eq@tags@glabel@step
1600 \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1601 \eq@Hy@anchor\eq@subequations@currentHref
1602 \edef\eq@subequations@thepage{\thepage}%
1603 \def\@currentcounter{equation}%
1604 \let\@currentHref\eq@subequations@currentHref
1605 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1606 \eq@tags@container@block
1607 \ifdefined\eq@tags@name
1608 \let\@currentlabelname\eq@tags@name
1609 \else
1610 \let\@currentlabelname\eq@tags@name@generic
1611 \fi
1612 \let\eq@subequations@active\eq@true
1613 \ifdefined\eq@tags@label
1614 \eq@SK@label\eq@tags@label
1615 \fi
1616 \ignorespaces
1617 }

```

eq@subequations@end End the subequations environment. Issue the label if one has been specified and an equation number has actually been used. Then close subequations mode:

```

1618 \def\eq@subequations@end{%
1619 \ifnum\c@equation>\z@
1620 \eq@tags@container@block
1621 \ifdefined\eq@tags@label
1622 \begingroup

```

```

1623      \def\@currentcounter{equation}%
1624      \let\thepage\eq\@subequations\thepage
1625      \let\@currentHref\eq\@subequations\@currentHref
1626 % \TODO how about tag* ?! also within equations!
1627      \protected@edef\@currentlabel{\p@equation\theparentequation}%
1628      \ifdefined\eq\@tags\@name
1629          \let\@currentlabelname\eq\@tags\@name
1630      \else
1631          \let\@currentlabelname\eq\@tags\@name@generic
1632      \fi
1633      \expandafter\eq\@label@clean\expandafter{\eq\@tags\@label}%
1634      \endgroup
1635      \fi
1636  \fi
1637  \eq\@subequations@close
1638  \ignorespacesafterend
1639 }

```

`subequations` (*env.*) The subequations environment tests for optional parameters and passes on to the start and end routines:

```

1640 \newenvironment{eq\@subequations}{%
1641   (dev)\eq\@dev@enterenv
1642   \eq\@subequations@testall\eq\@subequations@start%
1643 }{%
1644   \eq\@subequations@end
1645 (dev)\eq\@dev@leaveenv
1646 }

```

TODO: describe

```

1647 \def\eq\@subequations@testall{\eq\@parseopt\eq\@subequations@parseopt}
1648 \def\eq\@subequations@parseopt{%
1649   \ifx\eq\@parseopt@token[%]
1650     \let\eq\@parseopt@next\eq\@parseopt@opt
1651   \fi
1652   \ifx\eq\@parseopt@token\eq\@atxi
1653     \let\eq\@parseopt@next\eq\@parseopt@label
1654   \fi
1655   \ifx\eq\@parseopt@token\eq\@atxii
1656     \let\eq\@parseopt@next\eq\@parseopt@label
1657   \fi
1658   \ifx\eq\@parseopt@token\label
1659     \let\eq\@parseopt@next\eq\@parseopt@end
1660   \fi
1661 }

```

6.3 Subequation Scheme

TODO: describe

```

1662 \def\eq\@numbering@subeq@init{%
1663   \let\eq\@save@theequation\theequation
1664   \let\eq\@save@theHequation\theHequation
1665   \eq\@subequations@init
1666   \let\eq\@tags@container@parent\eq\@tags@container@block
1667   \let\eq\@tags@container@block\eq\@tags@container@clear
1668 }

```

TODO: describe

```
1669 \def\eq@numbering@subeq@test{%
1670   \ifnum\eq@tagrows@<\tw@
1671     \let\eq@tags@container@block\eq@tags@container@parent
1672     \let\eq@numbering@subeq@use\eq@false
1673     \let\theequation\eq@save@theequation
1674     \let\theHequation\eq@save@theHequation
1675     \eq@subequations@restorecounter
1676   \fi
1677 }
```

TODO: describe

```
1678 % \TODO note must not use setcounter here (when calc is loaded)
1679 \def\eq@numbering@subeq@close{%
1680   \eq@subequations@close
1681 }
```

7 Display Equations Support

TODO: describe

```
1682 \let\eq@display@injectbefore\@undefined
1683 \let\eq@display@injectafter\@undefined
1684 \let\eq@interline@container\@undefined
1685 \def\eq@interline@container@clear{%
1686   \eq@displaybreak@pen@\@MM
1687   \eq@vspaceskip@\z@skip
1688 }
```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```
1689 \interdisplaylinepenalty\@M
```

\eq@getdsp@pen **TODO:** isn't this the opposite order than \@getpen?!

```
1690 \def\eq@getdsp@pen#1{%
1691   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1692 }
```

TODO: allow a displaybreak before equations

```
1693 \protected\def\eq@displaybreak@default{%
1694   \eq@warning{Invalid use of \string\displaybreak}{}}%
1695   \eq@teststaroropt@loose\@gobble\eq@gobbleopt{}}
1696 \eq@amsmath@after{\let\eq@displaybreak@default\displaybreak}
1697 \eq@amsmath@let\displaybreak\eq@displaybreak@default

1698 \newcount\eq@displaybreak@pen@
1699 \newcount\eq@displaybreak@prepen@
1700 \newcount\eq@displaybreak@postpen@
```

TODO: describe

```
1701 \protected\def\eqldisplaybreak{%
1702   \relax
1703   \eql@ampproTECTtwo\eql@teststaroropt@tight
1704   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1705 }

1706 \def\eqldisplaybreak@star#1{%
1707   \global\eql@appendexpand\eql@interline@container{%
1708     \eqldisplaybreak@open@the\numexpr#1\relax\relax}%
1709 }

1710 \def\eqldisplaybreak@level[#1]{%
1711   \ifnum#1<\z@
1712     \global\eql@append\eql@interline@container{\eqldisplaybreak@open@MM}%
1713   \else
1714     \global\eql@appendexpand\eql@interline@container{%
1715       \eqldisplaybreak@open@-\@getpen{#1}\relax}%
1716   \fi
1717 }
```

TODO: describe

```
1718 \def\eqldisplaybreak@pre#1{%
1719   \ifnum#1<\z@
1720     \eqldisplaybreak@prepen@MM
1721   \else
1722     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1723   \fi
1724 }
```

TODO: describe

```
1725 \def\eqldisplaybreak@post#1{%
1726   \ifnum#1<\z@
1727     \eqldisplaybreak@postpen@MM
1728   \else
1729     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1730   \fi
1731 }
```

TODO: describe

```
1732 \def\eqldisplaybreak@inter#1{%
1733   \ifnum#1<\z@
1734     \interdisplaylinepenalty\@M
1735   \else
1736     \interdisplaylinepenalty\eql@getdsp@open{#1}\relax
1737   \fi
1738 }
```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```
1739 \newskip\eqlvspaceskip@
```

```

1740 \let\eql@vspace@org\vspace
1741 \def\eql@vspace{%
1742   \ifvmode
1743     \expandafter\eql@vspace@immediate
1744   \else
1745     \expandafter\eql@vspace@line
1746   \fi
1747 }

```

TODO: `\eql@vspace@addfixedafter` on last line has no effect. should apply outside environment

```

1748 \def\eql@vspace@line{%
1749   \eql@ifstar@loose\eql@vspace@addfixedbefore\eql@vspace@add
1750 }
1751 \def\eql@vspace@add#1{%
1752   \global\eql@appendexpand\eql@interline@container{%
1753     \advance\eql@vspaceskip@\the\glueexpr#1\relax\relax}}
1754 \def\eql@vspace@addfixedbefore#1{%
1755   \global\eql@appendexpand\eql@interline@container{%
1756     \noexpand\eql@append\noexpand\eql@display@injectbefore{%
1757       \skip@\the\glueexpr#1\relax\relax
1758       \penalty\@M
1759       \vskip\skip@
1760     \global\advance\eql@line@interline@\skip@
1761     }%
1762   }%
1763 }
1764 \def\eql@vspace@addfixedafter#1{%
1765   \global\eql@appendexpand\eql@interline@container{%
1766     \noexpand\eql@append\noexpand\eql@display@injectafter{%
1767       \dimen@\prevdepth
1768       \hrule\@height\z@
1769       \skip@\the\glueexpr#1\relax\relax
1770       \penalty\@M
1771       \vskip\skip@
1772     \global\advance\eql@line@interline@\skip@
1773     \prevdepth\dimen@
1774     }%
1775   }%
1776 }

```

TODO: careful to not expand `\eql@display@container` after measure

```

1777 \def\eql@vspace@immediate{%
1778   \noalign\bgroup
1779     \eql@ifstar@loose\eql@vspace@fixed\eql@vspace@discardable
1780 }
1781 \def\eql@vspace@fixed#1{%
1782   \skip@\glueexpr#1\relax
1783   \ifnum\eql@row@=\@ne
1784     \global\eql@appendexpand\eql@display@container{%
1785       \advance\eql@abovespace@\the\skip@\relax}%
1786   \else\ifnum\eql@row@>\eql@totalrows@
1787     \global\eql@appendexpand\eql@display@container{%
1788       \advance\eql@belowspace@\the\skip@\relax}%
1789   \else
1790     \dimen@\prevdepth
1791     \hrule\@height\z@
1792     \penalty\@M

```

```

1793     \vskip\skip@
1794     \global\advance\eq@line@interline@\skip@
1795     \prevdepth\dimen@
1796     \fi\fi
1797   \egroup
1798 }
1799 \def\eq@vspace@discordable#1{%
1800   \skip@\glueexpr#1\relax
1801   \ifnum\eq@row@=\@ne
1802     \global\eq@appendexpand\eq@display@container{%
1803       \advance\eq@abovespace@\the\skip@\relax}%
1804   \else\ifnum\eq@row@>\eq@totalrows@
1805     \global\eq@appendexpand\eq@display@container{%
1806       \advance\eq@belowspace@\the\skip@\relax}%
1807   \else
1808     \vskip\skip@
1809     \global\advance\eq@line@interline@\skip@
1810     \fi\fi
1811   \egroup
1812 }

```

7.3 Default Vertical Spacing

TODO: describe

`\eq@strut` Next follows a special internal strut which is supposed to match the height and the depth `\eq@strutbox@` of a normal `\strut` minus `\normallineskiplimit` according to M. Spivak.

```

1813 \newbox\eq@strutbox@
1814 \def\eq@strut@depth{.3}
1815 \def\eq@strut{\copy\eq@strutbox@}
1816 \let\eq@strut@cell\eq@strut
1817 \let\eq@strut@tag\eq@strut
1818 \def\eq@strut@make{%
1819   \setbox\eq@strutbox@\hbox{%
1820     \@tempdimb\dimexpr
1821       \eq@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1822     \@tempdima\dimexpr
1823       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1824     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1825   }
1826 }
1827 \AtBeginDocument{\eq@strut@make}

```

TODO: describe

```

1828 \def\eq@spread@set{%
1829   \eq@spread@\dimexpr\glueexpr\eq@spread@val\relax
1830   +\normalbaselineskip-\baselineskip\relax
1831   \ifdim\eq@spread@>\z@
1832     \openup\eq@spread@
1833     \ifdefined\spread@equation
1834       \let\spread@equation\@empty
1835     \fi
1836   \fi
1837 }

```

7.4 Entry and Exit

TODO: describe

```
1838 \let\eql@beamerbasecolor@fix@empty
1839 \AddToHook{package/beamerbasecolor/after}{%
1840   \def\eql@beamerbasecolor@fix{%
1841     \donotcolorouterdisplaymaths
1842     \donotcoloroutermaths
1843     \beamer@setdisplaymathcolor
1844   }%
1845 }
```

`\eql@abovespace@` (*skip*)

`\eql@belowspace@` (*skip*)

```
1846 \newskip\eql@abovespace@
1847 \newskip\eql@belowspace@
```

`\eql@display@enter`

```
1848 \def\eql@display@enter{%
1849   \if@noskipsec\leavevmode\par\fi
1850   \ifvmode
1851     \eql@prevdepth@\prevdepth
1852     \nointerlineskip
1853     \noindent
1854   \else
1855     \eql@prevdepth@\maxdimen
1856   \fi
1857   \eql@beamerbasecolor@fix
1858 }
```

`\eql@display@adjust`

```
1859 \def\eql@display@adjust{%
1860   \ifdefined\eql@display@linewidth
1861     \displaywidth\glueexpr\eql@display@linewidth\relax
1862     \advance\displaywidth-\displayindent
1863   \fi
1864   \ifdefined\eql@display@marginleft
1865     \advance\displaywidth\displayindent
1866     \displayindent\glueexpr\eql@display@marginleft\relax
1867     \advance\displaywidth-\displayindent
1868   \fi
1869   \ifdefined\eql@display@marginright
1870     \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1871   \fi
1872   \ifdim\displaywidth<\z@
1873     \displaywidth\z@
1874   \fi
1875 }
```

`\eql@display@init`

```
1876 \def\eql@display@init{%
1877   \let\displaybreak\eql@displaybreak
1878   \let\eql@vspace@org\vspace
1879   \let\vspace\eql@vspace
1880   \let\eqncontrol\eql@control
```

```

1881 \let\eqldisplay@injectbefore\@empty
1882 \let\eqldisplay@injectafter\@empty
1883 \eql@spread@set
1884 \eql@strut@make
1885 \let\eql@frame@cmd\@undefined
1886 }

```

\eqldisplay@print

```

1887 \def\eqldisplay@print{%
1888 \let\eqldisplay@container\@empty
1889 \eqldisplay@firstavail@z@
1890 \eqldisplay@aboveextend@z@
1891 \eqldisplay@belowextend@z@
1892 \global\let\eql@interline@container\eql@interline@container@clear
1893 }

```

@display@halign@init **TODO:** describe

```

1894 \def\eqldisplay@halign@init#1{%
1895 \eql@row@z@
1896 \eql@prevgraf@\prevgraf
1897 \everycr{\noalign{%
1898 \global\advance\eql@row@\@ne
1899 \prevgraf\numexpr\prevgraf+\@ne\relax
1900 #1%
1901 }}%
1902 }

```

TODO: how about penalty here? not for entry into display

```

1903 \def\eqldisplay@halign@start{%
1904 \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1905 \ifdim\eql@prevdepth=\maxdimen\else
1906 \prevdepth\eql@prevdepth@
1907 \fi
1908 \ifdim\prevdepth=-\@m\p@else
1909 \ifdefined\eqldisplay@height
1910 \skip@\baselineskip
1911 \advance\skip@-\glueexpr\eqldisplay@height\relax
1912 \advance\skip@-\prevdepth\relax
1913 \ifdim\skip@<\lineskiplimit
1914 \skip@\lineskip
1915 \fi
1916 \advance\skip@-\eql@spread@\relax
1917 \vskip\skip@
1918 \nointerlineskip
1919 \else
1920 \vskip-\eql@spread@\relax
1921 \fi
1922 \fi
1923 }

```

TODO: describe

```

1924 \def\eqldisplay@vspace{%
1925 \advance\abovedisplayskip\eql@abovespace@
1926 \advance\belowdisplayskip\eql@belowspace@
1927 }

```

TODO: describe

```
1928 \def\eqldisplay@vspace@native{%
1929   \advance\abovedisplayskip\eql@abovespace@
1930   \advance\belowdisplayskip\eql@belowspace@
1931   \advance\abovedisplayshortskip\eql@abovespace@
1932   \advance\belowdisplayshortskip\eql@belowspace@
1933 }
```

TODO: describe

```
1934 \def\eqldisplay@penalty{%
1935   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1936     \postdisplaypenalty\eqldisplaybreak@postpen@
1937   \fi
1938   \ifnum\eqldisplaybreak@pen@=\@MM\else
1939     \postdisplaypenalty\eqldisplaybreak@pen@
1940   \fi
1941   \ifnum\eqldisplaybreak@prepen@=\@MM\else
1942     \predisplayskip\eqldisplaybreak@prepen@
1943   \fi
1944 }
```

TODO: describe **TODO:** issue: `\vspace*{0pt}` has some effect if page is broken here

```
1945 \def\eqldisplay@halign@end{%
1946   \eql@interline@container
1947   \eqldisplay@injectbefore
1948   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
1949   \ifdefined\eqldisplay@depth
1950     \prevdepth\glueexpr\eqldisplay@depth\relax
1951   \fi
1952 }
```

`\eqldisplay@close` **TODO:** there seems to be an offset of 1em in `\predisplaysize` towards actual content, nice.

TODO: must not use `\setlength` or `\setcounter` when `\calc` is loaded **TODO:** do we actually need penalty adjustments in case of paragraphs above or below?

```
1953 \def\eqldisplay@close{%
1954   \eqldisplay@container
1955   \ifdim\eqldisplay@firstavail@<\z@
1956     \eqldisplay@firstavail@\z@
1957   \fi
1958   \eql@skip@mode@leave@\z@
1959   \ifdim\eql@prevdepth@=\maxdimen
1960     \ifdim\predisplaysize=-\maxdimen
1961       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
1962       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
1963     \else
1964       \eql@skip@mode@above@\z@
1965       \eql@skip@mode@below@\z@
1966       \advance\eqldisplay@firstavail@\displayindent
1967       \ifdim\eqldisplay@firstavail@>\predisplaysize
1968         \ifcase\eql@skip@mode@short\relax
1969           \or
1970             \eql@skip@mode@above@\@ne
1971           \or
1972             \eql@skip@mode@above@\@ne
1973           \ifnum\eql@totalrows@=\@ne
1974             \eql@skip@mode@below@\@ne
```

```

1975         \fi
1976     \or
1977         \eqL@skip@mode@above@ \@ne
1978         \eqL@skip@mode@below@ \@ne
1979     \fi
1980 \fi
1981 \fi
1982 \else
1983     \ifdim\eqL@prevdepth@=-\@m\p@
1984         \eqL@skip@mode@above@\eqL@skip@mode@top@above\relax
1985         \eqL@skip@mode@below@\eqL@skip@mode@top@below\relax
1986     \else
1987         \eqL@skip@mode@above@\eqL@skip@mode@par@above\relax
1988         \eqL@skip@mode@below@\eqL@skip@mode@par@below\relax
1989     \fi
1990 \fi
1991 \ifcase\eqL@skip@mode@above@
1992 \or\or\or
1993     \predisplaypenalty\z@
1994 \or
1995     \predisplaypenalty\z@
1996 \fi
1997 \ifcase\eqL@skip@mode@below@
1998 \or\or\or
1999     \eqL@skip@mode@leave@ \@ne
2000 \or
2001     \eqL@skip@mode@leave@ \tw@
2002 \fi
2003 \ifdefined\eqL@skip@force@above
2004     \eqL@skip@mode@above@\eqL@skip@force@above\relax
2005 \fi
2006 \ifdefined\eqL@skip@force@below
2007     \eqL@skip@mode@below@\eqL@skip@force@below\relax
2008 \fi
2009 \ifdefined\eqL@skip@force@leave
2010     \eqL@skip@mode@leave@\eqL@skip@force@leave\relax
2011 \fi
2012 \ifnum\eqL@skip@mode@leave@>\z@
2013     \postdisplaypenalty\z@
2014 \fi
2015 \ifcase\eqL@skip@mode@above@
2016     \abovedisplayskip\glueexpr\eqL@skip@long@above\relax
2017 \or
2018     \abovedisplayskip\glueexpr\eqL@skip@short@above\relax
2019 \or
2020     \abovedisplayskip\glueexpr\eqL@skip@cont@above\relax
2021 \or
2022     \abovedisplayskip\glueexpr\eqL@skip@par@above\relax
2023 \or
2024     \abovedisplayskip\glueexpr\eqL@skip@top@above\relax
2025 \or
2026     \abovedisplayskip\z@skip
2027 \or
2028     \abovedisplayskip\glueexpr\eqL@skip@med@above\relax
2029 \or
2030     \abovedisplayskip\glueexpr\eqL@skip@custom@above\relax
2031 \fi
2032 \ifcase\eqL@skip@mode@below@

```

```

2033 \belowdisplayskip\glueexpr\eq\skip@long@below\relax
2034 \or
2035 \belowdisplayskip\glueexpr\eq\skip@short@below\relax
2036 \or
2037 \belowdisplayskip\glueexpr\eq\skip@cont@below\relax
2038 \or
2039 \belowdisplayskip\glueexpr\eq\skip@par@below\relax
2040 \or
2041 \belowdisplayskip\glueexpr\eq\skip@top@below\relax
2042 \or
2043 \belowdisplayskip\z@skip
2044 \or
2045 \belowdisplayskip\glueexpr\eq\skip@med@below\relax
2046 \or
2047 \belowdisplayskip\glueexpr\eq\skip@custom@below\relax
2048 \fi
2049 \global\eq\skip@mode@leave@\eq\skip@mode@leave@
2050 \eq\interline@container
2051 \advance\eq\belowspace@\eq\vspaceskip@
2052 \eq\display@penalty
2053 \eq\display@vspace
2054 \skip@\glueexpr\eq\skip@tag@above\relax
2055 \ifdim\skip@>\abovedisplayskip
2056 \skip@\abovedisplayskip
2057 \fi
2058 \advance\abovedisplayskip-\eq\display@aboveextend@\relax
2059 \ifdim\abovedisplayskip<\skip@
2060 \abovedisplayskip\skip@
2061 \fi
2062 \skip@\glueexpr\eq\skip@tag@below\relax
2063 \ifdim\skip@>\belowdisplayskip
2064 \skip@\belowdisplayskip
2065 \fi
2066 \ifdim\eq\display@belowextend@>\z@
2067 \advance\belowdisplayskip-\eq\display@belowextend@\relax
2068 \ifdim\belowdisplayskip<\skip@
2069 \belowdisplayskip\skip@
2070 \fi
2071 \fi
2072 }

```

TODO: describe

```

2073 \def\eq\display@leave{%
2074 \prevgraf\eq\prevgraf@
2075 \ifcase\eq\skip@mode@leave@
2076 \or
2077 \endgraf
2078 \or
2079 \endgraf
2080 \prevdepth-\@m\p@
2081 \fi
2082 }

```

TODO: describe

```

2083 \def\eq\display@nest{%
2084 \let\displaybreak\eq\displaybreak@default
2085 \let\intertext\eq\intertext@default
2086 \let\vspace\eq\vspace@org

```

2087 }

TODO: describe

```
2088 \def\eqldisplay@restore{%
2089   \let\label\eqldlabel@org
2090   \let\tag\eqldtag@default
2091   \let\raisetag\eqldraisetag@default
2092   \let\displaybreak\eqldisplaybreak@default
2093   \let\intertext\eqldintertext@default
2094   \let\vspace\eqldvspace@org
2095 }
```

TODO: describe

```
2096 \eqldappend\@arrayparboxrestore{%
2097   \eqlddisplay@restore
2098   \ifdefined\eqldampproof@active
2099     \eqldampervert
2100   \fi
2101   \@displayfalse
2102 }
```

7.5 Stack

TODO: describe **TODO:** for each global variable declare global nature at its definition!

TODO: we must be consistent about global variables vs local variables global variables need to be saved at every level where they may be modified (even if modified only locally)

```
2103 \def\eqldstack@enable{%
2104   \let\eqldstack@save@equations\eqldstack@save@equations@
2105   \let\eqldstack@save@box\eqldstack@save@box@
2106 }
```

TODO: describe

```
2107 \let\eqldstack@save@equations\eqldstack@enable
2108 \let\eqldstack@save@box\eqldstack@enable
2109 \let\eqldstack@restore\@empty
```

TODO: describe

```
2110 \def\eqldstack@save@reg#1{\global#1\the#1\relax}
2111 \def\eqldstack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

TODO: further global variables: global registers: \eqldnextopt, \eqldtags@glabel@ used locally without possibility of change between setting and retrieving:

\eqldprevgraf@, \eqldskip@mode@leave@, \eqldshape@lastrow, \eqldframe@prevcmd

TODO: to be reviewed: \eqldintertext@after, \eqldintertext@opt **TODO:** describe

```
2112 \def\eqldstack@save@equations@{%
2113   \let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit
2114   \let\eqldstack@cell@container\eqldcell@container
2115   \let\eqldstack@tags@container\eqldtags@container
2116   \let\eqldstack@interline@container\eqldinterline@container
2117   \let\eqldstack@block@container\eqlddisplay@container
2118   \let\eqldstack@dimensions@tab\eqlddimensions@tab
2119   \edef\eqldstack@restore{%
2120     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2121     \eqldstack@save@let\eqldstack@numbering@eqnswinit\eqldnumbering@eqnswinit
```

```

2122 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2123 \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2124 \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2125 \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2126 \eql@stack@save@let\eql@stack@block@container\eql@display@container
2127 \eql@stack@save@reg\eql@column@
2128 \eql@stack@save@reg\eql@totalcolumns@
2129 \eql@stack@save@reg\eql@line@avail@
2130 \eql@stack@save@reg\eql@line@pos@
2131 \eql@stack@save@reg\eql@line@width@
2132 \eql@stack@save@reg\eql@line@depth@
2133 \eql@stack@save@reg\eql@line@height@
2134 \eql@stack@save@reg\eql@line@prevdepth@
2135 \eql@stack@save@reg\eql@line@interline@
2136 \eql@stack@save@reg\eql@totalheight@
2137 \eql@stack@save@reg\eql@tagwidth@max@
2138 \eql@stack@save@reg\eql@tagpos@row@
2139 \eql@stack@save@reg\eql@row@
2140 \eql@stack@save@reg\eql@tagrows@
2141 }%
2142 }

```

TODO: describe

```

2143 \def\eql@stack@save@box@{%
2144 \let\eql@stack@cell@container\eql@cell@container
2145 \edef\eql@stack@restore{%
2146 \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2147 \eql@stack@save@reg\eql@row@
2148 }%
2149 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2150 \def\eql@measure@init#1#2{%
2151 \eql@dimensions@reset
2152 \let\eql@display@container\@empty
2153 \eql@numbering@measure@init
2154 \eql@row@\z@
2155 \eql@totalheight@\z@
2156 \eql@totalrows@\@M
2157 \eql@line@prevdepth@-\@m\p@
2158 \eql@line@interline@\z@
2159 \tabskip\z@skip
2160 \everycr{\noalign{%
2161 \global\advance\eql@row@\@ne
2162 #1%
2163 }}%
2164 \global\let\eql@interline@container\eql@interline@container@clear
2165 \eql@measure@savestate
2166 \eql@display@halign@letcr{#2}%

```

2167 }

TODO: describe

```
2168 \def\eq@measure@tag{%
2169   \eq@tagwidth@z@
2170   \ifdefined\eq@numbering@multi
2171     \if@eqnsw
2172       \eq@tags@container
2173       \eq@tagbox@make\eq@composetag@measure
2174       \ifdefined\eq@tagpos@reserve\else
2175         \eq@tagwidth@z@
2176       \fi
2177     \fi
2178   \fi
2179 }
```

TODO: describe

```
2180 \def\eq@measure@endrow{%
2181   \ifdim\eq@line@prevdepth@=-\m@p\else
2182     \dimen@dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@relax
2183     \ifdim\dimen@<\lineskiplimit
2184       \dimen@lineskip
2185     \fi
2186     \advance\eq@line@interline@\dimen@
2187   \fi
2188   \eq@dimensions@endrow
2189   \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2190     \global\eq@tagwidth@max@\eq@tagwidth@
2191   \fi
2192   \ifdim\eq@tagwidth@>z@
2193     \global\advance\eq@tagrows@one
2194   \fi
2195   \global\advance\eq@totalheight@dimexpr
2196     \eq@line@interline@+\eq@line@height@+\eq@line@depth@
2197   \global\eq@line@interline@z@
2198   \global\eq@line@prevdepth@\eq@line@depth@
2199 }
```

TODO: describe

```
2200 \def\eq@measure@close{%
2201   \advance\eq@row@-\tw@
2202   \eq@totalrows@\eq@row@
2203   \ifnum\eq@totalrows@>z@
2204     \eq@dimensions@get@one
2205     \eq@topheight@dimexpr\eq@line@height@+\eq@line@interline@relax
2206     \eq@dimensions@get\eq@totalrows@
2207     \eq@bottomdepth@\eq@line@depth@
2208   \fi
2209   \eq@numbering@measure@blocktag
2210   \begingroup
2211     \eq@tags@container
2212     \if@eqnsw
2213       \eq@tagbox@make\eq@composetag@measure
2214       \ifdefined\eq@tagpos@reserve\else
2215         \eq@tagwidth@z@
2216       \fi
2217     \eq@dimensions@saveblocktag
```

```

2218     \else
2219     \eqldimensions@savenoblocktag
2220     \eqlnumbering@warnunused
2221     \fi
2222 \endgroup
2223 \eqldimensions@get\z@
2224 \eql@measure@restorestate
2225 }

```

measure@restorestate

ql@measure@savestate

```

2226 \let\eql@measure@restorestate\@empty
2227 \def\eql@measure@savestate{%
2228   \begingroup
2229   \def\@elt##1{%
2230     \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2231     \global\edef\@gtempa{\cl@ckpt}%
2232   \endgroup
2233   \let\eql@measure@restorestate\@gtempa
2234 }

```

8.2 Line Breaks

TODO: describe

\eqldisplay@cr

```

2235 \protected\def\eqldisplay@cr{\eqlsrbgroup
2236   \eql@ifnextchar@tight~%
2237   {\numbernext\let\eql@punct@line\@empty\eqldisplay@cr@star}%
2238   \eqldisplay@cr@star
2239 }

```

\eqldisplay@cr@star

```

2240 \def\eqldisplay@cr@star{%
2241   \eql@teststaropt@tight{%
2242     \global\eql@append\eql@interline@container{\eqldisplaybreak@pen@\@M}%
2243     \eqldisplay@cr@opt}%
2244   \eqldisplay@cr@opt\z@skip
2245 }

```

\eqldisplay@cr@opt

```

2246 \def\eqldisplay@cr@opt[#1]{\eqlsregroup
2247   \eqldisplay@endline
2248   \cr

2249   \noalign{%
2250     \eql@interline@container
2251     \eqldisplay@injectbefore
2252     \ifnum\eqldisplaybreak@pen@=\@MM
2253       \penalty\interdisplaylinepenalty
2254     \else
2255       \penalty\eqldisplaybreak@pen@
2256     \fi
2257     \advance\eql@vspaceskip@\glueexpr#1\relax
2258     \vskip\eql@vspaceskip@

```

```

2259 \global\advance\eql@line@interline@\eql@vspaceskip@
2260 \eql@display@injectafter
2261 \global\let\eql@interline@container\eql@interline@container@clear
2262 }%
2263 }

```

display@halign@letcr

```

2264 \def\eql@display@halign@letcr#1{%
2265 \let\\ \eql@display@cr
2266 \let\eql@display@endline#1%
2267 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2268 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2269 \eql@amsmath@let\intertext\eql@intertext@default

```

TODO: why does it fail in measuring? total width?! determine total width otherwise!

```

2270 \def\eql@intertext@process{%
2271 \eql@display@endline
2272 \cr
2273 \ifmeasuring@
2274 \expandafter\@gobble
2275 \else
2276 \expandafter\eql@intertext@print
2277 \fi
2278 }

```

TODO: describe **TODO:** prevdepth **TODO:** does this have to be in a vbox? **TODO:** vskip and penalty opposite order **TODO:** can we handle short? certainly needs two passes

```

2279 \def\eql@intertext@print#1{%
2280 \noalign{%
2281 \eql@display@halign@end
2282 \let\eql@skip@force@below\z@
2283 \let\eql@skip@force@above\z@
2284 \eql@setkeys{intertext}\eql@intertext@opt
2285 \openup-\eql@spread@
2286 \penalty\postdisplaypenalty
2287 \ifcase\eql@skip@force@below\relax
2288 \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2289 \or
2290 \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2291 \or
2292 \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2293 \or
2294 \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2295 \or
2296 \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2297 \or
2298 \advance\eql@vspaceskip@\z@skip
2299 \or
2300 \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2301 \or

```

```

2302     \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2303 \fi
2304 \vskip\eql@vspaceskip@
2305 \global\let\eql@interline@container\eql@interline@container@clear
2306 \vbox{%
2307     \@parboxrestore
2308     \ifdim
2309         \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2310     \else
2311         \parshape\@ne
2312         \@totalleftmargin\linewidth
2313     \fi
2314     \noindent
2315     \prevgraf\eql@prevgraf@
2316     \ignorespaces
2317     #1%
2318     \par
2319     \global\eql@prevgraf@\prevgraf
2320 }%
2321 \penalty\predisplaypenalty
2322 \ifcase\eql@skip@force@above\relax
2323     \vskip\glueexpr\eql@skip@long@above\relax
2324 \or
2325     \vskip\glueexpr\eql@skip@short@above\relax
2326 \or
2327     \vskip\glueexpr\eql@skip@cont@above\relax
2328 \or
2329     \vskip\glueexpr\eql@skip@par@above\relax
2330 \or
2331     \vskip\glueexpr\eql@skip@top@above\relax
2332 \or
2333     \vskip\z@skip
2334 \or
2335     \vskip\glueexpr\eql@skip@med@above\relax
2336 \or
2337     \vskip\glueexpr\eql@skip@custom@above\relax
2338 \fi
2339 % \eql@prevdepth@\maxdimen
2340 \eql@prevdepth@\z@
2341 \eql@display@halign@start
2342 }
2343 }

```

TODO: describe

```

2344 \newenvironment{eql@intertext}{%
2345     \eql@testopt@tight\eql@intertext@{}%
2346 }{%
2347     \aftergroup\eql@intertext@after
2348     \ignorespacesafterend
2349 }

```

TODO: describe

```

2350 \def\eql@intertext@env{intertext}
2351 \def\eql@intertext@[#1]{%
2352     \global\def\eql@intertext@opt{#1}%
2353     \ifx\@currenvir\eql@intertext@env
2354         \expandafter\eql@scan@env\expandafter\eql@intertext@inject
2355     \else

```

```

2356 \expandafter\eql@intertext@process
2357 \fi
2358 }

```

TODO: describe

```

2359 \def\eql@intertext@inject{%
2360 \global\edef\eql@intertext@after{%
2361 \noexpand\eql@intertext@process{%
2362 \ifx\eql@scan@body\eql@scan@body@dump
2363 \eql@scan@body@dump
2364 \else
2365 \noexpand\scantokens{\eql@scan@body@dump}%
2366 \fi
2367 }%
2368 }%
2369 }

```

8.4 Line Marks

TODO: describe

```

2370 \def\eql@markline@pos@below{below}
2371 \def\eql@markline@pos@bottom{bottom}
2372 \def\eql@markline@pos@baseline{baseline}
2373 \let\eql@markline@pos\eql@markline@pos@baseline
2374 \let\eql@markline@shift\z@
2375 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2376 \def\eql@markline@symbol{}

```

TODO: describe

```

2377 \def\eql@markline@select#1{%
2378 \let\eql@markline@shift\z@
2379 \eql@setkeys{markline}{#1}%
2380 \eql@markline@print
2381 }

```

TODO: describe

```

2382 \def\eql@markline@print{%
2383 \dimen@=\dimexpr\eql@markline@shift\relax
2384 \ifx\eql@markline@pos\eql@markline@pos@below
2385 \ifdim\dimen@=\z@\else
2386 \penalty\@M
2387 \vskip-\dimen@
2388 \fi
2389 \nointerlineskip
2390 \penalty\@M
2391 \vbox{\hfill\hbox{\eql@markline@symbol}}}%
2392 \else
2393 \ifx\eql@markline@pos\eql@markline@pos@baseline
2394 \advance\dimen@\prevdepth
2395 \fi
2396 \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}}%
2397 \dimen@\prevdepth
2398 \ht\z@\z@
2399 \dp\z@\z@
2400 \nointerlineskip

```

```

2401 \penalty\@M
2402 \vbox{\hfill\box\z@}%
2403 \prevdepth\dimen@
2404 \fi
2405 }

```

TODO: describe

```

2406 \def\eq@markline@inject#1{%
2407 \let\eq@markline@push\eq@false
2408 \ifx\eq@markline@pos\eq@markline@pos@below\else
2409 \ifdefined\eq@tagsleft\else
2410 \ifx\eq@equations@main\eq@multi@main
2411 \ifdefined\eq@numbering@multi
2412 \if@eqnsw
2413 \let\eq@markline@push\eq@true
2414 \fi
2415 \else
2416 \ifnum\eq@row@=\eq@tagpos@row@
2417 \let\eq@markline@push\eq@true
2418 \fi
2419 \fi
2420 \else
2421 \if@eqnsw
2422 \let\eq@markline@push\eq@true
2423 \fi
2424 \fi
2425 \fi
2426 \fi
2427 \ifdefined\eq@markline@push
2428 \global\eq@append\eq@interline@container{%
2429 \eq@append\eq@display@injectbefore{\eq@markline@select{push,#1}}}%
2430 \else
2431 \global\eq@append\eq@interline@container{%
2432 \eq@append\eq@display@injectbefore{\eq@markline@select{#1}}}%
2433 \fi
2434 }

```

TODO: describe

```

2435 \def\eq@markline@amsthm@opt[#1]{\eq@markline@inject{qed,#1}}
2436 \def\eq@markline@amsthm@staropt[#1]{\eq@markline@inject{qed,push,#1}}
2437 \def\eq@markline@amsthm@qed{\eq@teststaropt@tight
2438 \eq@markline@amsthm@staropt\eq@markline@amsthm@opt{}}
2439 \def\eq@markline@amsthm@register#1{\eq@letcs{#1@qed}\eq@markline@amsthm@qed}
2440 \def\eq@markline@amsthm@move#1#2{%
2441 \AddToHook{package/amsthm/after}{%
2442 \eq@letcs{#1@qed\expandafter}\csname#2@qed\endcsname}}

```

9 Column Placement

TODO: describe

9.1 Supporting Definitions

$\eq@shape@pos@$ (*dimen*) The registers $\eq@shape@pos@$ and $\eq@shape@amount@$ specify the currently selected horizontal alignment (0 for left, 1 for center, 2 for right) and the indentation amount,

respectively:

```
2443 \newcount\eq@shape@pos@
2444 \newdimen\eq@shape@amount@
2445 \let\eq@shape@lastrow\eq@false
```

`\eq@marginleft@` (*dimen*) The registers `\eq@marginleft@` and `\eq@marginright@` store the intended left and right margin for the equation lines: **TODO:** update

`\eq@marginright@` (*dimen*)
`\eq@centeroffset@` (*dimen*)

```
2446 \newdimen\eq@marginleft@
2447 \newdimen\eq@marginright@
2448 \newdimen\eq@marginleft@min@
2449 \newdimen\eq@centeroffset@
```

9.2 Shape Schemes

The horizontal alignment of each line is specified by a shape scheme.

`\eq@shape@tab@...` We select the scheme through a `\csname` selector with the following names:

```
2450 \def\eq@shape@tab@default{default}
2451 \def\eq@shape@tab@left{left}
2452 \def\eq@shape@tab@center{center}
2453 \def\eq@shape@tab@right{right}
2454 \def\eq@shape@tab@first{first}
2455 \def\eq@shape@tab@hanging{hanging}
2456 \def\eq@shape@tab@steps{steps}
```

For convenience, we add further alias names for the schemes:

```
2457 \let\eq@shape@tab@def\eq@shape@tab@default
2458 \let\eq@shape@tab@\eq@shape@tab@default
2459 \let\eq@shape@tab@l\eq@shape@tab@left
2460 \let\eq@shape@tab@c\eq@shape@tab@center
2461 \let\eq@shape@tab@r\eq@shape@tab@right
2462 \let\eq@shape@tab@rc\eq@shape@tab@first
2463 \let\eq@shape@tab@indent\eq@shape@tab@first
2464 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2465 \let\eq@shape@tab@lcr\eq@shape@tab@hanging
2466 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2467 \let\eq@shape@tab@lcr\eq@shape@tab@steps
```

`\eq@shape@mode` The currently selected scheme is stored in `\eq@shape@mode`. It is set to `default`:

```
2468 \let\eq@shape@mode\eq@shape@tab@default
```

`\eq@shape@set` Set the scheme via the translation table:

```
2469 \def\eq@shape@set#1{%
2470   \ifcsname eq@shape@tab@#1\endcsname
2471     \expandafter\let\expandafter\eq@shape@mode
2472       \csname eq@shape@tab@#1\endcsname
2473   \else
2474     \eq@error{shape '#1' unknown: setting to default}%
2475     \let\eq@shape@mode\eq@shape@tab@default
2476   \fi
2477 }
```

ape@layoutcenter@... Define the uniform shape schemes `left`, `center`, `right` and `default` for the central and
 shape@layoutleft@... left alignment layout. The scheme functions determine the desired alignment and
 indentation for the current row:

```

2478 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
2479 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2480 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2481 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2482 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
2483 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2484 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2485 \let\eq@shape@layoutleft@default\eq@shape@layoutleft@left

```

The **first** scheme implements left alignment with indentation for the first line (unless there is only one line):

```

2486 \def\eq@shape@layoutcenter@first{%
2487   \eq@shape@pos@z@
2488   \eq@shape@amount@z@
2489   \ifnum\eq@totalrows@>\@ne
2490     \ifnum\eq@row@=\@ne
2491       \eq@shape@amount@\eq@indent@
2492     \fi
2493   \fi
2494 }
2495 \def\eq@shape@layoutleft@first{%
2496   \eq@shape@pos@z@
2497   \eq@shape@amount@z@
2498   \ifnum\eq@totalrows@>\@ne
2499     \ifnum\eq@row@=\@ne
2500       \eq@shape@amount@\eq@indent@
2501     \fi
2502   \fi
2503 }

```

The **hanging** scheme implements left alignment with hanging indentation for the first line (unless there is only one line). In central alignment layout all but the first line are indented while in left aligned layout the first line has negative indentation:

```

2504 \def\eq@shape@layoutcenter@hanging{%
2505   \eq@shape@pos@z@
2506   \eq@shape@amount@\eq@indent@
2507   \ifnum\eq@totalrows@>\@ne
2508     \ifnum\eq@row@=\@ne
2509       \eq@shape@amount@z@
2510     \fi
2511   \fi
2512 }
2513 \def\eq@shape@layoutleft@hanging{%
2514   \eq@shape@pos@z@
2515   \eq@shape@amount@z@
2516   \ifnum\eq@totalrows@>\@ne
2517     \ifnum\eq@row@=\@ne
2518       \eq@shape@amount@-\eq@indent@
2519     \fi
2520   \fi
2521 }

```

The **steps** scheme implements singles out the first and last lines which are shifted left and right, respectively. In central alignment layout the shift operates on the alignment

whereas in left alignment layout the shift uses indentation:

```

2522 \def\eql@shape@layoutcenter@steps{%
2523   \eql@shape@amount@z@
2524   \eql@shape@pos@\@ne
2525   \ifnum\eql@totalrows@>\@ne
2526     \ifnum\eql@row@=\@ne
2527       \eql@shape@pos@z@
2528     \fi
2529     \ifnum\eql@row@=\eql@totalrows@
2530       \eql@shape@pos@tw@
2531     \fi
2532   \fi
2533 }
2534 \def\eql@shape@layoutleft@steps{%
2535   \eql@shape@pos@z@
2536   \eql@shape@amount@z@
2537   \ifnum\eql@totalrows@>\@ne
2538     \ifnum\eql@row@=\@ne
2539       \eql@shape@amount@-\eql@indent@
2540     \fi
2541     \ifnum\eql@row@=\eql@totalrows@
2542       \eql@shape@amount@\eql@indent@
2543     \fi
2544   \fi
2545 }
```

`\eql@shape@select` Select the shape selector function for the current scheme `@\eql@shape@mode` and layout
`\eql@shape@eval` and store it in `\eql@shape@eval`:

```

2546 \let\eql@shape@eval\undefined
2547 \def\eql@shape@select{%
2548   \expandafter\let\expandafter\eql@shape@eval
2549     \csname eql@shape%
2550     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
2551     @\eql@shape@mode\endcsname
2552 }
```

`\eql@shape@alignleft` Adjust the alignment of the current equation line. The optional argument specifies the
`\eql@shape@alignright` amount of indentation:
`\eql@shape@aligncenter`

```

2553 \protected\def\eql@shape@alignleft{%
2554   \global\eql@append\eql@cell@container{\eql@shape@pos@z@}%
2555   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2556 \protected\def\eql@shape@aligncenter{%
2557   \global\eql@append\eql@cell@container{\eql@shape@pos@\@ne}%
2558   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2559 \protected\def\eql@shape@alignright{%
2560   \global\eql@append\eql@cell@container{\eql@shape@pos@tw@}%
2561   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2562 \def\eql@shape@align@testpar#1{%
2563   \eql@ifstar@tight{#1[\eql@indent@]}%
2564   {\eql@ifnextgobble@tight{!}{#1[-\eql@indent@]}%
2565   {\eql@testopt@tight{#1}\z@}}%
2566 \def\eql@shape@alignamount@opt[#1]{\eql@shape@alignamount@set{#1}}
```

`\eql@shape@alignamount` **TODO:** describe

```

2567 \protected\def\eql@shape@alignamount{%
```

```

2568 \eql@ampprotecttwo\eql@ifstar@tight
2569 \eql@shape@alignamount@set\eql@shape@alignamount@add}
2570 \def\eql@shape@alignamount@add#1{%
2571 \global\eql@appendexpand\eql@cell@container{%
2572 \advance\eql@shape@amount@the\glueexpr#1\relax\relax}}
2573 \def\eql@shape@alignamount@set#1{%
2574 \global\eql@appendexpand\eql@cell@container{%
2575 \eql@shape@amount@the\glueexpr#1\relax\relax}}
2576 \def\eql@shape@align@enable{%
2577 \let\shoveleft\eql@shape@alignleft
2578 \let\shovecenter\eql@shape@aligncenter
2579 \let\shoveright\eql@shape@alignright
2580 \let\shoveby\eql@shape@alignamount
2581 }

```

TODO: describe

```

2582 \protected\def\eql@shape@align@default{%
2583 \eql@warn@here{\shove...}%
2584 \eql@ampprotect\eql@shape@align@testpar\eql@gobbleopt}
2585 \protected\def\eql@shape@alignamount@default{%
2586 \eql@warn@here{\shove...}%
2587 \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble}
2588 \def\eql@shape@align@disable{%
2589 \let\shoveleft\eql@shape@align@default
2590 \let\shovecenter\eql@shape@align@default
2591 \let\shoveright\eql@shape@align@default
2592 \let\shoveby\eql@shape@alignamount@default
2593 }

```

9.3 Width Data

width@block@ (*dimen*)

```

2594 \newdimen\eql@tagwidth@block@
2595 \newdimen\eql@tagheight@block@
2596 \newdimen\eql@tagdepth@block@

```

$\text{eql@dimensions@tab}$ **TODO:** new

```

2597 \let\eql@dimensions@tab\@empty

```

$\text{eql@dimensions@reset}$

```

2598 \def\eql@dimensions@reset{%
2599 \let\eql@dimensions@tab\@empty
2600 \eql@tagwidth@max@z@
2601 \eql@tagrows@z@
2602 }

```

$\text{eql@dimensions@add}$

```

2603 \def\eql@dimensions@add#1{%
2604 \global\eql@appendexpand\eql@dimensions@tab{#1}%
2605 }

```

$\text{eql@dimensions@addreg}$

```

2606 \def\eql@dimensions@addreg#1{#1\the#1\relax}

```

@dimensions@startrow

```
2607 \def\eqldimensions@startrow{%
2608   \eqldimensions@add{\eqldimensions@addreg\eqldrow@}%
2609 }
```

@dimensions@savecell

```
2610 \def\eqldimensions@savecell{%
2611   \eqldimensions@add{%
2612     \eqldimensions@addreg\eqldshape@pos@
2613     \eqldimensions@addreg\eqldcellwidth@
2614     \eqldimensions@addreg\eqldshape@amount@
2615     \noexpand\eqldimensions@cellcall
2616   }%
2617 }
```

@dimensions@savesep

```
2618 \def\eqldimensions@savesep{%
2619   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2620 }
```

@dimensions@endrow

```
2621 \def\eqldimensions@endrow{%
2622   \eqldimensions@add{,%
2623     \eqldimensions@addreg\eqldtagwidth@
2624     \eqldimensions@addreg\eqldline@height@
2625     \eqldimensions@addreg\eqldline@depth@
2626     \eqldimensions@addreg\eqldline@interline@
2627   ;}%
2628 }
```

@dimensions@saveblocktag

```
2629 \def\eqldimensions@saveblocktag{%
2630   \eqldimensions@add{\eqldrow@0\relax,%
2631     \eqldtagwidth@block@\the\eqldtagwidth@\relax
2632     \eqldtagheight@block@\the\ht\eqldtagbox@\relax
2633     \eqldtagdepth@block@\the\dp\eqldtagbox@\relax
2634     \eqldimensions@addreg\eqldtagpos@shift@
2635     \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
2636     \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
2637   ;}%
2638   \global\eqldtagwidth@max@\eqldtagwidth@
2639   \global\eqldtaggrows@\@ne
2640 }
```

@dimensions@savenoblocktag

```
2641 \def\eqldimensions@savenoblocktag{%
2642   \eqldimensions@add{\eqldrow@0\relax,;%
2643 }
```

@eqldimensions@for

```
2644 \def\eqldimensions@for#1{%
2645   \def\eqldimensions@forcall{#1}%
2646   \expandafter\eqldimensions@forstep\eqldimensions@tab
2647 }
```

l@dimensions@forstep

```

2648 \def\eql@dimensions@forstep\eql@row@#1\relax#2,##3;{%
2649   \eql@row@#1\relax
2650   \ifnum\eql@row@=\z@\else
2651     #3%
2652     \def\eql@dimensions@cells{##2}%
2653     \eql@dimensions@forall
2654     \expandafter\eql@dimensions@forstep
2655   \fi
2656 }
```

\eql@dimensions@get

```

2657 \def\eql@dimensions@get#1{%
2658   \eql@row@#1\relax
2659   \expandafter\eql@dimensions@getdef\expandafter{\the\eql@row@}%
2660   \expandafter\eql@dimensions@getparse\eql@dimensions@tab\@nil
2661 }
```

ql@dimensions@getdef

```

2662 \def\eql@dimensions@getdef#1{%
2663   \def\eql@dimensions@getparse
2664     ##1\eql@row@#1\relax##2,##3;##4\@nil{%
2665     ##3%
2666     \def\eql@dimensions@cells{##2}%
2667   }%
2668 }
```

\eql@colwidth@tab

```

2669 \let\eql@colwidth@tab\@empty
```

\eql@colwidth@get

```

2670 \def\eql@colwidth@get#1{%
2671   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2672 }
```

\eql@colwidth@save

```

2673 \def\eql@colwidth@save#1{%
2674   \edef\eql@colwidth@tab{%
2675     \noexpand\or\the#1%
2676     \unexpanded\expandafter{\eql@colwidth@tab}%
2677   }%
2678 }
```

\eql@dimensions@calc Compute the space that is available at the beginning and at the end of the row stored in \eql@dimensions@cells. The space available at the beginning is returned in \eql@line@avail@. and \eql@line@availsep@ describes the number of unused intercolumn separations. The total used width is returned in \eql@line@width@ and \eql@line@widthsep@ describes the number of used intercolumn separations. The available space at the end of the row is given as the difference to \eql@totalwidth@:

```

2679 \def\eql@dimensions@calc{%
2680   \eql@column@\z@
2681   \eql@line@pos@\z@
```

```

2682 \eq@line@possep@\z@
2683 \eq@line@avail@\eq@totalwidth@
2684 \eq@line@availsep@\eq@intercolumns@
2685 \eq@line@width@\z@
2686 \eq@line@widthsep@\z@
2687 \let\eq@dimensions@cellcall\eq@dimensions@calc@call
2688 \let\eq@dimensions@sepcall\eq@dimensions@calc@callsep
2689 \eq@dimensions@cells
2690 }

```

`ensions@calc@callsep` Callback for each intercolumn space.

```

2691 \def\eq@dimensions@calc@callsep{%
2692   \advance\eq@line@possep@\@ne
2693 }%

```

`dimensions@calc@call` Callback for each column. When a non-blank cell is encountered, the available space on the left will be fixed if it is still undetermined, and the total width is updated to the current position: **TODO**: implement an offset for central alignment (global?!)

```

2694 \def\eq@dimensions@calc@call{%
2695   \advance\eq@column@\@ne
2696   \ifnum\eq@totalcolumns@=\@ne
2697     \dimen@\eq@totalwidth@
2698   \else
2699     \dimen@\eq@colwidth@get\eq@column@\relax
2700   \fi
2701   \ifdim\eq@cellwidth@>\z@
2702     \ifdim\eq@line@width@=\z@
2703       \eq@line@avail@\eq@line@pos@
2704       \eq@line@availsep@\eq@line@possep@
2705       \ifcase\eq@shape@pos@
2706       \or
2707         \advance\eq@line@avail@\dimexpr
2708           (\dimen@-\eq@cellwidth@+\eq@centeroffset@)/\tw@\relax
2709       \or
2710         \advance\eq@line@avail@\dimexpr\dimen@-\eq@cellwidth@\relax
2711       \fi
2712       \advance\eq@line@avail@\eq@shape@amount@
2713     \fi
2714     \eq@line@width@\eq@line@pos@
2715     \eq@line@widthsep@\eq@line@possep@
2716     \ifcase\eq@shape@pos@
2717     \advance\eq@line@width@\eq@cellwidth@
2718     \or
2719     \advance\eq@line@width@\dimexpr
2720       (\dimen@+\eq@cellwidth@+\eq@centeroffset@)/\tw@\relax
2721     \or
2722     \advance\eq@line@width@\dimen@
2723     \fi
2724     \advance\eq@line@width@\eq@shape@amount@
2725   \fi
2726   \advance\eq@line@pos@\dimen@
2727 }

```

9.4 Best Line Selection

`@numbering@best@auto` **TODO:** describe

```
2728 \let\eql@numbering@best@auto\eql@false
```

`@best@row@` (*counter*)

`@best@space@` (*dimen*)

`@numbering@best@use` (*bool*)

```
2729 \newcount\eql@numbering@best@row@
```

```
2730 \newdimen\eql@numbering@best@space@
```

```
2731 \let\eql@numbering@best@use\eql@false
```

`@numbering@best@find` Determine the row with the largest available space on the side of the tags:

```
2732 \def\eql@numbering@best@find{%
2733   \eql@numbering@best@row@ \z@
2734   \eql@numbering@best@space@ \z@
2735   \eql@dimensions@for{%
2736     \eql@dimensions@calc
2737     \ifdefined\eql@tagsleft
2738       \dimen@ \eql@line@avail@
2739     \else
2740       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
2741     \fi
2742     \ifdim \dimen@ > \eql@numbering@best@space@
2743       \eql@numbering@best@row@ \eql@row@
2744       \eql@numbering@best@space@ \dimen@
2745     \fi
2746   }%
2747   \ifnum \eql@numbering@best@row@ > \z@
2748     \eql@tagpos@row@ \eql@numbering@best@row@
2749     \let\eql@tagpos@continuous\eql@false
2750     \eql@tagpos@prevrow@ \z@
2751   \fi
2752 }
```

`@numbering@best@test` **TODO:** describe

```
2753 \def\eql@numbering@best@test#1{%
2754   \eql@dimensions@get#1%
2755   \eql@dimensions@calc
2756   \ifdefined\eql@tagsleft
2757     \dimen@ \dimexpr \eql@line@avail@
2758       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
2759   \else
2760     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
2761       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
2762   \fi
2763   \ifdim \dimen@ < \eql@tagwidth@block@
2764     \let\eql@numbering@best@use\eql@true
2765   \fi
2766 }
```

`@numbering@best@eval` **TODO:** describe **TODO:** to test both lines individually may cause undesired effects

```
2767 \def\eql@numbering@best@eval{%
2768   \ifdefined\eql@numbering@best@auto
2769     \ifdefined\eql@numbering@best@use\else
2770       \ifdefined\eql@numbering@multi\else
```

```

2771         \ifnum\eql@tagpos@row@>\z@
2772             \eql@numbering@best@test\eql@tagpos@row@
2773         \fi
2774         \ifnum\eql@tagpos@prevrow@>\z@
2775             \eql@numbering@best@test\eql@tagpos@prevrow@
2776         \fi
2777     \fi
2778 \fi
2779 \fi
2780 \ifdefined\eql@numbering@best@use
2781     \eql@numbering@best@find
2782 \fi
2783 }

```

9.5 Tag Margin

TODO: describe **TODO:** if a tag margin is installed for a single line, it will shift the center even if there is no tag or importantly if a tag has been raised.

djust@calc@tagmargin

```

2784 \def\eql@adjust@calc@tagmargin{%
2785     \ifdefined\eql@tagmargin@val
2786         \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2787     \else
2788         \eql@tagmargin@\eql@tagwidth@max@
2789         \ifdim\eql@tagmargin@>\z@
2790             \advance\eql@tagmargin@-\eql@tagsepmin@
2791         \fi
2792     \fi

2793     \dimen@\eql@tagrows@\p@
2794     \ifnum\eql@totalrows@=\@ne
2795         \ifnum\eql@tagrows@=\@ne
2796             \advance\dimen@1sp\relax
2797         \fi
2798     \fi
2799     \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2800         \eql@tagmargin@\z@
2801     \fi

2802     \@tempdima\dimexpr\displaywidth
2803         -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2804     \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2805     \ifdim\@tempdimb>\z@
2806         \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
2807             \eql@tagmargin@\z@
2808         \fi
2809     \fi
2810 }

```

9.6 Single Column

ql@adjust@calc@lines

```

2811 \def\eql@adjust@calc@lines{%
2812     \eql@totalcolumns@\@ne

```

```

2813 \eq@intercolumns@z@
2814 \eq@colsep@z@
2815 \ifdefined\eq@layoutleft
2816 \eq@marginleft@glueexpr\eq@layoutleftmargin\relax
2817 \eq@marginleft@min@glueexpr\eq@layoutleftmarginmin\relax
2818 \ifdim\eq@marginleft@<\eq@marginleft@min@
2819 \eq@marginleft@\eq@marginleft@min@
2820 \fi
2821 \dimen@glueexpr\eq@layoutleftmarginmax\relax
2822 \ifdim\eq@marginleft@>\dimen@
2823 \eq@marginleft@\dimen@
2824 \fi
2825 \eq@marginright@z@
2826 \eq@centeroffset@z@
2827 \else
2828 \eq@adjust@calc@tagmargin
2829 \ifdefined\eq@paddingleft@val
2830 \eq@marginleft@\dimexpr
2831 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
2832 -\glueexpr\eq@paddingleft@val\relax\relax
2833 \ifdim\eq@marginleft@<z@
2834 \eq@marginleft@z@
2835 \fi
2836 \else
2837 \eq@marginleft@z@
2838 \fi
2839 \ifdefined\eq@paddingright@val
2840 \eq@marginright@\dimexpr
2841 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
2842 -\glueexpr\eq@paddingright@val\relax\relax
2843 \ifdim\eq@marginright@<z@
2844 \eq@marginright@z@
2845 \fi
2846 \else
2847 \eq@marginright@z@
2848 \fi
2849 \ifdim\eq@tagmargin@>z@
2850 \ifdefined\eq@tagsleft
2851 \ifdim\eq@marginleft@<\eq@tagsepmin@
2852 \eq@marginleft@\eq@tagsepmin@
2853 \fi
2854 \advance\eq@marginleft@\eq@tagmargin@
2855 \advance\eq@centeroffset@\eq@tagmargin@
2856 \else
2857 \ifdim\eq@marginright@<\eq@tagsepmin@
2858 \eq@marginright@\eq@tagsepmin@
2859 \fi
2860 \advance\eq@marginright@\eq@tagmargin@
2861 \advance\eq@centeroffset@-\eq@tagmargin@
2862 \fi
2863 \fi
2864 \eq@marginleft@min@z@
2865 \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
2866 \ifdefined\eq@tagsleft+ \else -\fi \eq@tagmargin@ \relax
2867 \fi

2868 \eq@totalwidth@\dimexpr\displaywidth
2869 -\eq@marginleft@-\eq@marginright@ \relax
2870 }

```

9.7 Multiple Columns

The following code computes the horizontal placement of columns. It distributes the columns evenly according to the layout presets and then determines whether there is enough space to place an equation tag on each line. If not, the intercolumn spacing and the space at the opposite margin can be reduced.

`@adjust@calc@columns` Main method to adjust column placement and spacing:

```
2871 \def\eql@adjust@calc@columns{%
```

If there is just a single alignment structure, there will be no intercolumn space that might stretch to adjust the columns to the margins. We disable fulllength to avoid a division by zero. Also guard against no columns at all (empty body), just in case:

```
2872 \ifnum\eql@totalcolumns@<\thr@@
2873   \eql@totalcolumns@\tw@
2874   \let\eql@columns@fulllength\eql@false
2875 \fi
```

Determine the number of intercolumn spaces `\eql@intercolumns@`:

```
2876 \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2877 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

Determine the left or target margin width depending on the layout:

```
2878 \ifdefined\eql@layoutleft
2879   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2880   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2881   \ifdim\eql@marginleft@<\eql@marginleft@min@
2882     \eql@marginleft@\eql@marginleft@min@
2883   \fi
2884 \else
```

Get the desired tag margin, increase by minimum tag separation if columns are aligned to the margins. Cancel tag margin if too wide:

```
2885   \eql@adjust@calc@tagmargin
2886   \ifdefined\eql@columns@fulllength
2887     \ifdim\eql@tagmargin@>\z@
2888       \advance\eql@tagmargin@\eql@tagsepmin@
2889     \fi
2890   \fi
2891   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2892     -\eql@intercolumns@\eql@colsepmin@\relax
2893     \eql@tagmargin@\z@
2894   \fi
2895   \eql@marginleft@min@\z@
2896 \fi
```

Compute the intercolumn space `\eql@colsep@`:

```
2897 \ifnum\eql@intercolumns@>\z@
```

Distribute the available horizontal space evenly onto the intercolumn spaces and the margins. Unless the columns are aligned to the margins, there are two margins in central alignment layout but only the right margin in left alignment layout:

```

2898 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2899 \ifdefined\eql@layoutleft
2900 \advance\eql@colsep@-\eql@marginleft@
2901 \else
2902 \advance\eql@colsep@-\eql@tagmargin@
2903 \fi
2904 \count@\eql@intercolumns@
2905 \ifdefined\eql@columns@fulllength\else
2906 \ifdefined\eql@layoutleft
2907 \advance\count@\@ne
2908 \else
2909 \advance\count@\tw@
2910 \fi
2911 \fi
2912 \divide\eql@colsep@\count@

```

Ensure that the intercolumn separation is within the specified bounds. Disable the upper bound if columns are to be aligned to the margins:

```

2913 \ifdim\eql@colsep@<\eql@colsepmin@
2914 \eql@colsep@\eql@colsepmin@
2915 \else
2916 \ifdefined\eql@columns@fulllength\else
2917 \dimen@\glueexpr\eql@colsepmax@val\relax
2918 \ifdim\eql@colsep@>\dimen@
2919 \eql@colsep@\dimen@
2920 \fi
2921 \fi
2922 \fi
2923 \else

```

For a single column, set the column separation to the minimum amount:

```

2924 \eql@colsep@\eql@colsepmin@
2925 \fi

```

Compute the left margin `\eql@marginleft@` depending on the layout:

```

2926 \ifdefined\eql@layoutleft

```

Set the default value:

```

2927 \ifdim\eql@colsep@=\eql@colsepmin@

```

If in left alignment layout the intercolumn space has been adjusted, compute the available space, determine left margin and make sure it is between the minimum and the default value:

```

2928 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
2929 -\eql@intercolumns@\eql@colsep@\relax
2930 \ifdim\dimen@<\eql@marginleft@
2931 \ifdim\dimen@<\eql@marginleft@min@
2932 \eql@marginleft@\eql@marginleft@min@
2933 \else
2934 \eql@marginleft@\dimen@
2935 \fi
2936 \fi
2937 \fi
2938 \else

```

In central alignment mode with column aligned to the margins, set margin to zero:

```

2939 \ifdefined\eql@columns@fulllength
2940 \eql@marginleft@<\z@

```

In central alignment mode with margins, distribute the available space equally to both margins, or remove the left margin if insufficient:

```

2941 \else
2942 \eql@marginleft@<\dimexpr(\displaywidth-\eql@totalwidth@
2943 -\eql@intercolumns@<\eql@colsep@-\eql@tagmargin@)/\tw@<\relax
2944 \ifdim\eql@marginleft@<<\z@
2945 \eql@marginleft@<\z@
2946 \fi
2947 \fi

```

Add tag margin in case of left tags:

```

2948 \ifdefined\eql@tagsleft
2949 \advance\eql@marginleft@<\eql@tagmargin@
2950 \fi
2951 \fi

```

Find the best row for tag placement:

```

2952 \eql@numbering@best@eval

```

Next consider all rows with tags and adjust the intercolumn and margin space to make the tags fit into the available space at the corresponding side as far as possible. First, select code depending on tag placement:

```

2953 \ifdefined\eql@tagsleft
2954 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
2955 \else
2956 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
2957 \fi

```

Loop over all rows or select the single row containing the tag. Fetch the width data for the current row. If a tag is present, compute the available space and try to adjust spaces if needed: **TODO:** complete for prevrow, ideally join treatment

```

2958 \ifdefined\eql@numbering@multi
2959 \eql@dimensions@for{%
2960 \ifdim\eql@tagwidth@>\z@
2961 \eql@dimensions@calc
2962 \eql@adjust@columns@test
2963 \fi
2964 }%
2965 \else
2966 \ifnum\eql@tagpos@row@>\z@
2967 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
2968 \eql@dimensions@get\eql@tagpos@row@
2969 \eql@tagwidth@\eql@tagwidth@block@
2970 \eql@dimensions@calc
2971 \eql@adjust@columns@test
2972 \fi
2973 \fi
2974 \ifnum\eql@tagpos@prevrow@>\z@
2975 \eql@dimensions@get\eql@tagpos@prevrow@
2976 \eql@tagwidth@\eql@tagwidth@block@
2977 \eql@dimensions@calc
2978 \eql@adjust@columns@test
2979 \fi

```

```
2980 \fi
```

From now on `\eq@totalwidth@` will include the left margin and the total intercolumn separation:

```
2981 \advance\eq@totalwidth@\dimexpr
2982 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
2983 }
```

Placement for Right Tags.

`\columns@test@tagsright` Test whether the spacing can be adjusted to make the current row fit:

```
2984 \def\eq@adjust@columns@test@tagsright{%
```

The register `\@tempdima` will hold the amount of available space. **TODO:** does this apply equally to left alignment layout?

```
2985 \@tempdima\dimexpr\displaywidth-\eq@line@width@-\eq@tagwidth@\relax
```

Test whether the space at the end of the row is sufficient to hold the tag with the current settings.

```
2986 \ifdim\@tempdima<\dimexpr
2987 \eq@marginleft@+\eq@line@widthsep@\eq@colsep@\relax
```

If not, determine whether the row and tag may at all fit into the available space with minimal intercolumn spaces and minimal left margin (in left alignment layout).

```
2988 \ifdim\@tempdima<\dimexpr
2989 \eq@marginleft@min@+\eq@line@widthsep@\eq@colsepmin@\relax\else
```

If so, hand over to `\eq@adjust@columns@modify@tagsright`.

```
2990 \eq@adjust@columns@modify@tagsright
2991 \fi
2992 \fi
2993 }
```

`\columns@modify@tagsright` Adjust the intercolumn space and left margin to make the row fit.

```
2994 \def\eq@adjust@columns@modify@tagsright{%
```

If there are any intercolumn spaces that contribute to the available space, determine how much intercolumn separation would be needed while keeping the current left margin fixed (in left alignment layout). In central alignment layout, assume that the left margin will be adjusted to match the intercolumn separation by stepping the number of columns to divide by.

```
2995 \ifnum\eq@line@widthsep@>\z@
2996 \dimen@\@tempdima
2997 \count@\eq@line@widthsep@
2998 \ifdefined\eq@layoutleft
2999 \advance\dimen@-\eq@marginleft@
3000 \else
3001 \ifdefined\eq@columns@fulllength\else
3002 \advance\count@\@ne
3003 \fi
3004 \fi
3005 \divide\dimen@\count@
```

If smaller, reduce the intercolumn separation, but make sure to not exceed the minimum allowed value.

```

3006 \ifdim\dimen@<\eq@colsep@
3007 \ifdim\dimen@<\eq@colsepmin@
3008 \eq@colsep@\eq@colsepmin@
3009 \else
3010 \eq@colsep@\dimen@
3011 \fi
3012 \fi
3013 \fi

```

Now adjust the left margin as much as needed to fit the contents.

```

3014 \dimen@\dimexpr\@tempdima-\eq@line@widthsep@\eq@colsep@\relax
3015 \ifdim\eq@marginleft@>\dimen@
3016 \eq@marginleft@\dimen@
3017 \fi
3018 }

```

Placement for Left Tags.

`columns@test@tagsleft` Test whether the spacing can be adjusted to make the current row fit:

```

3019 \def\eq@adjust@columns@test@tagsleft{%

```

The register `\@tempdima` will hold the deficit amount of space at the beginning of the row without adjustable space, and the register `\count@` will hold the number of intercolumn spaces that would contribute to space adjustments.

```

3020 \count@\numexpr\eq@intercolumns@-\eq@line@availsep@\relax
3021 \@tempdima\dimexpr\eq@tagwidth@-\eq@line@avail@\relax

```

Test whether the space at the beginning of the row is sufficient to hold the tag with the current settings.

```

3022 \ifdim\@tempdima>\dimexpr
3023 \eq@marginleft@+\eq@line@availsep@\eq@colsep@\relax

```

If not, first verify that the tag will fit the line (or the maximal left margin in left alignment layout).

```

3024 \ifdim\eq@tagwidth@<%
3025 \ifdefined\eq@layoutleft
3026 \glueexpr\eq@layoutleftmarginmax\relax
3027 \else
3028 \displaywidth
3029 \fi

```

If so, determine whether the row and tag may at all fit into the available space with minimal intercolumn spaces.

```

3030 \ifdim\@tempdima>\dimexpr
3031 \displaywidth-\eq@totalwidth@-\count@\eq@colsepmin@\relax\else

```

If so, hand over to `\eq@adjust@columns@modify@tagsleft`.

```

3032 \eq@adjust@columns@modify@tagsleft
3033 \fi
3034 \fi
3035 \fi
3036 }

```

umns@modify@tagsleft Adjust the intercolumn space and left margin to make the row fit.

```
3037 \def\eql@adjust@columns@modify@tagsleft{%
```

If there are any intercolumn spaces that contribute to the available space, determine how much intercolumn separation would be needed while keeping the current right margin fixed. In central alignment layout, assume that the right margin will be adjusted to match the intercolumn separation by stepping the number of columns to divide by.

```
3038 \ifnum\count@>\z@
3039 \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
3040 \ifdefined\eql@columns@fulllength\else
3041 \advance\count@\@ne
3042 \fi
3043 \divide\dimen@\count@
```

If smaller, reduce the intercolumn separation, but make sure to not exceed the minimum allowed value. Also adjust the left margin to keep the right margin fixed.

```
3044 \ifdim\dimen@<\eql@colsep@
3045 \ifdim\dimen@<\eql@colsepmin@
3046 \dimen@\eql@colsepmin@
3047 \fi
3048 \advance\dimen@-\eql@colsep@
3049 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3050 \advance\eql@colsep@\dimen@
3051 \fi
3052 \fi
```

Now adjust the left margin as much as needed to fit the contents.

```
3053 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3054 \ifdim\eql@marginleft@<\dimen@
3055 \eql@marginleft@\dimen@
3056 \fi
3057 }
```

10 Single Column Arrangement

The following code adjusts individual lines of equations for the equation and lines mode according to the selected layout and shape.

10.1 Supporting Definitions

\inf@bad The \inf@bad constant is for testing overfull boxes:

```
3058 \ifdefined\inf@bad\else%
3059 \newcount\inf@bad
3060 \inf@bad1000000\relax
3061 \fi
```

\eql@restore@hfuzz We need to change the value of \hfuzz temporarily. The method \eql@save@hfuzz stores the value for recovery through \eql@restore@hfuzz:

```
3062 \let\eql@restore@hfuzz\@empty
3063 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```

`\eqalignbadness@` The registers `\eqalignbadness@` and `\eqtagbadness@` store the allowable badness threshold for shrinking equation lines to the intended margin or to fit into the line at all before the tag is raised or lowered:

```
3064 \newcount\eqalignbadness@
3065 \newcount\eqtagbadness@
3066 \newcount\eqarrange@badness@
3067 \eqalignbadness@\inf@bad
3068 \eqtagbadness@\inf@bad
```

10.2 Arrangement Methods

`\eqarrange@try` Try to fit the current equation line in the available space. Argument #1 specifies the amount of reserved space. Unpack the box `\eqcellbox@`, replace the previous kerning with the new reserved space, and save the box back into `\eqcellbox@`:

```
3069 \def\eqarrange@try#1{%
3070   \ifdim#1>\dimexpr\displaywidth-\eqcellwidth@\relax
3071     \setbox\eqcellbox@\hbox to\displaywidth{%
3072       \unhbox\eqcellbox@\unkern\kern#1}%
3073     \eqarrange@badness@badness
3074   \else
3075     \eqarrange@badness@m@ne
3076   \fi
3077 }
```

`\eqarrange@print` We have found the final adjustment of the current line, so we typeset it with initial and final space adjustments #1 and #2, respectively. Restore the original value for `\hfuzz`:
TODO: adjust

```
3078 \def\eqarrange@print#1#2{%
3079   \eqrestore@hfuzz
3080   \if@eqnsw
3081     \ifdefined\eqtagsleft
3082       \eqtagbox@print@tagsleft
3083     \fi
3084   \fi
3085   \hbox to\displaywidth{%
3086     #1%
3087     \unhbox\eqcellbox@\unkern
3088     #2%
3089     \eqtagging@mathaddlast
3090   }%
3091   \if@eqnsw
3092     \ifdefined\eqtagsleft\else
3093       \eqtagbox@print@tagsright
3094     \fi
3095   \fi
3096 }
```

`\eqarrange@print@alignleft` Fit the current equation line with the selected alignment within a given left and right margins #1 and #2. If we're on the first line, adjust `\eqdisplay@firstavail@` to the minimum left available space we can guarantee:

```
3097 \def\eqarrange@print@alignleft#1#2{%
3098   \eqdisplay@firstavail@set{\dimexpr#1\relax}%
3099   \eqarrange@print{\kern#1}{\kern#2}%
3100 }
```

```

3101 \def\eql@arrange@print@alignright#1#2{%
3102   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3103   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3104 }

3105 \def\eql@arrange@print@aligncenter#1{%
3106   \eql@display@firstavail@set{\dimexpr
3107     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3108   \ifdim#1>\z@
3109     \eql@arrange@print{\kern#1\hfil}{}%
3110   \else
3111     \eql@arrange@print{\hfil}{\kern-#1}%
3112   \fi
3113 }

```

`\eql@arrange@init` Initialise the horizontal adjustment framework. Turn off overfull box messages temporarily – otherwise there would be unwanted extra ones emitted during our measuring operations. Select the shape scheme:

```

3114 \def\eql@arrange@init{%
3115   \eql@save@hfuzz
3116   \hfuzz\maxdimen
3117   \eql@shape@select
3118 }

```

`\eql@arrange@print@line` Select the appropriate adjustment method depending on the current alignment position, the selected tag placement if any: **TODO:** adjust

```

3119 \def\eql@arrange@print@line{%
3120   \eql@tagging@tagaddbox
3121   \csname eql@arrange%
3122     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3123     @init\endcsname
3124   \csname eql@arrange%
3125     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3126     @\ifdefined\eql@tagpos@reserve
3127       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3128       notag\fi\endcsname
3129 }

```

10.3 Central Alignment

TODO: describe

```

3130 \def\eql@arrange@aligncenter@init{%
3131   \eql@tagging@aligncenter
3132   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3133     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3134 }

```

TODO: describe

```

3135 \def\eql@arrange@aligncenter@notag{%
3136   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3137     \ifdim\eql@line@offset@<\eql@marginleft@min@
3138       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3139     \else
3140       \eql@line@offset@
3141     \fi

```

```

3142 \eql@arrange@print@aligncenter\eql@line@offset@
3143 \else
3144 \ifdim\eql@line@offset@<\eql@marginleft@min@
3145 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3146 \else
3147 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3148 \fi
3149 \fi
3150 }

```

TODO: describe

```

3151 \def\eql@arrange@aligncenter@tagsright{%
3152 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3153 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3154 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3155 \else
3156 \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3157 \fi
3158 \eql@arrange@print@aligncenter\eql@line@offset@
3159 \else
3160 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3161 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3162 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3163 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3164 \else
3165 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3166 \fi
3167 \else
3168 \let\eql@tagpos@reserve\eql@false
3169 \eql@arrange@aligncenter@notag
3170 \fi
3171 \fi
3172 }

```

```

3173 \def\eql@arrange@aligncenter@tagsleft{%
3174 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3175 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3176 \ifdim\eql@line@offset@<\eql@tagwidth@
3177 \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@\relax
3178 \else
3179 \eql@line@offset@
3180 \fi
3181 \eql@arrange@print@aligncenter\eql@line@offset@
3182 \else
3183 \eql@arrange@try\eql@tagwidth@
3184 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3185 \ifdim\eql@line@offset@<\eql@tagwidth@
3186 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3187 \else
3188 \eql@arrange@print@alignright\eql@tagwidth@\z@
3189 \fi
3190 \else
3191 \let\eql@tagpos@reserve\eql@false
3192 \eql@arrange@aligncenter@notag
3193 \fi
3194 \fi
3195 \else
3196 \eql@arrange@aligncenter@notag

```

```

3197 \fi
3198 }

```

10.4 Left Alignment

```

3199 \def\eql@arrange@alignleft@init{%
3200   \eql@tagging@alignleft
3201   \eql@line@offset@dimexpr\eql@marginleft@+\eql@shape@amount@relax
3202   \ifdim\eql@line@offset@<\eql@marginleft@min@
3203     \eql@line@offset@\eql@marginleft@min@
3204   \fi
3205 }

3206 \def\eql@arrange@alignleft@notag{%
3207   \ifdim\eql@line@offset@>\eql@marginleft@min@
3208     \eql@arrange@try\eql@line@offset@
3209     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3210       \eql@arrange@print@alignleft\eql@line@offset@\z@
3211     \else
3212       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3213     \fi
3214   \else
3215     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3216   \fi
3217 }

3218 \def\eql@arrange@alignleft@tagsright{%
3219   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@relax}%
3220   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3221     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3222   \else
3223     \ifdim\eql@line@offset@>\eql@marginleft@min@
3224       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@relax}%
3225     \fi
3226     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3227       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3228     \else
3229       \let\eql@tagpos@reserve\eql@false
3230       \eql@arrange@alignleft@notag
3231     \fi
3232   \fi
3233 }

3234 \def\eql@arrange@alignleft@tagsleft{%
3235   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3236     \ifdim\eql@line@offset@>\eql@tagwidth@
3237       \eql@arrange@try\eql@line@offset@
3238       \ifnum\eql@arrange@badness@<\eql@alignbadness@
3239         \eql@arrange@print@alignleft\eql@line@offset@\z@
3240       \else
3241         \eql@arrange@try\eql@tagwidth@
3242         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3243           \eql@arrange@print@alignright\eql@tagwidth@\z@
3244         \else
3245           \let\eql@tagpos@reserve\eql@false
3246           \eql@arrange@print@alignright\eql@marginleft@min@\z@
3247         \fi
3248       \fi
3249   \else
3250     \eql@arrange@try\eql@tagwidth@

```

```

3251     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3252     \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3253     \else
3254     \let\eqL@tagpos@reserve\eqL@false
3255     \eqL@arrange@alignleft@notag
3256     \fi
3257   \fi
3258 \else
3259   \eqL@arrange@alignleft@notag
3260 \fi
3261 }

```

10.5 Right Alignment

```

3262 \def\eqL@arrange@alignright@init{%
3263   \eqL@tagging@alignright
3264   \eqL@line@offset@dimexpr\eqL@marginright@-\eqL@shape@amount@relax
3265   \ifdim\eqL@line@offset@<z@
3266     \eqL@line@offset@z@
3267   \fi
3268 }

```

TODO: describe

```

3269 \def\eqL@arrange@alignright@notag{%
3270   \ifdim\eqL@line@offset@>z@
3271     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3272     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3273       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3274     \else
3275       \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3276     \fi
3277   \else
3278     \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3279   \fi
3280 }

```

TODO: describe

```

3281 \def\eqL@arrange@alignright@tagsright{%
3282   \ifdim\eqL@line@offset@>\eqL@tagwidth@
3283     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3284     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3285       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3286     \else
3287       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3288       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3289         \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3290       \else
3291         \let\eqL@tagpos@reserve\eqL@false
3292         \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3293       \fi
3294     \fi
3295   \else
3296     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3297     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3298       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3299     \else
3300       \let\eqL@tagpos@reserve\eqL@false
3301       \eqL@arrange@alignright@notag
3302     \fi

```

```
3303 \fi
3304 }
```

TODO: describe

```
3305 \def\eql@arrange@alignright@tagsleft{%
3306   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3307     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3308     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3309       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3310     \else
3311       \ifdim\eql@line@offset@>\z@
3312         \eql@arrange@try\eql@tagwidth@
3313       \fi
3314       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3315         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3316       \else
3317         \let\eql@tagpos@reserve\eql@false
3318         \eql@arrange@alignright@notag
3319       \fi
3320     \fi
3321   \else
3322     \eql@arrange@alignright@notag
3323   \fi
3324 }
```

11 Equations Box Environment

TODO: outline sequence of calls

TODO: describe

TODO: fixed width version (works only towards intercolumn stretch)?

TODO: vspace?!

11.1 Line Breaks

`\eql@box@cr`

```
3325 \protected\def\eql@box@cr{\eql@srbgroup%
3326   \eql@ifnextgobble@tight~%
3327   {\let\eql@punct@line\@empty\eql@box@cr@opt}%
3328   \eql@box@cr@opt}
3329 \def\eql@box@cr@opt{\eql@ifnextchar@tight[\eql@box@cr@skip\eql@box@cr@plain}
3330 \def\eql@box@cr@plain{\eql@sregroup\eql@box@cr@}
3331 \def\eql@box@cr@skip[#1]{\eql@sregroup
3332   \eql@box@cr@
3333   \noalign{%
3334     \vskip\glueexpr#1\relax
3335   }}%
3336 }
3337 \def\eql@box@cr@{%
3338   \eql@punct@apply@line
3339   \eql@hook@lineout
3340   \eql@box@lastcell
3341   \cr
3342 }
```

11.2 Stacked Mode

```
3343 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}
```

TODO: templates

```
3344 \def\eql@box@open@stacked{%
3345   \eql@shape@align@enable
3346   \let\eql@box@lastcell\eql@box@lastcell@stacked
3347   \everycr{\noalign{%
3348     (dev)\eql@dev{starting line \the\eql@row@}%
3349     \global\advance\eql@row@\@ne
3350   }}%
3351   \tabskip\z@skip
3352   \halign\bgroup
3353     &%
3354     \global\let\eql@cell@container\@empty
3355     \setbox\eql@cellbox@\hbox{%
3356       \eql@strut@cell
3357       \@lign
3358       $\m@th\displaystyle
3359       \eql@hook@colin
3360       ##%
3361       \eql@punct@apply@col
3362       \eql@hook@colout
3363       \eql@tagging@mathsave
3364       $%
3365       \eql@tagging@mathaddlast
3366     }%
3367     \ifdefined\eql@shape@lastrow
3368       \eql@totalrows@\eql@row@
3369     \fi
3370     \eql@shape@eval
3371     \eql@cell@container
3372     \ifdefined\eql@frame@cmd
3373       \ifcase\eql@shape@pos@
3374         \eql@frame@measure
3375         \advance\eql@shape@amount@-\eql@frame@margin@
3376       \or\or
3377         \eql@frame@measure
3378         \advance\eql@shape@amount@+\eql@frame@margin@
3379       \fi
3380     \eql@frame@print
3381   \fi
3382   \ifcase\eql@shape@pos@
3383     \kern\eql@shape@amount@
3384     \box\eql@cellbox@
3385     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3386     -\eql@shape@amount@+\@flushglue\relax
3387     \eql@tagging@alignleft
3388   \or
3389     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3390     \box\eql@cellbox@
3391     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3392     \eql@tagging@aligncenter
3393   \or
3394     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3395     +\eql@shape@amount@+\@flushglue\relax
3396     \box\eql@cellbox@
3397     \kern-\eql@shape@amount@
```

```

3398     \eql@tagging@alignright
3399     \fi
3400     \tabskip\eql@colsep@\relax
3401     \crrc
3402     \noalign{%
3403         \global\let\eql@shape@lastrow\eql@false
3404         \eql@hook@blockbefore
3405     }%
3406     \eql@hook@blockin
3407 }
3408 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

11.3 Aligned Mode

```

3409 \def\eql@box@lastcell@odd{%
3410     &\omit
3411     \eql@prevwidth@\wd\eql@cellbox@
3412     \let\eql@frame@cmd\eql@frame@prevcmd
3413     \ifdefined\eql@frame@cmd
3414         \eql@frame@measure
3415         \advance\eql@prevwidth@\eql@frame@margin@
3416         \eql@frame@print
3417     \fi
3418     \kern-\eql@prevwidth@
3419     \unhbox\eql@cellbox@
3420     \hfil
3421     &\omit\kern-\eql@colsep@
3422 }%
3423 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}
3424 \def\eql@box@open@aligned{%
3425 % \TODO templates
3426 \eql@shape@align@disable
3427 \let\eql@box@lastcell\@empty
3428 \everycr{\noalign{%
3429 (dev)\eql@dev{starting new line}%
3430 }}%
3431 \tabskip\z@skip
3432 \halign\bgroup
3433     &%
3434     \let\eql@box@lastcell\eql@box@lastcell@odd
3435     \global\let\eql@cell@container\@empty
3436     \global\setbox\eql@cellbox@\hbox{%
3437         \eql@strut@cell
3438         \@lign
3439         $\m@th\displaystyle
3440         \eql@hook@colin
3441         ##%
3442         \eql@class@innerleft
3443         \eql@hook@innerleft
3444         \eql@tagging@mathsave
3445         $%
3446         \eql@tagging@mathaddlast
3447     }%
3448     \eql@cell@container
3449     \hfil
3450     \kern\wd\eql@cellbox@
3451     \ifdefined\eql@frame@cmd
3452         \eql@frame@measure

```

```

3453     \kern\eq@frame@margin@
3454     \fi
3455     \global\let\eq@frame@prevcmd\eq@frame@cmd
3456     \tabskip\z@skip
3457     &%
3458     \eq@prevwidth@\wd\eq@cellbox@
3459     \let\eq@box@lastcell\eq@box@lastcell@even
3460     \let\eq@frame@cmd\eq@frame@prevcmd
3461     \global\let\eq@cell@container\@empty
3462     \setbox\eq@cellbox@\hbox{%
3463         \unhbox\eq@cellbox@
3464         \eq@strut@cell
3465         \@lign
3466         $\m@th\displaystyle
3467         \eq@hook@innerright
3468         \eq@class@innerright@sel
3469         ##%
3470         \eq@punct@apply@col
3471         \eq@hook@colout
3472         \eq@tagging@mathsave
3473         $%
3474         \eq@tagging@mathaddlast
3475     }%
3476     \eq@cell@container
3477     \ifdefined\eq@frame@cmd
3478         \eq@frame@measure
3479         \advance\eq@prevwidth@\eq@frame@margin@
3480         \eq@frame@print
3481     \fi
3482     \kern-\eq@prevwidth@
3483     \unhbox\eq@cellbox@
3484     \hfil
3485     \tabskip\eq@colsep@\relax
3486     \crrr
3487     \noalign{%
3488         \eq@hook@blockbefore
3489     }%
3490     \eq@hook@blockin
3491 }
3492 \def\eq@mode@aligned{\let\eq@box@open\eq@box@open@aligned}

```

11.4 Main

```

3493 \let\eq@box@box\vcenter
3494 \let\eq@box@open\@undefined
3495 \let\eq@box@frame\@firstofone
3496 \def\eq@box@wrap#1#2{\def\eq@box@frame##1{#1##1#2}}

```

TODO: can we avoid setting `\eq@totalrows@` globally here? **TODO:** this is needed for escaping the box and then set the alignment **TODO:** maybe determine alignment within inner math?! **TODO:** difficulty: last line being known (for steps) only after all cells have been processed. Note: only works for single column anyway! we do not have to cater for more!

```

3497 \def\eq@box@close{%
3498     \ifvmode\else
3499         \global\let\eq@shape@lastrow\eq@true
3500         \eq@punct@apply@block
3501         \eq@box@cr@

```

```

3502 \fi
3503 \noalign{%
3504 \eqLhook@blockafter
3505 \global\let\eqL@shape@lastrow\eqL@false
3506 }%
3507 \eqL@tagging@tablesaveinner
3508 \egroup
3509 }

```

\eqL@box@vcenter

```

3510 \def\eqL@box@vcenter#1{%
3511 \ifmmode
3512 \vcenter{#1}%
3513 \else
3514 $\m@th\vcenter{#1}$%
3515 \fi
3516 }

```

\eqL@box@start

```

3517 \let\eqL@box@endmath\eqL@false
3518 \def\eqL@box@start{%
3519 \relax
3520 \ifmmode
3521 \let\eqL@box@endmath\eqL@false
3522 \else
3523 \let\eqL@box@endmath\eqL@true
3524 \expandafter$%$
3525 \fi
3526 \eqL@box@processopt
3527 \eqL@stack@save@box
3528 \let\eqL@frame@cmd\@undefined
3529 \let\eqL@layoutleft\eqL@false
3530 \eqL@row@z@
3531 \eqL@totalrows@\@M
3532 \eqL@shape@select
3533 \setbox\z@\ifx\eqL@box@box\vcenter
3534 \expandafter\vbox
3535 \else
3536 \expandafter\eqL@box@box
3537 \fi\bgroup
3538 \eqL@display@nest
3539 \let\\eqL@box@cr
3540 \eqL@spread@set
3541 \eqL@strut@make
3542 \eqL@box@open
3543 }

```

\eqL@box@end

```

3544 \def\eqL@box@end{%
3545 \eqL@box@close
3546 \egroup
3547 \eqL@box@frame{%
3548 \ifdefined\eqL@display@marginleft
3549 \hskip\glueexpr\eqL@display@marginleft\relax
3550 \fi
3551 \ifx\eqL@box@box\vcenter

```

```

3552     \eql@box@vcenter{\unvbox\z@}%
3553   \else
3554     \box\z@
3555   \fi
3556   \eql@tagging@tableaddinner
3557   \ifdefined\eql@display@marginright
3558     \hskip\glueexpr\eql@display@marginright\relax
3559   \fi
3560 }%
3561 \eql@stack@restore
3562 \ifdefined\eql@box@endmath
3563   \expandafter$%$
3564 \fi
3565 }

```

11.5 Environment

`equationsbox` (*env.*)

```

3566 \newenvironment{equationsbox}{%
3567 (dev)\eql@dev@enterenv
3568   \eql@ampprotect\eql@box@testall\eql@box@start
3569 }{%
3570   \eql@box@end
3571 (dev)\eql@dev@leaveenv
3572 }

3573 \def\eql@box@testall{\eql@parseopt\eql@box@parseopt}
3574 \def\eql@box@parseopt{%
3575   \ifx\eql@parseopt@token[%]
3576     \let\eql@parseopt@next\eql@parseopt@opt
3577   \fi
3578   \ifx\eql@parseopt@token=%
3579     \let\eql@parseopt@next\eql@parseopt@lines
3580   \fi
3581   \ifx\eql@parseopt@token|
3582     \let\eql@parseopt@next\eql@parseopt@columns
3583   \fi
3584   \ifx\eql@parseopt@token'
3585     \let\eql@parseopt@next\eql@parseopt@punctall
3586   \fi
3587 }

```

`\eql@box@processopt` **TODO:** describe

```

3588 \def\eql@box@processopt{%
3589   \let\eql@box@frame\@firstofone
3590   \let\eql@display@marginleft\@undefined
3591   \let\eql@display@marginright\@undefined
3592   \eql@nextopt@process{equationsbox}%
3593   \let\eql@punct@block\eql@punct@main
3594   \let\eql@punct@main\relax
3595   \eql@colsep@\glueexpr\eql@box@colsep\relax
3596   \ifdefined\eql@paddingleft@val
3597     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3598   \else
3599     \eql@paddingleft@\z@
3600   \fi

```

```

3601 \ifdefined\eql@paddingright@val
3602   \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3603 \else
3604   \eql@paddingright@\z@
3605 \fi
3606 \eql@indent@\glueexpr\eql@indent@val\relax
3607 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

3608 \def\eql@single@start@native{%
3609   \eql@display@init
3610   \eql@display@print
3611   \let\raisetag\eql@raisetag@default
3612   \eql@shape@align@disable
3613   \eql@hook@eqin
3614   % \mathopen{}}%
3615 }%

```

TODO: describe

```

3616 \def\eql@single@end@native{%
3617   % \mathclose{}}%
3618   \eql@tags@container
3619   \eql@numbering@single@eval
3620   \if@eqnsw
3621     \ifdefined\eql@tagsleft
3622       \leqno
3623     \else
3624       \eqno
3625     \fi
3626     \eql@composetag@print
3627   \fi
3628   \eql@interline@container
3629   \advance\eql@belowspace@\eql@vspaceskip@
3630   \eql@display@container
3631   \eql@display@penalty
3632   \eql@display@vspace@native
3633 }%

```

12.2 Print

```

3634 \def\eql@single@start@print{%
3635   \eql@display@init
3636   \eql@display@print
3637   \eql@shape@align@enable
3638   \eql@totalrows@\@ne
3639   \eql@row@\@ne
3640   \eql@arrange@init
3641   \global\let\eql@cell@container\@empty
3642   \prevgraf\numexpr\prevgraf+\@ne\relax
3643   \setbox\eql@cellbox@\hbox\bgroup
3644     \eql@restore@hfuzz

```

```

3645 \eq@strut@cell
3646 $\m@th\displaystyle%$
3647 \eq@hook@eqin
3648 }

3649 \def\eq@single@end@print{%
3650 \eq@tagging@mathsave
3651 $%$
3652 \hfil
3653 \kern\z@
3654 \egroup
3655 \prevgraf\numexpr\prevgraf-\@ne\relax

3656 \eq@shape@eval
3657 \eq@cell@container

3658 \ifdefined\eq@frame@cmd
3659 \eq@frame@adjust
3660 \fi

3661 \eq@cellwidth@\wd\eq@cellbox@
3662 \eq@line@height@\ht\eq@cellbox@
3663 \eq@line@depth@\dp\eq@cellbox@
3664 \eq@totalwidth@\eq@cellwidth@
3665 \eq@totalheight@\dimexpr\eq@line@height@+\eq@line@depth@\relax
3666 \eq@topheight@\eq@line@height@
3667 \eq@bottomdepth@\eq@line@depth@

3668 \eq@tags@container
3669 \eq@numbering@single@eval
3670 \if@eqnsw
3671 \eq@tagbox@make\eq@composetag@print
3672 \eq@tagrows@\@ne
3673 \ifdefined\eq@tagpos@reserve\else
3674 \eq@tagwidth@\z@
3675 \fi
3676 \eq@tagheight@block@\ht\eq@tagbox@
3677 \eq@tagdepth@block@\dp\eq@tagbox@
3678 \else
3679 \eq@numbering@warnunused
3680 \eq@tagwidth@\z@
3681 \eq@tagrows@\z@
3682 \fi
3683 \eq@tagwidth@max@\eq@tagwidth@
3684 \eq@tagpos@single@eval
3685 \eq@tagpos@print@line@eval

3686 \eq@intercolumns@\z@
3687 \eq@adjust@calc@lines

3688 \eq@display@halign@init{}%
3689 \halign{##\crr
3690 \noalign{\eq@display@halign@start}%
3691 \eq@arrange@print@line
3692 \cr
3693 \noalign{\eq@display@halign@end}%
3694 \eq@tagging@tablesavelines
3695 }%
3696 \eq@tagpos@print@line@end
3697 \eq@display@close
3698 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```
3699 \def\eql@lines@measure@line@begin{%
3700 (dev)\eql@dev{starting line \the\eql@row}%
3701 \eql@numbering@measure@line@begin
3702 \eql@hook@linein
3703 }
```

TODO: describe

```
3704 \def\eql@lines@measure@line@end{%
3705 \eql@punct@apply@line
3706 \eql@hook@lineout
3707 }
```

TODO: describe **TODO:** it would be an option to add the absolute shove amount to the calculation of the maximum width

```
3708 \def\eql@lines@measure@cell{%
3709 \ifdefined\eql@frame@cmd
3710 \ifcase\eql@shape@pos@
3711 \eql@frame@measure
3712 \advance\eql@shape@amount@-\eql@frame@margin@
3713 \or\or
3714 \eql@frame@measure
3715 \advance\eql@shape@amount@+\eql@frame@margin@
3716 \fi
3717 \eql@frame@print
3718 \fi
3719 \eql@cellwidth@\wd\eql@cellbox@
3720 \eql@line@height@\ht\eql@cellbox@
3721 \eql@line@depth@\dp\eql@cellbox@
3722 \eql@dimensions@startrow
3723 \eql@dimensions@savecell
3724 \kern\eql@cellwidth@
3725 }
```

`\eql@lines@measure`

```
3726 \def\eql@lines@measure{%
3727 (dev)\eql@dev@center\eql@lines@measure
3728 \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3729 \eql@totalrows@\@M
3730 \eql@shape@select

3731 \setbox\z@\vbox{\measuring@true\halign{%
3732 \global\let\eql@cell@container\@empty
3733 \setbox\eql@cellbox@\hbox{%
3734 \eql@strut@cell
3735 \@lign
3736 $\m@th\displaystyle
3737 \eql@hook@colin
3738 ###}
```

```

3739         \eql@punct@apply@col
3740         \eql@hook@colout
3741     $%
3742 }%
3743 \ifdefined\eql@shape@lastrow
3744     \eql@totalrows@\eql@row@
3745 \fi
3746 \eql@shape@eval
3747 \eql@cell@container
3748 \eql@lines@measure@cell
3749 \eql@measure@tag
3750 \eql@measure@endrow
3751 \crrr

3752 \noalign{%
3753     \global\let\eql@shape@lastrow\eql@false
3754     \eql@hook@blockbefore
3755 }%
3756 \eql@hook@blockin
3757 \eql@scan@body
3758 \ifvmode\else
3759     \global\let\eql@shape@lastrow\eql@true
3760     \eql@punct@apply@block
3761     \eql@hook@blockout
3762     \eql@display@endline
3763 \cr
3764 \fi
3765 \omit
3766 \cr
3767 \noalign{%
3768     \eql@hook@blockafter
3769     \global\let\eql@shape@lastrow\eql@false
3770 }%
3771 }%

3772 \eql@measure@close

3773 \setbox\z@\vbox{%
3774     \unvbox\z@
3775     \unpenalty
3776     \global\setbox\@ne\lastbox
3777 }%
3778 \eql@totalwidth@\wd\@ne

3779 <dev>\eql@dev@leave\eql@lines@measure
3780 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3781 \def\eql@lines@adjust{%
3782     \eql@tagpos@adjust@eval
3783     \eql@adjust@calc@lines
3784     \eql@numbering@best@eval
3785 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```
3786 \def\eq@lines@print@line@begin{%
3787 <dev>\eq@dev{starting line \the\eq@row}%
3788 \eq@numbering@print@line@begin
3789 \eq@hook@linein
3790 }
```

TODO: describe

```
3791 \def\eq@lines@print@line@end{%
3792 \eq@punct@apply@line
3793 \eq@hook@lineout
3794 }
```

TODO: describe

```
3795 \def\eq@lines@print@line@adjust{%
3796 \ifdefined\eq@frame@cmd
3797 \ifcase\eq@shape@pos@
3798 \eq@frame@measure
3799 \advance\eq@shape@amount@-\eq@frame@margin@
3800 \or\or
3801 \eq@frame@measure
3802 \advance\eq@shape@amount@+\eq@frame@margin@
3803 \fi
3804 \eq@frame@adjust
3805 \fi
3806 \eq@cellwidth@\wd\eq@cellbox@
3807 \eq@line@height@\ht\eq@cellbox@
3808 \eq@line@depth@\dp\eq@cellbox@
3809 \eq@numbering@print@line@eval
3810 \if@eqnsw
3811 \eq@tagbox@make\eq@composetag@print
3812 \fi
3813 \eq@tagpos@print@line@eval
3814 \eq@arrange@print@line
3815 \eq@tagpos@print@line@end
3816 }
```

TODO: describe

```
3817 \def\eq@lines@print{%
3818 <dev>\eq@dev@enter\eq@lines@print
3819 \eq@arrange@init
3820 \eq@display@halign@init\eq@lines@print@line@begin
3821 \eq@display@halign@letcr\eq@lines@print@line@end
3822 \tabskip\z@skip

3823 \halign{%
3824 \global\let\eq@cell@container\@empty
3825 \setbox\eq@cellbox@\hbox{%
3826 \eq@restore@hfuzz
3827 \eq@strut@cell
3828 \@lign
3829 $\m@th\displaystyle
3830 \eq@hook@colin
```

```

3831      ##%
3832      \eql@punct@apply@col
3833      \eql@hook@colout
3834      \eql@tagging@mathsave
3835      $%
3836      \hfil
3837      \kern\z@
3838      }%
3839      \eql@shape@eval
3840      \eql@cell@container
3841      \eql@lines@print@line@adjust
3842      \crrr

3843      \noalign{%
3844      \eql@display@halign@start
3845      \eql@numbering@print@block@begin
3846      \eql@hook@blockbefore
3847      }%
3848      \eql@hook@blockin
3849      \eql@scan@body
3850      \ifvmode\else
3851      \relax
3852      \eql@punct@apply@block
3853      \eql@hook@blockout
3854      \eql@display@endline
3855      \cr
3856      \fi
3857      \noalign{%
3858      \eql@hook@blockafter
3859      \eql@display@halign@end
3860      (dev)\eql@dev@leave\eql@lines@print
3861      }%
3862      \eql@tagging@tablesavelines
3863      }%
3864      }

```

14 Multi-Line with Multiple Columns

TODO: describe **TODO:** outline sequence of calls

14.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
3865 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
3866 \def\eql@columns@completerow{%
3867   \count@=\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
3868   \edef\eql@tmp{%
3869     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
3870   \eql@tmp
3871   }

3872 \def\eql@columns@overfull{%
3873   \dimen@=\eql@line@width@

```

```

3874 \advance\dimen@-\hfuzz
3875 \ifdim\dimen@>\displaywidth
3876 \setbox\z@\hbox to\displaywidth{\hbox to\eq@line@width@{\hfil}}%
3877 \wd\z@\z@
3878 \ht\z@\eq@line@height@
3879 \dp\z@\eq@line@depth@
3880 \box\z@
3881 \fi
3882 }

```

14.2 Transpose

TODO: describe

TODO: describe

```

3883 \let\eq@transpose@active\eq@false
3884 \def\eq@transpose@end{\eq@transpose@end}
3885 \def\eq@transpose@skip{&\eqnpunct{}}
3886 \def\eq@transpose@complete{%
3887 \relax\ifodd\eq@column@\expandafter\eq@transpose@skip\fi&}

```

TODO: describe

```

3888 \def\eq@transpose{%
3889 \eq@totalcolumns@\z@
3890 \eq@totalrows@\z@
3891 \expandafter\eq@transpose@scan@col\the\eq@scan@reg@&\eq@transpose@end&
3892 \eq@scan@reg@{}%
3893 \eq@row@\z@
3894 \eq@transpose@output@row
3895 }

```

TODO: describe

```

3896 \def\eq@transpose@save@col#1{%
3897 \@namedef{eq@transpose@data@col@\the\eq@totalcolumns@}{%
3898 \ifcase\eq@row@#1\else\let\eq@tmp\eq@transpose@skip\fi}}

```

TODO: describe

```

3899 \def\eq@transpose@scan@col#1&{%
3900 \def\eq@tmpa{#1}%
3901 \ifx\eq@tmpa\eq@transpose@end\else
3902 \advance\eq@totalcolumns@\@ne
3903 \eq@row@\z@
3904 \let\eq@transpose@data@col\@empty
3905 \eq@transpose@scan@row#1\\eq@transpose@end\\%
3906 \ifnum\eq@row@>\eq@totalrows@
3907 \eq@totalrows@\eq@row@
3908 \fi
3909 \expandafter\eq@transpose@save@col\expandafter{\eq@transpose@data@col}%
3910 \expandafter\eq@transpose@scan@col
3911 \fi
3912 }

```

TODO: describe

```

3913 \def\eq@transpose@append@row#1{%
3914 \advance\eq@row@\@ne
3915 \eq@append\eq@transpose@data@col{\or\def\eq@tmp{#1}}}

```

TODO: describe

```
3916 \def\eq@transpose@scan@row#1\\{%
3917   \def\eq@tmpa{#1}%
3918   \ifx\eq@tmpa\eq@transpose@end\else
3919     \ifx\eq@transpose@active+
3920       \eq@transpose@scan@cell#1&\eq@transpose@end&%
3921     \else
3922       \eq@transpose@append@row{#1}%
3923     \fi
3924     \expandafter\eq@transpose@scan@row
3925   \fi
3926 }
```

TODO: describe

```
3927 \def\eq@transpose@scan@cell#1&#2{%
3928   \def\eq@tmpa{#2}%
3929   \ifx\eq@tmpa\eq@transpose@end
3930     \eq@transpose@append@row{#1}%
3931   \else
3932     \eq@transpose@append@row{#1&#2}%
3933     \expandafter\eq@transpose@scan@cell@next
3934   \fi
3935 }
```

TODO: describe

```
3936 \def\eq@transpose@scan@cell@next#1&{%
3937   \def\eq@tmpa{#1}%
3938   \ifx\eq@tmpa\eq@transpose@end\else
3939     \eq@transpose@append@row{&#1}%
3940     \expandafter\eq@transpose@scan@cell@next
3941   \fi
3942 }
```

TODO: describe

```
3943 \def\eq@transpose@output@row{%
3944   \ifnum\eq@row@<\eq@totalrows@
3945     \advance\eq@row@\@ne
3946     \eq@column@\z@
3947     \eq@transpose@output@col
3948     \ifnum\eq@row@<\eq@totalrows@
3949       \eq@scan@addto\\%
3950     \fi
3951     \expandafter\eq@transpose@output@row
3952   \fi
3953 }
```

TODO: describe

```
3954 \def\eq@transpose@output@col{%
3955   \ifnum\eq@column@<\eq@totalcolumns@
3956     \advance\eq@column@\@ne
3957     \csname eq@transpose@data@col@\the\eq@column@\endcsname
3958     \expandafter\eq@scan@addto\expandafter{\eq@tmp}%
3959     \ifnum\eq@column@<\eq@totalcolumns@
3960       \eq@scan@addto{\eq@transpose@complete}%
3961     \fi
3962     \expandafter\eq@transpose@output@col
```

```

3963 \fi
3964 }

```

14.3 Measure

TODO: describe **TODO:** this is called also for extra line and concluding cr

```

s@measure@line@begin

```

```

3965 \def\eql@columns@measure@line@begin{%
3966 (dev)\eql@dev{starting line \the\eql@row}%
3967 \global\eql@column@\z@
3968 \global\eql@line@height@\z@
3969 \global\eql@line@depth@\z@
3970 \eql@numbering@measure@line@begin
3971 \eql@hook@linein
3972 }

```

```

3973 \def\eql@columns@measure@cell{%
3974 \eql@cellwidth@\wd\eql@cellbox@
3975 \ifdefined\eql@frame@cmd
3976 \eql@frame@measure
3977 \advance\eql@cellwidth@\eql@frame@margin@
3978 \fi
3979 \ifdim\ht\eql@cellbox@>\eql@line@height@
3980 \global\eql@line@height@\ht\eql@cellbox@
3981 \fi
3982 \ifdim\dp\eql@cellbox@>\eql@line@depth@
3983 \global\eql@line@depth@\dp\eql@cellbox@
3984 \fi
3985 \ifnum\eql@column@=\@ne
3986 \eql@dimensions@startrow
3987 \fi
3988 \ifodd\eql@column@
3989 \eql@shape@pos@\tw@
3990 \else
3991 \eql@shape@pos@\z@
3992 \fi
3993 \eql@shape@amount@\z@
3994 \eql@dimensions@savcell
3995 \ifodd\eql@column@\else
3996 \eql@dimensions@savesep
3997 \fi
3998 \kern\eql@cellwidth@
3999 }

```

```

mns@measure@line@end

```

```

4000 \def\eql@columns@measure@line@end{%
4001 \eql@punct@apply@line
4002 \eql@hook@lineout
4003 &\omit
4004 \ifnum\eql@column@>\eql@totalcolumns@
4005 \global\eql@totalcolumns@\eql@column@
4006 \fi

```

TODO: not sure whether saving the last cell value makes sense, but rather not increase $\backslash\eql@totalcolumns@$ because that will disable the fallback to lines mode. **TODO:**

additional column in width table is accounted for in column table

```

4007 \ifdefined\eql@frame@cmd
4008 \advance\eql@column@\@ne
4009 \wd\eql@cellbox@\z@
4010 \eql@columns@measure@cell
4011 \fi
4012 \eql@measure@tag
4013 \eql@measure@endrow
4014 }

```

\eql@columns@measure

```

4015 \def\eql@columns@measure{%
4016 (dev)\eql@dev@center\eql@columns@measure
4017 \eql@totalcolumns@\z@
4018 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end

4019 \setbox\z@\vbox{\measuring@true\halign{%
4020 &%
4021 \global\advance\eql@column@\@ne
4022 \global\let\eql@cell@container\@empty
4023 \global\setbox\eql@cellbox@\hbox{%
4024 \eql@strut@cell
4025 \@lign
4026 $\m@th\displaystyle
4027 \eql@hook@colin
4028 ##%
4029 \eql@class@innerleft
4030 \eql@hook@innerleft
4031 $%
4032 }%
4033 \eql@cell@container
4034 \hfil
4035 \eql@columns@measure@cell
4036 \global\let\eql@frame@prevcmd\eql@frame@cmd
4037 &%
4038 \eql@prevwidth@\wd\eql@cellbox@
4039 \let\eql@frame@cmd\eql@frame@prevcmd
4040 \global\advance\eql@column@\@ne
4041 \global\let\eql@cell@container\@empty
4042 \setbox\eql@cellbox@\hbox{%
4043 \eql@strut@cell
4044 \@lign
4045 $\m@th\displaystyle
4046 \eql@hook@innerright
4047 \eql@class@innerright@sel
4048 ##%
4049 \eql@punct@apply@col
4050 \eql@hook@colout
4051 $%
4052 }%
4053 \eql@cell@container
4054 \eql@columns@measure@cell
4055 \hfil
4056 \crcr

4057 \noalign{%
4058 \eql@hook@blockbefore

```

```

4059 }%
4060 \eql@hook@blockin
4061 \eql@scan@body

4062 \ifvmode\else
4063 \eql@punct@apply@block
4064 \eql@hook@blockout
4065 \eql@display@endline
4066 \cr
4067 \fi
4068 \noalign{%
4069 \eql@hook@blockafter
4070 }%

```

TODO: note we also include the tag column as a backup

```

4071 \omit
4072 \eql@column@\@ne
4073 \eql@columns@completerow
4074 \cr
4075 }}%

4076 \eql@measure@close

4077 \setbox\z@\vbox{%
4078 \unvbox\z@
4079 \unpenalty
4080 \global\setbox\@ne\lastbox
4081 }%
4082 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4083 \let\eql@colwidth@tab\@empty
4084 \loop
4085 \setbox\@ne\hbox{%
4086 \unhbox\@ne
4087 \unskip
4088 \global\setbox\thr@@\lastbox
4089 }%
4090 \ifhbox\thr@@
4091 \eql@colwidth@save{\wd\thr@@}%
4092 \repeat

4093 <dev>\eql@dev@leave\eql@columns@measure
4094 }

```

14.4 Columns Placement

TODO: describe Make sure we have complete pairs of right and left adjusted columns, otherwise add a final empty column:

```

4095 \def\eql@columns@adjust{%
4096 \ifodd\eql@totalcolumns@
4097 \advance\eql@totalcolumns@\@ne
4098 \fi
4099 \eql@tagpos@adjust@eval
4100 \eql@adjust@calc@columns
4101 }

```

14.5 Print

TODO: describe

ms@print@line@begin

```
4102 \def\eq@columns@print@line@begin{%
4103 (dev)\eq@dev{starting line \the\eq@row}%
4104 \global\eq@column@z@
4105 \global\eq@line@pos@\eq@marginleft@
4106 \global\eq@line@width@z@
4107 \global\eq@line@avail@\eq@totalwidth@
4108 \global\eq@line@height@z@
4109 \global\eq@line@depth@z@
4110 \eq@numbering@print@line@begin
4111 \eq@hook@linein
4112 }
```

l@columns@print@cell

```
4113 \def\eq@columns@print@cell{%
4114 \eq@cellwidth@\wd\eq@cellbox@
4115 \ifodd\eq@column@
4116 \ifdefined\eq@frame@cmd
4117 \eq@frame@measure
4118 \advance\eq@cellwidth@\eq@frame@margin@
4119 \fi
4120 \dimen@z@
4121 \else
4122 \advance\eq@cellwidth@-\eq@prevwidth@
```

draw a frame

```
4123 \ifdefined\eq@frame@cmd
4124 \eq@frame@measure
4125 \advance\eq@cellwidth@\eq@frame@margin@
4126 \advance\eq@prevwidth@\eq@frame@margin@
4127 \eq@frame@print
4128 \fi
```

update height and depth

```
4129 \ifdim\ht\eq@cellbox@>\eq@line@height@
4130 \global\eq@line@height@\ht\eq@cellbox@
4131 \fi
4132 \ifdim\dp\eq@cellbox@>\eq@line@depth@
4133 \global\eq@line@depth@\dp\eq@cellbox@
4134 \fi
```

print box

```
4135 \kern-\eq@prevwidth@
4136 \unhbox\eq@cellbox@
4137 \dimen@-\eq@cellwidth@
4138 \fi
```

enforce given width: hopefully measure was correct, but need a precise width for tag placement

```
4139 \advance\dimen@\eq@colwidth@get\eq@column@\relax
4140 \kern\dimen@
```

update available and used space

```

4141 \dimen@eql@colwidth@get\eql@column@\relax
4142 \ifdim\eql@cellwidth@>\z@
4143   \ifdim\eql@line@width@=\z@
4144     \eql@line@avail@\eql@line@pos@
4145     \ifodd\eql@column@
4146       \advance\eql@line@avail@\dimen@
4147       \advance\eql@line@avail@-\eql@cellwidth@
4148     \fi
4149     \global\eql@line@avail@\eql@line@avail@
4150   \fi
4151   \eql@line@width@\eql@line@pos@
4152   \ifodd\eql@column@
4153     \advance\eql@line@width@\dimen@
4154   \else
4155     \advance\eql@line@width@\eql@cellwidth@
4156   \fi
4157   \global\eql@line@width@\eql@line@width@
4158 \fi
4159 \advance\eql@line@pos@\dimen@
4160 \ifodd\eql@column@\else
4161   \advance\eql@line@pos@\eql@colsep@
4162 \fi
4163 \global\eql@line@pos@\eql@line@pos@
4164 }

4165 \def\eql@columns@print@trailright{%
4166   &\omit
4167   \eql@prevwidth@\wd\eql@cellbox@
4168   \let\eql@frame@cmd\eql@frame@prevcmd
4169   \global\advance\eql@column@\@ne
4170   \eql@columns@print@cell
4171 }

```

lums@print@line@end

```

4172 \def\eql@columns@print@line@end{%
4173   \eql@punct@apply@line
4174   \eql@hook@lineout
4175 % \TODO add an even column with empty stuff if box processing deferred
4176   \ifodd\eql@column@
4177     \expandafter\eql@columns@print@trailright
4178   \fi
4179   \eql@columns@completerow
4180   \eql@columns@print@tag
4181 }

```

ql@columns@print@tag

```

4182 \def\eql@columns@print@tag{%
4183   \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4184 \eql@display@firstavail@set\eql@line@avail@
4185 \eql@columns@overfull
4186 \eql@numbering@print@line@eval
4187 \if@eqnsw
4188   \eql@tagbox@make\eql@composetag@print

```

```

4189 \fi
4190 \eql@tagpos@print@line@eval
4191 \eql@tagbox@print@cell
4192 \eql@tagpos@print@line@end
4193 }

```

\eql@columns@print

```

4194 \def\eql@columns@print{%
4195 (dev)\eql@dev@enter\eql@columns@print
4196 \eql@shape@align@disable
4197 \eql@display@halign@init\eql@columns@print@line@begin
4198 \eql@display@halign@letcr\eql@columns@print@line@end
4199 \tabskip\eql@marginleft@

4200 \halign{%
4201 &%
4202 \global\advance\eql@column@ \@ne
4203 \global\let\eql@cell@container \@empty
4204 \global\setbox\eql@cellbox@ \hbox{%
4205 \eql@strut@cell
4206 \@lign
4207 $\m@th\displaystyle
4208 \eql@hook@colin
4209 ##%
4210 \eql@class@innerleft
4211 \eql@hook@innerleft
4212 \eql@tagging@mathsave
4213 $%
4214 \eql@tagging@mathaddlast
4215 }%
4216 \eql@cell@container
4217 \hfil
4218 \eql@columns@print@cell
4219 \global\let\eql@frame@prevcmd\eql@frame@cmd
4220 \tabskip\z@skip
4221 &%
4222 \eql@prevwidth@ \wd\eql@cellbox@
4223 \let\eql@frame@cmd\eql@frame@prevcmd
4224 \global\advance\eql@column@ \@ne
4225 \global\let\eql@cell@container \@empty
4226 \setbox\eql@cellbox@ \hbox{%
4227 \unhbox\eql@cellbox@
4228 \eql@strut@cell
4229 \@lign
4230 $\m@th\displaystyle
4231 \eql@hook@innerright
4232 \eql@class@innerright@sel
4233 ##%
4234 \eql@punct@apply@col
4235 \eql@hook@colout
4236 \eql@tagging@mathsave
4237 $%
4238 \eql@tagging@mathaddlast
4239 }%
4240 \eql@cell@container
4241 \eql@columns@print@cell
4242 \hfil
4243 \tabskip\eql@colsep@ \relax

```

```

4244 \crrr

4245 \noalign{%
4246 \eqldisplay@halign@start
4247 \eqlnumbering@print@block@begin
4248 \eqlhook@blockbefore
4249 }%
4250 \eqlhook@blockin
4251 \eqlscan@body
4252 \ifvmode\else
4253 \relax
4254 \eqlpunct@apply@block
4255 \eqlhook@blockout
4256 \eqldisplay@endline
4257 \cr
4258 \fi
4259 \noalign{%
4260 \eqlhook@blockafter
4261 \eqldisplay@halign@end
4262 (dev)\eqldev@leave\eqlcolumns@print
4263 }%
4264 \eqltagging@tablesalign
4265 }%
4266 }

```

15 Interface

15.1 Scanning the Equation Body

The multi-line equatiuon environment must scan its body twice: once to determine how wide the columns are and then to actually typeset them. This means that we must collect all text in this body before calling the environment macros. The mechanism and its description follows `amsmath` closely.

Token Register.

`\eqlscan@reg@` We start by defining a token register to hold the equation body.

```
4267 \newtoks\eqlscan@reg@
```

`\eqlscan@body@dump` The macro `\eqlscan@body@dump` dumps the equation body from the register so that we do not have to pass it around in arguments. The macro `\eqlscan@body@rescan` rescans the tokens so that special commands such as `\verb` can be processed properly. The register `\eqlscan@body` holds the currently selected mode of operation:

```

4268 \def\eqlscan@body@dump{\the\eqlscan@reg@}
4269 \def\eqlscan@body@rescan{%
4270 \expandafter\scantokens\expandafter{\the\eqlscan@reg@}}
4271 \let\eqlscan@body\eqlscan@body@dump

```

`\eqlscan@addto` We define a macro to append to the token register `\eqlscan@reg@`:

```
4272 \long\def\eqlscan@addto#1{\eqlscan@reg@\expandafter{\the\eqlscan@reg@#1}}
```

Environment Body. The following mechanism scans the contents of an environment taking into account nested environments that may be contained in the body.

`\eql@scan@env` The macro `\eql@scan@env` starts the scan for the `\end{...}` command of the current environment. The argument is a call-back macro to process the body in `\eql@scan@reg@`:

```
4273 \def\eql@scan@env#1{%
4274 (dev)\eql@dev@enter\eql@scan@env
4275 \def\eql@scan@end{#1\expandafter\end\expandafter{\@currenvir}}%
4276 \eql@scan@reg@{\def\eql@scan@stack{b}}%
```

We call `\eql@scan@env@iterate` which will scan until the next occurrence of `\end` and then count the number of occurrences of `\begin` before `\end` in `\eql@scan@stack`. If we simply called `\eql@scan@env@iterate` directly, the error message for an unwanted `\par` token (usually from a blank line) would refer to `\eql@scan@env@iterate` which would not be illuminating. We use a little finesse to get a more intelligible error message: We use the actual environment name as the name of the temporary function that is `\let` to `\eql@scan@env@iterate`:

```
4277 \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
4278 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
4279 \ifdefined\eql@scan@par
4280 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4281 \else
4282 \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4283 \fi
4284 \eql@scan@iterate
4285 }
```

`\eql@scan@env@iterate` `\eql@scan@env@iterate` takes two arguments: the first will consist of all text up to the next `\end` command, the second will be the `\end` command's argument. If there are any extra `\begin` commands in the body text, a marker is pushed onto a stack via `\eql@scan@env@count`. An empty state for this stack means that we have reached the `\end` that matches our original `\begin`. Otherwise we need to include the `\end` and its argument in the material that we are adding to our environment body accumulator:

```
4286 \long\def\eql@scan@env@iterate#1\end#2{%
4287 \edef\eql@scan@stack{%
4288 \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4289 \ifx\@empty\eql@scan@stack
4290 \@checkend{#2}%
4291 \eql@scan@addto{#1}%
4292 \expandafter\let\eql@scan@iterate\eql@scan@env@org
4293 (dev)\eql@dev@leave\eql@scan@env
4294 \expandafter\eql@scan@end
4295 \else
4296 \eql@scan@addto{#1\end{#2}}%
4297 \expandafter\eql@scan@iterate
4298 \fi
4299 }
```

`\eql@scan@env@iterate@nopar` Version of `\eql@scan@env@iterate` which does not accept `\par` within the argument:

```
4300 \def\eql@scan@env@iterate@nopar#1\end#2{\eql@scan@env@iterate#1\end{#2}}
```

`\eql@scan@env@count` When adding a piece of the current environment's contents to `\eql@scan@reg@`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

4301 \long\def\eql@scan@env@count#1\begin#2{%
4302   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4303 }

```

The call-back macro `\eql@scan@env@cancel` ignores the body as well as the end clause for the environment:

```

4304 \def\eql@scan@env@cancel{%
4305   \@namedef{end\@currentvir}{\ignorespacesafterend}%
4306 }

```

Square Brackets. The following is a version of the above mechanism that scans for an equation body enclosed by `\[...]` paying attention to potential further instances of the square bracket enclosures contained in the body.

`\eql@scan@sqr` Start scanning for `\[`:

```

4307 \def\eql@scan@sqr#1{%
4308 (dev)\eql@dev@enter\eql@scan@sqr
4309   \def\eql@scan@end{#1\}}%
4310   \eql@scan@reg@{\def\eql@scan@stack{b}%
4311     \let\eql@scan@sqr@org\[%\]
4312     \ifdefined\eql@scan@par
4313       \let\[\eql@scan@sqr@iterate%\]
4314     \else
4315       \let\[\eql@scan@sqr@iterate@nopar%\]
4316     \fi
4317     \[%\]
4318 }

```

`\eql@scan@sqr@iterate` Iterate until we find a balanced pairing of square brackets. Then call the call-back macro:

```

4319 \long\def\eql@scan@sqr@iterate#1\{%
4320   \edef\eql@scan@stack{%
4321     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4322   \ifx\@empty\eql@scan@stack
4323     \let\[\eql@scan@sqr@org%\]
4324     \eql@scan@addto{#1}%
4325 (dev)\eql@dev@leave\eql@scan@sqr
4326   \expandafter\eql@scan@end
4327   \else
4328     \eql@scan@addto{#1\}}%
4329   \expandafter\[%\]
4330   \fi
4331 }

```

`\eql@scan@sqr@iterate@nopar` Version of `\eql@scan@sqr@iterate` which does not accept `\par` within the argument:

```

4332 \def\eql@scan@sqr@iterate@nopar#1\{\eql@scan@sqr@iterate#1\}}

```

`\eql@scan@sqr@count` Push a ‘b’ for every encountered instance of ‘`\[`’:

```

4333 \long\def\eql@scan@sqr@count#1\[#2{%\]
4334   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4335 }

```

`\eql@scan@sqrang@cancel` The call-back macro `\eql@scan@sqrang@cancel` ignores the body and the closing bracket:

```

4336 \def\eql@scan@sqrang@cancel{\expandafter\ignorespaces\@gobble}

```

Angle Brackets. The following is another version of the mechanism which scans for an equation body enclosed by `\<... \>`.

`\eql@scan@ang` Start scanning for `\>`:

```
4337 \def\eql@scan@ang#1{%
4338 (dev)\eql@dev@enter\eql@scan@ang
4339 \def\eql@scan@end{#1\>}%
4340 \eql@scan@reg@{ }\def\eql@scan@stack{b}%
4341 \let\eql@scan@ang@org\<%\>
4342 \ifdefined\eql@scan@par
4343 \let\<\eql@scan@ang@iterate%\>
4344 \else
4345 \let\<\eql@scan@ang@iterate@nopar%\>
4346 \fi
4347 \<%\>
4348 }
```

`\eql@scan@ang@iterate` Iterate until we find a balanced pairing of angle brackets:

```
4349 \long\def\eql@scan@ang@iterate#1\>{%
4350 \edef\eql@scan@stack{%
4351 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4352 \ifx\@empty\eql@scan@stack
4353 \let\<\eql@scan@ang@org%\>
4354 \eql@scan@addto{#1}%
4355 (dev)\eql@dev@leave\eql@scan@ang
4356 \expandafter\eql@scan@end
4357 \else
4358 \eql@scan@addto{#1\>}%
4359 \expandafter\<%\>
4360 \fi
4361 }
```

`\an@ang@iterate@nopar` Version of `\eql@scan@ang@iterate` which does not accept `\par` within the argument:

```
4362 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}
```

`\eql@scan@ang@count` Push a ‘b’ for every encountered instance of ‘`\<`’:

```
4363 \long\def\eql@scan@ang@count#1\<#2{%\>
4364 \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4365 }
```

15.2 Options Processing

`\eql@equations@testall` The macro sequence started by `\eql@equations@testall` scans for optional arguments to the equation environments and appends them to the argument list using `\eqnaddopt`. All arguments are scanned such that any spaces stop the scanning and such that any alignment markers ‘&’ cannot interfere: **TODO:** update

```
4366 \def\eql@equations@testall{\eql@parseopt\eql@equations@parseopt}
4367 \def\eql@equations@parseopt{%
4368 \ifx\eql@parseopt@token*%
4369 \let\eql@parseopt@next\eql@parseopt@nonumber
4370 \fi
4371 \ifx\eql@parseopt@token!%
4372 \let\eql@parseopt@next\eql@parseopt@donumber
```

```

4373 \fi
4374 \ifx\eql@parseopt@token/%
4375   \let\eql@parseopt@next\eql@parseopt@transpose
4376 \fi
4377 \ifx\eql@parseopt@token[%]
4378   \let\eql@parseopt@next\eql@parseopt@opt
4379 \fi
4380 \ifx\eql@parseopt@token\eql@atxi
4381   \let\eql@parseopt@next\eql@parseopt@label
4382 \fi
4383 \ifx\eql@parseopt@token\eql@atxii
4384   \let\eql@parseopt@next\eql@parseopt@label
4385 \fi
4386 \ifx\eql@parseopt@token.%
4387   \let\eql@parseopt@next\eql@parseopt@punctdot
4388 \fi
4389 \ifx\eql@parseopt@token,%
4390   \let\eql@parseopt@next\eql@parseopt@punctcomma
4391 \fi
4392 \ifx\eql@parseopt@token~%
4393   \let\eql@parseopt@next\eql@parseopt@punctoff
4394 \fi
4395 \ifx\eql@parseopt@token'%
4396   \let\eql@parseopt@next\eql@parseopt@punctall
4397 \fi
4398 \ifx\eql@parseopt@token-%
4399   \let\eql@parseopt@next\eql@parseopt@single
4400 \fi
4401 \ifx\eql@parseopt@token=%
4402   \let\eql@parseopt@next\eql@parseopt@lines
4403 \fi
4404 \ifx\eql@parseopt@token|%
4405   \let\eql@parseopt@next\eql@parseopt@columns
4406 \fi
4407 \ifx\eql@parseopt@token\label
4408   \let\eql@parseopt@next\eql@parseopt@end
4409 \fi
4410 \ifx\eql@parseopt@token\begin
4411   \let\eql@parseopt@next\eql@parseopt@end
4412 \fi
4413 }

```

`\equations@processopt` The macro `\eql@equations@processopt` processes the options received by `\eqnaddopt`. First, clear several non-persistent registers (labels, tags, direct vertical spacing). Then process the arguments. Finally evaluate `\eql@indent@val` and `\eql@tagsepmin@val` and prevent main punctuation from being passed to nested environments:

```

4414 \def\eql@equations@processopt{%
4415   \let\eql@tags@container@block\eql@tags@container@clear
4416   \let\eql@tags@frame@cmd\@firstofone
4417   \let\eql@skip@force@above\@undefined
4418   \let\eql@skip@force@below\@undefined
4419   \let\eql@skip@force@leave\@undefined
4420   \let\eql@display@linewidth\@undefined
4421   \let\eql@display@marginleft\@undefined
4422   \let\eql@display@marginright\@undefined
4423   \eql@abovespace@\z@skip
4424   \eql@belowspace@\z@skip

```

```

4425 \eqldisplaybreak@prepen@\@MM
4426 \eqldisplaybreak@postpen@\@MM
4427 \eqlnextopt@process{equations}%
4428 \let\eqlpunct@block\eqlpunct@main
4429 \let\eqlpunct@main\relax
4430 \eqlindent@\glueexpr\eqlindent@val\relax
4431 \eqltagsepmin@\glueexpr\eqltagsepmin@val\relax
4432 }

```

15.3 Single-Line Main

In the following, we define the main routine for the single-line equation mode.

`\eqlsingle@cr@error` Cannot use line breaks, produce an error message:

```

4433 \def\eqlsingle@cr@error{%
4434   \eql@error{Cannot use '\string\\' within display equation.
4435     Please switch to equations environment}%
4436 }

```

`\eqlsingle@start` Opening code for single-line equation. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, set native vs. manual equation tag mode, install error message for using `\.`. Hand over to mode-specific opening:

```

4437 \def\eqlsingle@start{%
4438   \eqldisplay@enter
4439   \eql@tagging@start
4440   \eqldollardollar@begin
4441   \eqldisplay@adjust
4442   \eql@numbering@init
4443   \eql@stack@save@equations
4444   \eql@numbering@single@init
4445   \ifdefined\eqlsingle@cr@mode
4446     \let\\\eqlsingle@cr@mode
4447   \fi
4448   \ifdefined\eqlsingle@native
4449     \let\eqlsingle@start@sel\eqlsingle@start@native
4450     \let\eqlsingle@end@sel\eqlsingle@end@native
4451   \else
4452     \let\eqlsingle@start@sel\eqlsingle@start@print
4453     \let\eqlsingle@end@sel\eqlsingle@end@print
4454   \fi
4455   \eqlsingle@start@sel
4456 }

```

`\eqlsingle@end` Closing code for single-line equation. Apply punctuation for the block, perform mode-specific ending, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

4457 \def\eqlsingle@end{%
4458   \eqlpunct@apply@block
4459   \eql@hook@eqout
4460   \eqlsingle@end@sel
4461   \eql@stack@restore
4462   \eqldollardollar@end
4463   \eql@tagging@end

```

```

4464 \eqldisplay@leave
4465 }

```

`\eqlsingle@main` Combined opening, body and closing for pre-scanned body: **TODO:** is `\expandafter` needed? relic?

```

4466 \def\eqlsingle@main{%
4467   \expandafter\eqlsingle@start
4468   \eql@scan@body
4469   \eqlsingle@end
4470 }

```

`\eql@mode@single` Configure equations macros to single-line mode:

```

4471 \def\eql@mode@single{%
4472   \ifdefined\eqlsingle@doscan
4473     \let\eqlequations@main\eqlsingle@main
4474     \let\eqlequations@end\@empty
4475   \else
4476     \let\eqlequations@main\@undefined
4477     \let\eqlequations@end\eqlsingle@end
4478   \fi
4479 }

```

15.4 Multi-Line Main

`\eql@mode@lines` (*bool*) Switch register for lines vs. columns mode:

```

4480 \let\eql@multi@mode@lines\eql@false

```

`\eql@multi@main` Main routine for multi-line modes. Capture current vertical mode, trigger PDF tagging, enter display math mode, initialise numbering scheme, backup current state of global registers, initialise macros for use within equations: **TODO:** shove depends on lines vs columns

```

4481 \def\eql@multi@main{%
4482   \eqldisplay@enter
4483   \eql@tagging@start
4484   \eqldollardollar@begin
4485   \eqldisplay@adjust
4486   \eql@numbering@init
4487   \eql@stack@save@equations
4488   \ifdefined\eql@transpose@active
4489     \ifdefined\eql@multi@mode@lines\else
4490       \eql@transpose
4491     \fi
4492   \fi
4493   \ifdefined\eql@numbering@subeq@use
4494     \eql@numbering@subeq@init
4495   \fi
4496   \eqldisplay@init
4497   \let\intertext\eql@intertext
4498   \let\endintertext\endeql@intertext
4499   \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

4500   \ifdefined\eql@multi@mode@lines

```

```

4501   \eql@lines@measure
4502 \else
4503   \ifdefined\eql@ampproof@active
4504     \eql@ampproof
4505   \fi
4506   \eql@columns@measure
4507 \fi

```

If only a single equation number is used for subequation numbering, revert to normal equation numbering. If only a single column is used in columns mode, may fallback to lines mode. Switching from columns to lines mode, the width can be incorrect, expect only minor discrepancies, but for accurateness, should call `\eql@lines@measure`:

```

4508 \ifdefined\eql@numbering@subeq@use
4509   \eql@numbering@subeq@test
4510 \fi
4511 \ifdefined\eql@multi@mode@lines\else
4512   \ifdefined\eql@multi@lines@fallback
4513     \ifnum\eql@totalcolumns@=\@ne
4514       \let\eql@multi@mode@lines\eql@true
4515       \ifx\eql@multi@lines@fallback\z@\else
4516         \eql@lines@measure
4517       \fi
4518     \fi
4519   \fi
4520 \fi

```

Adjust the multi-line equations body:

```

4521 \ifdefined\eql@multi@mode@lines
4522   \eql@lines@adjust
4523 \else
4524   \eql@columns@adjust
4525 \fi

```

Now print the multi-line equations body:

```

4526 \eql@display@print
4527 \eql@numbering@print@init
4528 \ifdefined\eql@multi@mode@lines
4529   \eql@lines@print
4530 \else
4531   \eql@columns@print
4532 \fi
4533 \eql@display@close

```

Close numbering, restore global variables, end display math, indicate end to PDF tagging, return to vertical mode if desired:

```

4534 \ifdefined\eql@numbering@subeq@use
4535   \eql@numbering@subeq@close
4536 \fi
4537 \eql@stack@restore
4538 \eql@dollar@dollar@end
4539 \eql@tagging@end
4540 \eql@display@leave
4541 }

```

`\eql@mode@columns` Configure equations macros to one of the two multi-line modes:

```

\eql@mode@lines
4542 \def\eql@mode@columns{%

```

```

4543 \let\eqlequations@main\eql@multi@main
4544 \let\eqlequations@end\@empty
4545 \let\eql@multi@mode@lines\eql@false
4546 }
4547 \def\eql@mode@lines{%
4548 \let\eqlequations@main\eql@multi@main
4549 \let\eqlequations@end\@empty
4550 \let\eql@multi@mode@lines\eql@true
4551 }

```

15.5 Equations Environment

We now declare the main environment and its symbolic versions.

Environment.

equations (*env.*) Declare the main equations environment. If already in math mode, fail and cancel the environment body. Otherwise scan for optional arguments and pass on to `\eqlequations@start`:

```

4552 \newenvironment{equations}{%
4553 (dev)\eql@dev@enterenv
4554 \ifmmode
4555 \eqLError@mathmode{\string\begin{\@currentvir}}%
4556 \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4557 \else
4558 \expandafter\eql@ampprotect\expandafter\eqlequations@testall
4559 \expandafter\eqlequations@start
4560 \fi
4561 }{%
4562 \eqlequations@end
4563 \ignorespacesafterend
4564 (dev)\eql@dev@leaveenv
4565 }
4566 \eql@markline@amsthm@register{equations}
4567 \eql@tagging@register@luamml{equations}

```

`\eqlequations@start` The macro `\eqlequations@start` first processes the arguments. Depending on the chosen mode of operation, scan the environment body passing on to `\eqlequations@main` or process a single-line equation via `\eql@single@start`:

```

4568 \def\eqlequations@start{%
4569 \eqlequations@processopt
4570 \ifdefined\eqlequations@main
4571 \expandafter\eql@scan@env\expandafter\eqlequations@main
4572 \else
4573 \expandafter\eql@single@start
4574 \fi
4575 }

```

Square Brackets.

equations@sqr (*env.*) Define a pseudo-environment `equations@sqr` such that `\@currentvir` may point to it when needed:

```

4576 \newenvironment{equations@sqr}{}{}

```

```

4577 \eql@markline@amsthm@register{equations@sqr}
4578 \eql@tagging@register@luamml{equations@sqr}

```

`\eql@equations@sqr@open` Definition for ‘`\[`’. If already in math mode, ignore the enclosed contents. Otherwise add the default arguments `\eql@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eql@equations@sqr@start`:

```

4579 \protected\def\eql@equations@sqr@open{%
4580   \ifmmode
4581     \eql@error@mathmode{\string\[...\string\]}%
4582     \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4583   \else
4584 (dev)\eql@dev@enter{\string\[...\string\]}%
4585     \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
4586     \begin{equations@sqr}%
4587     \let\]\eql@equations@sqr@close
4588     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4589     \expandafter\eql@equations@sqr@start
4590   \fi
4591 }

```

`\eql@equations@sqr@start` Process arguments. Depending on mode of operation, scan and process enclosed contents via `\eql@equations@main` or pass on to `\eql@single@start`:

```

4592 \def\eql@equations@sqr@start{%
4593   \eql@equations@processopt
4594   \ifdefined\eql@equations@main
4595     \expandafter\eql@scan@sqr\expandafter\eql@equations@main
4596   \else
4597     \expandafter\eql@single@start
4598   \fi
4599 }

```

`\eql@equations@sqr@close` Definition for ‘`\]`’:

```

4600 \protected\def\eql@equations@sqr@close{%
4601   \eql@equations@end
4602 (dev)\eql@dev@leave{\string\]}%
4603   \end{equations@sqr}%
4604   \ignorespaces
4605 }

```

TODO: describe

```

\eql@sqr@open
\eql@sqr@close
4606 \let\eql@sqr@open\eql@equations@sqr@open
4607 \protected\def\eql@sqr@close{%
4608   \eql@error{'\string\]' may only close '\string\[']}%\]
4609 }

```

Angle Brackets.

`\equations@ang` (*env.*) Define a pseudo-environment `\equations@ang`:

```

4610 \newenvironment{equations@ang}{}{}
4611 \newenvironment{equationsbox@ang}{}{}
4612 \eql@markline@amsthm@register{equations@ang}
4613 \eql@tagging@register@luamml{equations@ang}

```

`\eql@ang@open` Definition for ‘<’. Forward to equationsbox if in math mode, otherwise to equations:

```

4614 \protected\def\eql@ang@open{%
4615 (dev)\eql@dev@enter{\<...\string\>}%
4616 \ifmmode
4617   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4618   \begin{equationsbox@ang}%
4619   \let\>\eql@box@ang@close
4620   \expandafter\eql@ampprotect\expandafter\eql@box@testall
4621   \expandafter\eql@box@start
4622 \else
4623   \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
4624   \begin{equations@ang}%
4625   \let\>\eql@equations@ang@close
4626   \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4627   \expandafter\eql@equations@ang@start
4628 \fi
4629 }
```

`\eql@ang@close` Definition for ‘>’: **TODO:** NOTE: `\protected` acts as `\relax` and starts a row in `\halign`, so we overwrite `\>` when starting.

```

4630 \protected\def\eql@ang@close{%
4631   \eql@error{'\string\>' may only close '\string\<'}%\>
4632 }
```

`@equations@ang@start` Process arguments and start handling the equation:

```

4633 \def\eql@equations@ang@start{%
4634   \eql@equations@processopt
4635   \ifdefined\eql@equations@main
4636     \expandafter\eql@scan@ang\expandafter\eql@equations@main
4637   \else
4638     \expandafter\eql@single@start
4639   \fi
4640 }
```

`@equations@ang@close` **TODO:** describe

```

4641 \def\eql@equations@ang@close{%
4642   \eql@equations@end
4643   \end{equations@ang}%
4644 (dev)\eql@dev@leave{\<...\string\>}%
4645   \ignorespaces
4646 }
```

`\eql@box@ang@close` **TODO:** describe

```

4647 \def\eql@box@ang@close{%
4648   \eql@box@end
4649   \end{equationsbox@ang}%
4650 (dev)\eql@dev@leave{\<...\string\>}%
4651   \ignorespaces
4652 }
```

16 Options

16.1 Selection Tools

`eq\decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```
4653 \def\eq\decide@abovebelow#1#2#3#4#5{%
4654   \eq\decide@select{#1}{#2}{#3}{%
4655     {,abovebelow,both,tb}{#4#5},%
4656     {above,top,t}{#4},%
4657     {below,bottom,b}{#5}}}
```

`eq\decide@situation` Select a particular vertical spacing situation and store it in the macro #4:

```
4658 \def\eq\decide@situation#1#2#3#4{%
4659   \eq\decide@select{#1}{#2}{#3}{%
4660     {{long}}{\def#4{0}}},%
4661     {{short}}{\def#4{1}}},%
4662     {{cont}}{\def#4{2}}},%
4663     {{par}}{\def#4{3}}},%
4664     {{top}}{\def#4{4}}},%
4665     {{noskip}}{\def#4{5}}},%
4666     {{medskip}}{\def#4{6}}}}}
```

16.2 Options Declarations

We now declare all key-value pairs for options sorted by their category.

Modes for Equations Box Environment. Declare horizontal and vertical alignment modes for the boxed equations environment. Also declare spacing of columns:

```
4667 \eq\define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{}%
4668   \eq\mode@stacked}
4669 \eq\define@key{equationsbox}{aligned,align,al,columns,col}[]{}%
4670   \eq\mode@aligned}
4671 \eq\define@key{equationsbox}{top,t}[]{\let\eq\box@box\vtop}
4672 \eq\define@key{equationsbox}{center,c}[]{\let\eq\box@box\vcenter}
4673 \eq\define@key{equationsbox}{bottom,b}[]{\let\eq\box@box\vbbox}
4674 \eq\define@key{setup}{boxangopt}[]{}%
4675   \def\eq\box@ang@opt{columns,#1}}
```

Modes for Equations Environment. Declare modes and switches for the equations environment:

```
4676 \eq\define@key{equations}{equation,eq,single,1}[]{\eq\mode@single}
4677 \eq\define@key{equations}{gathered,gather,ga,lines,ln}[]{}%
4678   \eq\mode@lines}
4679 \eq\define@key{equations}{aligned,align,al,columns,col}[]{}%
4680   \eq\mode@columns}
4681 \eq\define@key{equations,setup}{transpose}[true]{}%
4682   \eq\decide@select{#3}{#2}{#1}{%
4683     {\eq\decide@false{\let\eq\transpose@active\eq\false}},%
4684     {{noamp,plain,restricted}}{\let\eq\transpose@active\eq\true}},%
4685     {{\eq\decide@true,amp,cont}}{\let\eq\transpose@active=+}}}%
4686 \eq\define@key{equations}{native}[true]{}%
```

```

4687 \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
4688 \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
4689 \eql@define@key{setup}{native}[true]{%
4690 \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
4691 \eql@define@key{setup}{scanequation}[true]{%
4692 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
4693 \eql@define@key{setup}{sqropt}[]{%
4694 \def\eql@equations@sqr@opt{equation,#1}}
4695 \eql@define@key{setup}{angopt}[]{%
4696 \def\eql@equations@ang@opt{columns,#1}}

```

Vertical Spacing. Settings concerning the spacing of lines: **TODO:** set at end of env only!

```

4697 \def\eql@keycat{equations,equationsbox,setup}
4698 \eql@define@key\eql@keycat{spread}{\def\eql@spread@val{#1}}
4699 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
4700 {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
4701 {\cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
4702 {\tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
4703 {\eql@decide@true
4704 {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
4705 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings concerning page breaks:

```

4706 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4707 {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
4708 {{high,3}{\eql@displaybreak@pre3}},%
4709 {{med,medium,2}{\eql@displaybreak@pre2}},%
4710 {{low,1}{\eql@displaybreak@pre1}},%
4711 {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
4712 {{default,inherit,-1}{\eql@displaybreak@pre@mone}}}}
4713 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
4714 {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
4715 {{high,3}{\eql@displaybreak@post3}},%
4716 {{med,medium,2}{\eql@displaybreak@post2}},%
4717 {{low,1}{\eql@displaybreak@post1}},%
4718 {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
4719 {{default,inherit,-1}{\eql@displaybreak@post@mone}}}}
4720 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4721 \eql@decide@select{#3}{#2}{#1}{%
4722 {{full,4}{\eql@displaybreak@inter4}},%
4723 {{high,3}{\eql@displaybreak@inter3}},%
4724 {{med,medium,2}{\eql@displaybreak@inter2}},%
4725 {{low,1}{\eql@displaybreak@inter1}},%
4726 {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
4727 \eql@define@key{equations}{prepenalty}{%
4728 \eql@displaybreak@prepen@numexpr#1\relax}
4729 \eql@define@key{equations}{postpenalty}{%
4730 \eql@displaybreak@postpen@numexpr#1\relax}
4731 \eql@define@key{equations,setup}{interpenalty}{%
4732 \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4733 \eql@define@key{control}{vspace}[]{\eql@vspace@add{#1}}
4734 \eql@define@key{control}{vspace*}[]{\eql@vspace@addfixedbefore{#1}}
4735 \eql@define@key{control}{vspace!}[]{\eql@vspace@addfixedafter{#1}}

```

```

4736 \eqld@define@key{control}{break}[4]{\eqld@displaybreak@level[{#1}]}
4737 \eqld@define@key{control}{penalty}[]{\eqld@displaybreak@star{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4738 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
4739   \eqld@decide@select{#3}{#2}{#1}{%
4740     {\eqld@decide@false{\let\eqld@display@height\@undefined}},%
4741     {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
4742     {\relax{\def\eqld@display@height{#1}}}}}%
4743 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
4744   \eqld@decide@select{#3}{#2}{#1}{%
4745     {\eqld@decide@false{\let\eqld@display@depth\@undefined}},%
4746     {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
4747     {\relax{\def\eqld@display@depth{#1}}}}}%

```

Override vertical spacing situation: **TODO**: short should just apply to above?! or as far as short would apply...

```

4748 \eqld@define@key{equations}{noskip}[]{%
4749   \eqld@decide@abovebelow{#3}{#2}{#1}%
4750   {\def\eqld@skip@force@above{5}}%
4751   {\def\eqld@skip@force@below{5}}}
4752 \eqld@define@key{equations}{short}[above]{%
4753   \eqld@decide@abovebelow{#3}{#2}{#1}%
4754   {\def\eqld@skip@force@above{1}}%
4755   {\def\eqld@skip@force@below{1}}}
4756 \eqld@define@key{equations}{long}[]{%
4757   \eqld@decide@abovebelow{#3}{#2}{#1}%
4758   {\def\eqld@skip@force@above{0}}%
4759   {\def\eqld@skip@force@below{0}}}
4760 \eqld@define@key{equations}{medskip}[]{%
4761   \eqld@decide@abovebelow{#3}{#2}{#1}%
4762   {\def\eqld@skip@force@above{6}}%
4763   {\def\eqld@skip@force@below{6}}}
4764 \eqld@define@key{equations}{par}[par]{%
4765   \eqld@decide@select{#3}{#2}{#1}{%
4766     {{default},{\let\eqld@skip@force@leave\@undefined}},%
4767     {{cont,hmode}{\let\eqld@skip@force@leave\z@}},%
4768     {{par,vmode}{\let\eqld@skip@force@leave\@ne
4769       \ifdefined\eqld@skip@force@below\else
4770       \def\eqld@skip@force@below{3}%
4771       \fi}},%
4772     {{top}{\let\eqld@skip@force@leave\tw@
4773       \ifdefined\eqld@skip@force@below\else
4774       \def\eqld@skip@force@below{4}
4775       \fi}}}}}%

```

Specify vertical spacing explicitly:

```

4776 \eqld@define@key{equations}{skip}{%
4777   \def\eqld@skip@force@above{7}%
4778   \def\eqld@skip@custom@above{#1}%
4779   \let\eqld@skip@force@below\eqld@skip@force@above
4780   \let\eqld@skip@custom@below\eqld@skip@custom@above}%
4781 \eqld@define@key{equations}{aboveskip}{%
4782   \def\eqld@skip@force@above{7}%
4783   \def\eqld@skip@custom@above{#1}}
4784 \eqld@define@key{equations}{belowskip}{%
4785   \def\eqld@skip@force@below{7}%

```

```

4786 \def\eql@skip@custom@below{#1}}
4787 \eql@define@key{equations}{abovespace}{%
4788 \advance\eql@abovespace@glueexpr#1\relax}
4789 \eql@define@key{equations}{belowspace}{%
4790 \advance\eql@belowspace@glueexpr#1\relax}

```

Vertical spacing for intertext:

```

4791 \eql@define@key{intertext}{skip}{%
4792 \def\eql@skip@force@above{7}%
4793 \def\eql@skip@custom@above{#1}%
4794 \let\eql@skip@force@below\eql@skip@force@above
4795 \let\eql@skip@custom@below\eql@skip@custom@above}
4796 \eql@define@key{intertext}{aboveskip}{%
4797 \def\eql@skip@force@below{7}%
4798 \def\eql@skip@custom@below{#1}}
4799 \eql@define@key{intertext}{belowskip}{%
4800 \def\eql@skip@force@above{7}%
4801 \def\eql@skip@custom@above{#1}}
4802 \eql@define@key{intertext}{noskip}[]{%
4803 \eql@decide@abovebelow{#3}{#2}{#1}%
4804 {\def\eql@skip@force@below{5}}%
4805 {\def\eql@skip@force@above{5}}}
4806 \eql@define@key{intertext}{short}[]{%
4807 \eql@decide@abovebelow{#3}{#2}{#1}%
4808 {\def\eql@skip@force@below{1}}%
4809 {\def\eql@skip@force@above{1}}}
4810 \eql@define@key{intertext}{long}[]{%
4811 \eql@decide@abovebelow{#3}{#2}{#1}%
4812 {\def\eql@skip@force@below{0}}%
4813 {\def\eql@skip@force@above{0}}}
4814 \eql@define@key{intertext}{medskip}[]{%
4815 \eql@decide@abovebelow{#3}{#2}{#1}%
4816 {\def\eql@skip@force@below{6}}%
4817 {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

4818 \eql@define@key{setup}{skip,longskip}{%
4819 \abovedisplayskip\glueexpr#1\relax
4820 \belowdisplayskip\abovedisplayskip
4821 \def\eql@skip@long@above{#1}%
4822 \let\eql@skip@long@below\eql@skip@long@above}
4823 \eql@define@key{setup}{aboveskip,abovelongskip}{%
4824 \abovedisplayskip\glueexpr#1\relax
4825 \def\eql@skip@long@above{#1}}
4826 \eql@define@key{setup}{belowskip,belowlongskip}{%
4827 \belowdisplayskip\glueexpr#1\relax
4828 \def\eql@skip@long@below{#1}}
4829 \eql@define@key{setup}{aboveshortskip}{%
4830 \abovedisplayshortskip\glueexpr#1\relax
4831 \def\eql@skip@short@above{#1}}
4832 \eql@define@key{setup}{belowshortskip}{%
4833 \belowdisplayshortskip\glueexpr#1\relax
4834 \def\eql@skip@short@below{#1}}
4835 \eql@define@key{setup}{tagskip}{%
4836 \def\eql@skip@tag@above{#1}%
4837 \let\eql@skip@tag@below\eql@skip@tag@above}
4838 \eql@define@key{setup}{abovetagskip}{%
4839 \def\eql@skip@tag@above{#1}}

```

```

4840 \eqld@define@key{setup}{belowtagskip}{%
4841   \def\eqld@skip@tag@below{#1}}
4842 \eqld@define@key{setup}{medskip}{%
4843   \def\eqld@skip@med@above{#1}%
4844   \let\eqld@skip@med@below\eqld@skip@med@above}
4845 \eqld@define@key{setup}{abovemedskip}{%
4846   \def\eqld@skip@med@above{#1}}
4847 \eqld@define@key{setup}{belowmedskip}{%
4848   \def\eqld@skip@med@below{#1}}
4849 \eqld@define@key{setup}{abovetopskip}{%
4850   \def\eqld@skip@top@above{#1}}
4851 \eqld@define@key{setup}{belowtopskip}{%
4852   \def\eqld@skip@top@below{#1}}
4853 \eqld@define@key{setup}{aboveparskip}{%
4854   \def\eqld@skip@par@above{#1}}
4855 \eqld@define@key{setup}{belowparskip}{%
4856   \def\eqld@skip@par@below{#1}}
4857 \eqld@define@key{setup}{abovecontskip}{%
4858   \eqld@decide@select{#3}{#2}{#1}{%
4859     {\hide}{\def\eqld@skip@cont@above{\eqld@spread@val-\eqld@skip@long@below}}},%
4860     {\relax{\def\eqld@skip@cont@above{#1}}}}}%
4861 \eqld@define@key{setup}{belowcontskip}{%
4862   \def\eqld@skip@cont@below{#1}}
4863 \eqld@define@key{setup}{shortmode}{%
4864   \eqld@decide@select{#3}{#2}{#1}{%
4865     {\off,never,no}{\def\eqld@skip@mode@short{0}}},%
4866     {\above,neverbelow,notbelow,belowoff}{\def\eqld@skip@mode@short{1}}},%
4867     {\belowone,belowsingle}{\def\eqld@skip@mode@short{2}}},%
4868     {\belowall,always,on}{\def\eqld@skip@mode@short{3}}}}}%
4869 \eqld@define@key{setup}{abovecontmode}{%
4870   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@above}
4871 \eqld@define@key{setup}{belowcontmode}{%
4872   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@below}
4873 \eqld@define@key{setup}{aboveparmode}{%
4874   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@above}
4875 \eqld@define@key{setup}{belowparmode}{%
4876   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@below}
4877 \eqld@define@key{setup}{abovetopmode}{%
4878   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@above}
4879 \eqld@define@key{setup}{belowtopmode}{%
4880   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@below}

```

Labels and Tag Declaration. Specify label and tag for equations and subequations:

```

4881 \def\eqld@keycat{equations,subequations}
4882 \eqld@define@key\eqld@keycat{label}{\eqld@tags@addblock@label{#1}}
4883 \eqld@define@key\eqld@keycat{labelname}{\eqld@tags@addblock@name{#1}}
4884 \eqld@define@key\eqld@keycat{tag}{\eqld@tags@addblock@tag{#1}}
4885 \eqld@define@key\eqld@keycat{tag*}{%
4886   \eqld@tags@addblock@tagform@off\eqld@tags@addblock@tag{#1}}
4887 \eqld@define@key\eqld@keycat{taglabel}{\eqld@tags@addblock@ref{#1}}

```

TODO: describe

```

4888 \eqld@define@key{control}{label}{\eqld@tags@add@label{#1}}
4889 \eqld@define@key{control}{labelname}{\eqld@tags@add@name{#1}}
4890 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
4891 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
4892 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}

```

```

4893 \eqld@define@key{control}{\shifftag}{\eqld@tags@add@raiseshift{#1}}
4894 \eqld@define@key{control}{\smashtag}{\eqld@tags@add@raisesmash{#1}}
4895 \eqld@define@key{control}{\pushtag}{\eqld@tags@add@forceraise}

```

TODO: describe

```

4896 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
4897 \eqld@define@key{setup}{autolabel}[true]{%
4898   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
4899 \eqld@define@key{setup}{autotag}[true]{%
4900   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

4901 \def\eqld@keycat{equations,setup}
4902 \eqld@define@key\eqld@keycat{tagmargin}[auto]{%
4903   \eqld@decide@select{#3}{#2}{#1}{%
4904     {auto,\eqld@decide@false}{\let\eqld@tagmargin@val\@undefined}},%
4905     {\relax{\def\eqld@tagmargin@val{#1}}}}}%
4906 \eqld@define@key\eqld@keycat{tagmargin*}{%
4907   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
4908 \eqld@define@key\eqld@keycat{tagmarginratio}{%
4909   \eqld@tagmargin@ratio@\dimexpr#1pt\relax}
4910 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
4911   \def\eqld@tagmargin@threshold{#1}}
4912 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
4913 \eqld@define@key\eqld@keycat{mintagwidth}{%
4914   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
4915 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin@{#1}}
4916 \eqld@define@key\eqld@keycat{tagsnap}{%
4917   \eqld@decide@select{#3}{#2}{#1}{%
4918     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
4919     {\relax{\def\eqld@tagpos@snap{#1}}}}}%

```

Tag Layout. Configure methods to declare equation tag layout:

```

4920 \def\eqld@keycat{equations,setup}
4921 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
4922   \eqld@tags@taglayout@set{#1}}
4923 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
4924   \eqld@tags@taglayout@set@direct{#1}}
4925 \eqld@define@key\eqld@keycat{tagform}{%
4926   \eqld@tags@tagform@set#1}
4927 \eqld@define@key\eqld@keycat{tagform*}{%
4928   \eqld@tags@tagform@set@direct{#1}}
4929 \eqld@define@key\eqld@keycat{subeqtemplate}{%
4930   \def\eqld@subequations@template####1####2{#1}%
4931   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}

4932 \eqld@define@key{control}{tagbox,taglayout}{%
4933   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
4934 \eqld@define@key{control}{tagbox*,taglayout*}{%
4935   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
4936 \eqld@define@key{control}{tagform}{%
4937   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set#1}}
4938 \eqld@define@key{control}{tagform*}[\####1]{%
4939   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

4940 \def\eql@keycat{equations,setup}
4941 \eql@define@key\eql@keycat{numberline,number,num,numline,n}[all]{%
4942   \eql@decide@select{#3}{#2}{#1}{%
4943     {\eql@decide@false,0,*}{\let\eql@numbering@active\eql@false}},%
4944     {\eql@decide@true,!}{\let\eql@numbering@active\eql@true}},%
4945     {\none,n,-}{\let\eql@numbering@mode\eql@numbering@mode@multi
4946       \let\eql@numbering@active\eql@false}},%
4947     {\single,1}{\let\eql@numbering@mode\eql@numbering@mode@single
4948       \let\eql@numbering@active\eql@true}},%
4949     {\multi,@}{\let\eql@numbering@mode\eql@numbering@mode@multi
4950       \let\eql@numbering@active\eql@true}},%
4951     {\relax{\eql@numbering@set{#1}}}}
4952 \eql@define@key\eql@keycat{nonumber,nn,*}[]{}%
4953   \let\eql@numbering@active\eql@false}
4954 \eql@define@key\eql@keycat{donumber,dn,!}[]{}%
4955   \let\eql@numbering@active\eql@true}
4956 \eql@define@key\eql@keycat{tagsleft,leqno}[]{\let\eql@tagsleft\eql@true}
4957 \eql@define@key\eql@keycat{tagsright,reqno}[]{\let\eql@tagsleft\eql@false}
4958 \eql@define@key\eql@keycat{tags,eqno}{%
4959   \eql@decide@select{#3}{#2}{#1}{%
4960     {\right,r}{\let\eql@tagsleft\eql@false}},%
4961     {\left,l}{\let\eql@tagsleft\eql@true}}}}
4962 \eql@define@key\eql@keycat{evadetag,avoidtag}[true]{%
4963   \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
4964 \eql@define@key\eql@keycat{tagbetween}[true]{%
4965   \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}

```

TODO: describe

```

4966 \eql@define@key{control}{nonumber,nn,*}[]{\global\@eqnswfalse}
4967 \eql@define@key{control}{donumber,dn,!}[]{\global\@eqnswtrue}
4968 \eql@define@key{control}{numberhere}[]{\eql@numberhere}
4969 \eql@define@key{control}{numbernext}[]{\eql@numbernext}

```

Horizontal Layout. Configure horizontal alignment mode and margin for left alignment:

```

4970 \def\eql@keycat{equations,setup}
4971 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}{%
4972   {\center,c}{\let\eql@layoutleft\eql@false}},%
4973   {\left,l}{\let\eql@layoutleft\eql@true}}}}
4974 \eql@define@key\eql@keycat{center}[]{\let\eql@layoutleft\eql@false}
4975 \eql@define@key\eql@keycat{flushleft,left}[]{\let\eql@layoutleft\eql@true}
4976 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
4977 \eql@define@key\eql@keycat{leftmargin*}{%
4978   \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
4979 \eql@define@key\eql@keycat{minleftmargin}{%
4980   \def\eql@layoutleftmarginmin{#1}}
4981 \eql@define@key\eql@keycat{maxleftmargin}{%
4982   \eql@decide@select{#3}{#2}{#1}{%
4983     {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
4984     {\relax{\def\eql@layoutleftmarginmax{#1}}}}}
4985 \def\eql@keycat{equations,equationsbox}
4986 \eql@define@key\eql@keycat{margin}{%
4987   \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
4988 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}

```

```

4989 \eqld@define@key\eqld@keycat{marginright}{\def\eqld@display@marginright{#1}}
4990 \eqld@define@key{equations}{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

Horizontal Spacing and Columns. Configure column spacing and compression threshold:

```

4991 \def\eqld@keycat{equations,setup}
4992 \eqld@define@key\eqld@keycat{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4993   {\max,full,4}{\eqld@alignbadness@inf@bad}},%
4994   {\high,3}{\eqld@alignbadness@54\relax}},%
4995   {\med,medium,2}{\eqld@alignbadness@18\relax}},%
4996   {\low,1}{\eqld@alignbadness@6\relax}},%
4997   {\0,\eqld@decide@false}{\eqld@alignbadness@z@}}}}
4998 \eqld@define@key\eqld@keycat{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
4999   {\max,full,4}{\eqld@tagbadness@inf@bad}},%
5000   {\high,3}{\eqld@tagbadness@54\relax}},%
5001   {\med,medium,2}{\eqld@tagbadness@18\relax}},%
5002   {\low,1}{\eqld@tagbadness@6\relax}},%
5003   {\0,\eqld@decide@false}{\eqld@tagbadness@z@}}}}
5004 \eqld@define@key\eqld@keycat{alignbadness}{\eqld@alignbadness@numexpr#1\relax}
5005 \eqld@define@key\eqld@keycat{tagbadness}{\eqld@tagbadness@numexpr#1\relax}
5006 \eqld@define@key\eqld@keycat{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5007   {\0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
5008   {\relax{\def\eqld@colsepmin@val{#1}}}}}
5009 \eqld@define@key\eqld@keycat{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5010   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5011   {\relax{\def\eqld@colsepmax@val{#1}}}}}
5012 \eqld@define@key\eqld@keycat{fulllength}[true]{%
5013   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

5014 \eqld@define@key\eqld@keycat{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5015   {\0,\eqld@decide@false}{\def\eqld@break@line@sep{0pt}}},%
5016   {\relax{\def\eqld@break@line@sep{#1}}}}}
5017 \eqld@define@key\eqld@keycat{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
5018   {\0,\eqld@decide@false}{\def\eqld@break@line@shortsep{0pt}}},%
5019   {\relax{\def\eqld@break@line@shortsep{#1}}}}}
5020 \eqld@define@key{equationsbox,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5021   {\0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
5022   {\relax{\def\eqld@box@colsep{#1}}}}}
5023 \let\eqld@box@colsep\eqld@break@col@sep
5024 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5025   {\0,\eqld@decide@false}{\def\eqld@break@col@sep{0pt}}},%
5026   {\relax{\def\eqld@break@col@sep{#1}}}}}
5027 \let\eqld@colsepmin@val\eqld@box@colsep
5028 \let\eqld@colsepmax@val\eqld@box@colsep
5029 \let\eqld@box@colsep\eqld@break@col@sep
5030 \eqld@define@key\eqld@keycat{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5031   {\0,\eqld@decide@false}{\def\eqld@break@col@shortsep{0pt}}},%
5032   {\relax{\def\eqld@break@col@shortsep{#1}}}}}

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5033 \def\eqld@keycat{equations,equationsbox,setup}
5034 \eqld@define@key\eqld@keycat{shape}[default]{\eqld@shape@set{#1}}
5035 \eqld@define@key\eqld@keycat{padding,pad}[indent]{%
5036   \eqld@decide@select{#3}{#2}{#1}{%
5037     {\max}{\let\eqld@paddingleft@val\undefined}},%
5038     {\indent}{\def\eqld@paddingleft@val{\eqld@indent@val}}},%

```

```

5039    {{0,\eq@decide@false}}{\def\eq@paddingleft@val{0pt}}},%
5040    {\relax{\def\eq@paddingleft@val{#1}}}}}%
5041    \let\eq@paddingright@val\eq@paddingleft@val}
5042 \eq@define@key\eq@keycat{padleft}[indent]{%
5043   \eq@decide@select{#3}{#2}{#1}{%
5044     {{max}}{\let\eq@paddingleft@val\@undefined}},%
5045     {{indent}}{\def\eq@paddingleft@val{\eq@indent@val}}},%
5046     {{0,\eq@decide@false}}{\def\eq@paddingleft@val{0pt}}},%
5047     {\relax{\def\eq@paddingleft@val{#1}}}}}%
5048 \eq@define@key\eq@keycat{padright}[indent]{%
5049   \eq@decide@select{#3}{#2}{#1}{%
5050     {{max}}{\let\eq@paddingright@val\@undefined}},%
5051     {{indent}}{\def\eq@paddingright@val{\eq@indent@val}}},%
5052     {{0,\eq@decide@false}}{\def\eq@paddingright@val{0pt}}},%
5053     {\relax{\def\eq@paddingright@val{#1}}}}}%
5054 \eq@define@key\eq@keycat{indent}[2em]{%
5055   \def\eq@indent@val{#1}}

```

TODO: describe

```

5056 \eq@define@key{control}{align}[]{%
5057   \eq@decide@select{#3}{#2}{#1}{%
5058     {{l,left}}{\global\eq@append\eq@cell@container{\eq@shape@pos@z}}},%
5059     {{c,center}}{\global\eq@append\eq@cell@container{\eq@shape@pos@one}}},%
5060     {{r,right}}{\global\eq@append\eq@cell@container{\eq@shape@pos@tw}}}}}%
5061 \eq@define@key{control}{shift,shifto}[]{%
5062   \eq@decide@select{#3}{#2}{#1}{%
5063     {{*,indent}}{\eq@shape@alignamount@set{\eq@indent@val}}},%
5064     {{!,outdent}}{\eq@shape@alignamount@set{-\eq@indent@val}}},%
5065     {\relax{\eq@shape@alignamount@set{#1}}}}}%
5066 \eq@define@key{control}{shift*,shiftby}[]{\eq@shape@alignamount@add{#1}}

```

Math Classes at Alignment. Configure math classes at alignment marker:

```

5067 \def\eq@keycat{equations,equationsbox,setup}
5068 \eq@define@key\eq@keycat{classout}{\eq@class@innerleft@set{#1}}
5069 \eq@define@key\eq@keycat{classin}{\eq@class@innerright@set{#1}}
5070 \eq@define@key\eq@keycat{classlead,classin*}{\eq@class@innerlead@set{#1}}
5071 \eq@define@key\eq@keycat{ampeq}[]{\eq@class@ampeq}
5072 \eq@define@key\eq@keycat{eqamp}[]{\eq@class@eqamp}
5073 \eq@define@key\eq@keycat{class}{\eq@decide@select{#3}{#2}{#1}{%
5074   {{ampeq,amprel,eqafter,beforerel}}\eq@class@ampeq},%
5075   {{eqamp,relamp,eqbefore,afterrel}}\eq@class@eqamp}}}

```

Punctuation. Configure punctuation defaults: **TODO:** describe

```

5076 \def\eq@punct@all#1#2#3#4#5\eq@punct@end{%
5077   \def\eq@tmp{#4}\def\eq@tmpa{1}%
5078   \ifx\eq@tmp\eq@tmpa
5079     \ifnum#5=1111\relax
5080       \eq@punct@set\eq@punct@col{#1}%
5081       \eq@punct@set\eq@punct@line{#2}%
5082       \eq@punct@set\eq@punct@main{#3}%
5083     \else\ifnum#5=111\relax
5084       \eq@punct@set\eq@punct@line{#1}%
5085       \eq@punct@set\eq@punct@main{#2}%
5086     \else\ifnum#5=11\relax
5087       \eq@punct@set\eq@punct@main{#1}%

```

```

5088 \else
5089 \let\eql@punct@col\empty
5090 \let\eql@punct@line\empty
5091 \let\eql@punct@main\empty
5092 \fi\fi\fi
5093 \else
5094 \eql@error{Too many arguments to punctall}%
5095 \fi
5096 }

```

TODO: describe

```

5097 \def\eql@keycat{equations,equationsbox,setup}
5098 \eql@define@key\eql@keycat{punctsep}{\,}{\def\eql@punct@sep{#1}}
5099 \eql@define@key\eql@keycat{punct}{.}{\eql@punct@set\eql@punct@main{#1}}
5100 \eql@define@key\eql@keycat{punct*}[]{\let\eql@punct@main\relax}
5101 \eql@define@key\eql@keycat{punctline}{,}{\eql@punct@set\eql@punct@line{#1}}
5102 \eql@define@key\eql@keycat{punctline*}[]{\let\eql@punct@line\relax}
5103 \eql@define@key\eql@keycat{punctcol}{,}{\eql@punct@set\eql@punct@col{#1}}
5104 \eql@define@key\eql@keycat{punctcol*}[]{\let\eql@punct@col\relax}
5105 \eql@define@key\eql@keycat{punctall}{[,.]{\eql@punct@all#111111\eql@punct@end}}

5106 \eql@define@key{control}{punctsep}{\,}{\def\eql@punct@sep{#1}}
5107 \eql@define@key{control}{punct}{.}{\eql@punct@set\eql@punct@block{#1}%
5108 \eql@punct@set\eql@punct@line{#1}\eql@punct@set\eql@punct@col{#1}}
5109 \eql@define@key{control}{punct*}[]{\let\eql@punct@block\relax}
5110 \eql@define@key{control}{punctapply}[]{\eql@punct@apply@block}

```

Frames. **TODO:** describe

```

5111 \eql@define@key{equationsbox}{frame}{\fbox}{%
5112 \def\eql@box@frame{#1}%
5113 \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
5114 \eql@define@key{equationsbox}{wrap}[]{\eql@box@wrap#1}

```

TODO: describe

```

5115 \eql@define@key{control}{framecell}{\fbox}{%
5116 \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}}
5117 \eql@define@key{control}{frametag}{\fbox}{%
5118 \global\eql@append\eql@tags@container{\def\eql@tags@frame@cmd{#1}}}

```

Alternative Content Description. Alternative content description for accessibility or documentation purposes: **TODO:** implement in PDF tagging

```

5119 \eql@define@key{equations,equationsbox}{alt}{\fbox}{%

```

Injectsions.

```

5120 \eql@define@key{control}{inject}{%
5121 \global\eql@append\eql@interline@container{%
5122 \eql@append\eql@display@injectbefore{#1}}}
5123 \eql@define@key{control}{inject*}{%
5124 \global\eql@append\eql@interline@container{%
5125 \eql@append\eql@display@injectafter{#1}}}
5126 \eql@define@key{control}{markline}[]{\eql@markline@inject{#1}}
5127 \eql@define@key{control}{markline*}[]{\eql@markline@inject{push,#1}}
5128 \eql@define@key{control}{qed}[]{\eql@markline@inject{qed,#1}}
5129 \eql@define@key{control}{qed*}[]{\eql@markline@inject{qed,push,#1}}

```

TODO: describe

```
5130 \eqld@define@key{markline}{pos}{%
5131   \eqld@decide@select{#3}{#2}{#1}{%
5132     {{below,push}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5133     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5134     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%
5135 \eqld@define@key{markline}{below,push}[]{%
5136   \let\eqld@markline@pos\eqld@markline@pos@below}
5137 \eqld@define@key{markline}{baseline}[]{%
5138   \let\eqld@markline@pos\eqld@markline@pos@baseline}
5139 \eqld@define@key{markline}{bottom}[]{%
5140   \let\eqld@markline@pos\eqld@markline@pos@bottom}
5141 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
5142 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}
5143 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
5144 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
5145 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
5146 \eqld@define@key{setup}{markpos}{%
5147   \eqld@decide@select{#3}{#2}{#1}{%
5148     {{below}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5149     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5150     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}%

```

Global Switches. Set global switches:

```
5151 \let\eqld@multi@linesfallback\eqld@false
5152 \let\eqld@scan@par\eqld@false
5153 \let\eqld@single@cr@mode\eqld@false
5154 \let\eqld@ampproof@active\eqld@false

5155 \eqld@define@key{equations,setup}{linesfallback}[true]{%
5156   \eqld@decide@select{#3}{#2}{#1}{%
5157     {\eqld@decide@false{\let\eqld@multi@linesfallback\eqld@false}},%
5158     {{reuse,lean}}{\let\eqld@multi@linesfallback\z@}},%
5159     {{measure,full,\eqld@decide@true}}{\let\eqld@multi@linesfallback\eqld@true}}}%
5160 \eqld@define@key{setup}{ampproof}[true]{%
5161   \eqld@decide@bool{#3}{#2}{#1}\eqld@ampproof@active}
5162 \eqld@define@key{equations,setup}{equationcr}{%
5163   \eqld@decide@select{#3}{#2}{#1}{%
5164     {\eqld@decide@false{\let\eqld@single@cr@mode\eqld@false}},%
5165     {{\eqld@decide@true,break}}{\let\eqld@single@cr@mode\eqld@break@line}},%
5166     {{error,verbose}}{\let\eqld@single@cr@mode\eqld@single@cr@error}}}%
5167 \eqld@define@key{setup}{modifierwarning}[true]{%
5168   \eqld@decide@select{#3}{#2}{#1}{%
5169     {\eqld@decide@false{\let\eqld@parseopt@warn\@empty}},%
5170     {\eqld@decide@true{\let\eqld@parseopt@warn\eqld@warn@parseopt}},%
5171     {{verbose,+}}{\let\eqld@parseopt@warn\eqld@warn@parseopt@verbose}}}%
5172 \let\eqld@parseopt@warn\eqld@warn@parseopt
5173 \eqld@define@key{equations,setup}{rescan}[true]{%
5174   \eqld@decide@if{#3}{#2}{#1}%
5175   {\let\eqld@scan@body\eqld@scan@body@rescan}%
5176   {\let\eqld@scan@body\eqld@scan@body@dump}}
5177 \eqld@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5178   \eqld@decide@bool{#3}{#2}{#1}\eqld@scan@par}
5179 \eqld@define@key{setup}{defaults}{%
5180   \eqld@decide@select{#3}{#2}{#1}{%
5181     {{classic}}{\eqld@defaults@classic}},%
5182     {{eqnlines}}{\eqld@defaults@eqnlines}}}%

```

Package Options. Declare choices available at loading of package only: **TODO:** adjust

```

5183 \let\eql@provide@opt@env\tw@
5184 \let\eql@provide@opt@amsmathends\eql@true
5185 \let\eql@provide@opt@backup\eql@false
5186 \let\eql@provide@opt@ang\eql@true
5187 \let\eql@provide@opt@eqref\eql@true

5188 \eql@define@key{setup}{amsmathends}[true]{%
5189   \eql@error@packageoption{#2}%
5190   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathends}
5191 \eql@define@key{setup}{backup}[true]{%
5192   \eql@error@packageoption{#2}%
5193   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5194 \eql@define@key{setup}{env}[equation]{%
5195   \eql@error@packageoption{#2}%
5196   \eql@decide@select{#3}{#2}{#1}{%
5197     {none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5198     {{equation,latex}{\let\eql@provide@opt@env\@ne}},%
5199     {\amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}%
5200 \eql@define@key{setup}{ang}[true]{%
5201   \eql@error@packageoption{#2}%
5202   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5203 \eql@define@key{setup}{eqref}[true]{%
5204   \eql@error@packageoption{#2}%
5205   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}

```

Shortcut Options. **TODO:** describe

```

5206 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5207 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5208 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5209 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5210 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5211 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5212 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5213 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5214 \def\eql@parseopt@punctdot#1{\eqnaddopt{punct={.}}\eql@parseopt@peek}
5215 \def\eql@parseopt@punctcomma#1{\eqnaddopt{punct={,}}\eql@parseopt@peek}
5216 \def\eql@parseopt@punctoff#1{\eqnaddopt{punct={}}\eql@parseopt@peek}
5217 \def\eql@parseopt@punctall#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}

```

16.3 Parameter Presets

The package offers two parameter presets which lead to somewhat different layout. Instead of setting the internal parameters directly, we expose them as public settings so that they are easier to read and such that individual settings can be used to compose own layouts.

eql@defaults@classic The preset `classic` aims to reproduce the $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and `amsmath` layout closely. These presets mostly use fixed dimensions:

```

5218 \def\eql@defaults@classic{%
5219   \eqnlineset{numberline=all}%
5220   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5221   \eqnlineset{maxcolsep=off}%
5222   \eqnlineset{spread={\jot}}%
5223   \eqnlineset{tagmargin}%

```

```

5224 \eqnlineset{tagmarginratio=1}%
5225 \eqnlineset{tagmarginthreshold=0.5}%
5226 \eqnlineset{leftmargin={\leftmargini}}%
5227 \eqnlineset{padding=max}%
5228 \eqnlineset{evadetag=off}%
5229 \eqnlineset{displayheight=off}%
5230 \eqnlineset{displaydepth=off}%
5231 \eqnlineset{shortmode=belowsingle}%
5232 \eqnlineset{abovecontmode=short}%
5233 \eqnlineset{belowcontmode=short}%
5234 \eqnlineset{aboveparmode=long}%
5235 \eqnlineset{belowparmode=long}%
5236 \eqnlineset{abovetopmode=long}%
5237 \eqnlineset{belowtopmode=long}%
5238 \eqnlineset{abovelongskip={\abovedisplayskip}}%
5239 \eqnlineset{belowlongskip={\belowdisplayskip}}%
5240 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5241 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5242 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5243 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5244 \eqnlineset{abovecontskip=0pt}%
5245 \eqnlineset{belowcontskip=0pt}%
5246 \eqnlineset{aboveparskip=0pt}%
5247 \eqnlineset{belowparskip=0pt}%
5248 \eqnlineset{abovetopskip=0pt}%
5249 \eqnlineset{belowtopskip=0pt}%
5250 \eqnlineset{abovetagskip=0pt}%
5251 \eqnlineset{belowtagskip=0pt}%
5252 \eqnlineset{equationcr=off}%
5253 \eqnlineset{linesfallback=false}%
5254 }

```

values based on 10pt vs 12pt

`\eql@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the `classic` spacing for a 12 pt computer modern font such that 10 pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

5255 \def\eql@defaults@eqnlines{%
5256   \eqnlineset{numberline=all}%
5257   \eqnlineset{mintagsep=.5em}%
5258   \eqnlineset{maxcolsep=2em}%
5259   \eqnlineset{spread={0.2\normalbaselineskip}}%
5260   \eqnlineset{tagmargin}%
5261   \eqnlineset{tagmarginratio=.334}%
5262   \eqnlineset{tagmarginthreshold=0.5}%
5263   \eqnlineset{leftmargin={\leftmargini}}%
5264   \eqnlineset{padding=0pt}%
5265   \eqnlineset{evadetag}%
5266   \eqnlineset{displayheight=strut}%
5267   \eqnlineset{displaydepth=strut}%
5268   \eqnlineset{shortmode=above}%
5269   \eqnlineset{abovecontmode=noskip}%
5270   \eqnlineset{belowcontmode=long}%
5271   \eqnlineset{aboveparmode=long}%
5272   \eqnlineset{belowparmode=long}%

```

```

5273 \eqnlineset{abovetopmode=noskip}%
5274 \eqnlineset{belowtopmode=long}%
5275 \eqnlineset{longskip={0.75\normalbaselineskip
5276   plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5277 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5278   plus 0.25\normalbaselineskip}}%
5279 \eqnlineset{belowshortskip={0.0\normalbaselineskip
5280   plus 0.25\normalbaselineskip}}%
5281 \eqnlineset{medskip={0.4\normalbaselineskip
5282   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5283 \eqnlineset{abovecontskip=0pt}%
5284 \eqnlineset{belowcontskip=0pt}%
5285 \eqnlineset{aboveparskip=0pt}%
5286 \eqnlineset{belowparskip=0pt}%
5287 \eqnlineset{abovetopskip=0pt}%
5288 \eqnlineset{belowtopskip=0pt}%
5289 \eqnlineset{abovetagskip={0.25\normalbaselineskip
5290   minus 0.25\normalbaselineskip}}%
5291 \eqnlineset{belowtagskip={0.25\normalbaselineskip
5292   minus 0.25\normalbaselineskip}}%
5293 \eqnlineset{equationcr=break}%
5294 \eqnlineset{linesfallback=true}%
5295 }

```

16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\provide@moveenv
\provide@undefinecmd
\provide@undefineenv
5296 \def\eql@provide@movecmd#1#2{%
5297   \eql@letcs{#1\expandafter}\csname#2\endcsname
5298 }
5299 \def\eql@provide@moveenv#1#2{%
5300   \eql@provide@movecmd{#1}{#2}%
5301   \ifcsname end#2\endcsname
5302     \eql@provide@movecmd{end#1}{end#2}%
5303   \fi
5304 }
5305 \def\eql@provide@undefinecmd#1{%
5306   \eql@letcs{#1}\@undefined
5307 }
5308 \def\eql@provide@undefineenv#1{%
5309   \eql@provide@undefinecmd{#1}%
5310   \eql@provide@undefinecmd{end#1}%
5311 }

```

Fix Endings for `amsmath` Environments. The `amsmath` derived environments forward their ending routines directly to the ending routines for the main environments `gather`, `multline`, `align`, `aligned`. This causes a problem when the main environments are replaced but the derived ones are still used. We fix the potential problem by copying the ending routines of the main environments to the ending routines of the derived environments.

`\eql@amsmath@endfix` Check whether the original forwarding of an ending routine is still in place (other packages or future updates to `amsmath` might change the behaviour). If so, copy the ending routine into place:

```
5312 \def\eql@amsmath@endfix#1#2{%
5313   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5314   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5315   \eql@provide@movecmd{end#1}{end#2}%
5316   \fi
5317 }
```

`\eql@amsmath@fixends` Perform the replacement for all `amsmath` environments whenever `amsmath` is loaded:

```
5318 \def\eql@amsmath@fixends{%
5319   \eql@amsmath@after{%
5320     \eql@amsmath@endfix{flalign}{align}%
5321     \eql@amsmath@endfix{alignat}{align}%
5322     \eql@amsmath@endfix{xalignat}{align}%
5323     \eql@amsmath@endfix{xxalignat}{align}%
5324     \eql@amsmath@endfix{gather*}{gather}%
5325     \eql@amsmath@endfix{multline*}{multline}%
5326     \eql@amsmath@endfix{align*}{align}%
5327     \eql@amsmath@endfix{flalign*}{align}%
5328     \eql@amsmath@endfix{alignat*}{align}%
5329     \eql@amsmath@endfix{xalignat*}{align}%
5330     \eql@amsmath@endfix{gathered}{aligned}%
5331     \eql@amsmath@endfix{alignedat}{aligned}%
5332   }
5333 }
```

Backup `amsmath` Environments. We can backup all `amsmath` environments *env* to `amsenv` so that they can be used in parallel if needed.

`\provide@backup@amsenv` Copy an `amsmath` environment *env* to `amsenv` whenever `amsmath` is loaded: **TODO:** describe

```
5334 \def\eql@provide@backup@amsenv#1{%
5335   \eql@amsmath@after{%
5336     \eql@provide@moveenv{ams#1}{#1}%
5337     \eql@tagging@register@luamml{ams#1}%
5338     \eql@markline@amsthm@move{ams#1}{#1}%
5339   }%
5340 }
```

`\provide@backup@amssub` **TODO:** describe

```
5341 \def\eql@provide@backup@amssub#1{%
5342   \eql@amsmath@after{%
5343     \eql@provide@moveenv{ams#1}{#1}%
5344   }%
5345 }
```

`\provide@backup@eqref` Copy an `eqref` to `amseqref` whenever `amsmath` is loaded:

```
5346 \def\eql@provide@backup@eqref{%
5347   \eql@amsmath@after{%
5348     \eql@provide@movecmd{amseqref}{eqref}%
5349   }%
5350 }
```

vide@backup@multlined The environment `multlined` is supplied by `mathtools`. We copy it to `amsmultlined` anyway, but whenever `mathtools` is loaded:

```
5351 \def\eql@provide@backup@multlined{%
5352   \AddToHook{package/mathtools/after}{%
5353     \eql@provide@moveenv{amsmultlined}{multlined}}%
5354 }
```

vide@backup@equation The L^AT_EX environment `equation` is overwritten by several packages to implement their adjustments. Here we cater for adjustments through `amsmath`, `hyperref` and the PDF tagging mechanism. Copy `equation` and `equation*` whenever `amsmath` is loaded. Whenever `hyperref` is loaded, and `amsmath` is not yet present, backup the original L^AT_EX and `hyperref` versions of `equation`. If neither `hyperref` nor `amsmath` are present, just backup the original L^AT_EX `equation`. The PDF tagging mechanism registers `equation` upon `\begin{document}`. We thus need to register all copies of `equation` on our own, so that they can be used with their new names:

```
5355 \def\eql@provide@backup@equation{%
5356   \eql@amsmath@after{%
5357     \eql@provide@moveenv{amsequeation}{equation}%
5358     \eql@provide@moveenv{amsequeation*}{equation*}%
5359     \eql@tagging@register@env{amsequeation}%
5360     \eql@tagging@register@env{amsequeation*}%
5361     \eql@tagging@register@luamml{amsequeation}%
5362     \eql@tagging@register@luamml{amsequeation*}%
5363     \eql@markline@amsthm@move{amsequeation}{equation}%
5364     \eql@markline@amsthm@move{amsequeation*}{equation*}%
5365   }%
5366   \AddToHook{package/hyperref/after}{%
5367     \@ifpackageloaded{amsmath}{}%
5368     \eql@provide@moveenv{hyperrefequation}{equation}%
5369     \eql@tagging@register@env{hyperrefequation}%
5370     \eql@tagging@register@luamml{hyperrefequation}%
5371     \eql@markline@amsthm@move{hyperequation}{equation}%
5372   }%
5373 }%
5374 \@ifpackageloaded{amsmath}{}%
5375   \@ifpackageloaded{hyperref}{%
5376     \let\latexequation\H@equation
5377     \let\endlatexequation\H@endequation
5378   }{\eql@provide@moveenv{latexequation}{equation}}%
5379   \eql@tagging@register@env{latexequation}%
5380   \eql@tagging@register@luamml{latexequation}%
5381   \eql@markline@amsthm@move{latexequation}{equation}%
5382 }%
5383 }
```

e@backup@displaymath **TODO:** describe

```
5384 \def\eql@provide@backup@displaymath{%
5385   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5386   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
5387 }
```

@backup@subequations The `amsmath` `subequations` environment is adjusted by `hyperref` through an environment hook, but this hook gets applied only later at `\begin{document}`. Hence, we need to supply the hook routine to the new routine ourselves:

```

5388 \def\eql@provide@backup@subequations{%
5389   \eql@amsmath@after{%
5390     \eql@provide@moveenv{amssubequations}{subequations}%
5391   }%
5392   \AddToHook{package/hyperref/after}{%
5393     \AddToHook{cmd/amssubequations/before}{%
5394       {%
5395         \stepcounter{equation}%
5396         \protected@edef\theHparentequation{\theHequation}%
5397         \addtocounter{equation}{-1}%
5398       }%
5399       \AddToHook{cmd/amssubequations/after}{%
5400         {%
5401           \def\theHequation{\theHparentequation\alph{equation}}%
5402           \ignorespaces
5403         }%
5404       }%
5405 }

```

`\eql@provide@backup` Backup all amsmath environments:

```

5406 \def\eql@provide@backup{%
5407   \eql@provide@backup@eqref
5408   \eql@provide@backup@equation
5409   \eql@provide@backup@displaymath
5410   \eql@provide@backup@amsenv{gather}%
5411   \eql@provide@backup@amsenv{multline}%
5412   \eql@provide@backup@amsenv{align}%
5413   \eql@provide@backup@amsenv{flalign}%
5414   \eql@provide@backup@amsenv{alignat}%
5415   \eql@provide@backup@amsenv{xalignat}%
5416   \eql@provide@backup@amsenv{xxalignat}%
5417   \eql@provide@backup@amsenv{gather*}%
5418   \eql@provide@backup@amsenv{multline*}%
5419   \eql@provide@backup@amsenv{align*}%
5420   \eql@provide@backup@amsenv{flalign*}%
5421   \eql@provide@backup@amsenv{alignat*}%
5422   \eql@provide@backup@amsenv{xalignat*}%
5423   \eql@provide@backup@amssub{aligned}%
5424   \eql@provide@backup@amssub{alignedat}%
5425   \eql@provide@backup@amssub{gathered}%
5426   \eql@provide@backup@multlined
5427   \eql@provide@backup@subequations
5428 }

```

Replacement amsmath Environments. **TODO:** describe

```

5429 \def\eql@alignat@gobblecol#1{%
5430   \eql@ifnextchar@tight\bgroup{\@firstoftwo{#1}}{#1}}

```

`eql@gathered` (*env.*) Define replacement versions for boxed environments `gathered`, `multlined` and `aligned`
`eql@multlined` (*env.*) which forward to `equationsbox` with specific presets:

`eql@aligned` (*env.*)

```

5431 \newenvironment{eql@gathered}
5432   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5433 \newenvironment{eql@multlined}
5434   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5435 \newenvironment{eql@aligned}

```

```

5436 {\eqnaddopt{columns}\equationsbox}\endequationsbox}
5437 \newenvironment{eql@alignedat}
5438 {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5439 {\endequationsbox}

```

eql@equation (*env.*) Define replacement versions for display environments `equation`, `gather`, `multline`,
eql@gather (*env.*) `aligned` and derivatives which forward to `equations` with specific presets: **TODO:**
eql@multline (*env.*) `amsmath` at variants would need predefined columns for full operation
eql@align (*env.*)

```

5440 \newenvironment{eql@equation}
5441 {\eqnaddopt{equation}\equations}\endequations}
5442 \newenvironment{eql@displaymath}
5443 {\eqnaddopt{equation,nonumber}\equations}\endequations}
5444 \newenvironment{eql@gather}
5445 {\eqnaddopt{lines}\equations}\endequations}
5446 \newenvironment{eql@multline}
5447 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5448 {\endequations}
5449 \newenvironment{eql@align}
5450 {\eqnaddopt{columns}\equations}\endequations}
5451 \newenvironment{eql@flalign}
5452 {\eqnaddopt{fulllength}\eql@align}\endequations}
5453 \newenvironment{eql@alignat}
5454 {\eqnaddopt{colsep=off}\eql@xalignat}\endequations}
5455 \newenvironment{eql@xalignat}
5456 {\eql@alignat@gobblecol\eql@align}\endequations}
5457 \newenvironment{eql@xxalignat}
5458 {\eqnaddopt{fulllength}\eql@xalignat}\endequations}
5459 \newenvironment{eql@equation*}
5460 {\eqnaddopt{nonumber}\eql@equation}\endequations}
5461 \newenvironment{eql@gather*}
5462 {\eqnaddopt{nonumber}\eql@gather}\endequations}
5463 \newenvironment{eql@multline*}
5464 {\eqnaddopt{nonumber}\eql@multline}\endequations}
5465 \newenvironment{eql@align*}
5466 {\eqnaddopt{nonumber}\eql@align}\endequations}
5467 \newenvironment{eql@flalign*}
5468 {\eqnaddopt{nonumber}\eql@flalign}\endequations}
5469 \newenvironment{eql@alignat*}
5470 {\eqnaddopt{nonumber}\eql@alignat}\endequations}
5471 \newenvironment{eql@xalignat*}
5472 {\eqnaddopt{nonumber}\eql@xalignat}\endequations}

```

Install Additional Environments. The additional environments need to be installed at their intended names which can be adjusted by the user.

eql@provide@onlyonce Process arguments for providing a specific environment. #1 describes the environment using the `amsmath` name. #2 specifies the desired target name. If #2 is empty or equals #1, overwrite the `amsmath` environment in place making sure that the replacement is robust against loading `amsmath` before or after. If #2 equals ‘*’, just overwrite the `amsmath` environment in place immediately (e.g. within a block in the document body):

```

5473 \def\eql@provide@onlyonce#1#2{%
5474   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5475   \ifx\eql@tmp\eql@tmpa
5476     \let\eql@tmp@empty
5477   \fi

```

```

5478 \ifx\eql@tmp\@empty
5479 \let\eql@tmp\@undefined
5480 \ifx\@nodocument\relax
5481 \def\eql@tmp{#1}%
5482 \fi
5483 \ifcsname eql@provided@#1\endcsname
5484 \def\eql@tmp{#1}%
5485 \fi
5486 \eql@letcs{eql@provided@#1}\eql@true
5487 \else
5488 \def\eql@tmpa{*}%
5489 \ifx\eql@tmp\eql@tmpa
5490 \def\eql@tmp{#1}%
5491 \fi
5492 \fi
5493 }

```

\eql@provide@eqref Provide `\eqref` as the macro #1. We have to check whether #1 is empty or equals `\eqref` or takes the value ‘`*`’. If not, we should strip the backslash for further processing. Copy the macro into place, and copy again when `amsmath` or `mathtools` are loaded. Remove definition before `amsmath` is loaded in the future to avoid a potential error:

```

5494 \def\eql@provide@eqref#1{%
5495 \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
5496 \ifx\eql@tmp\eql@tmpa
5497 \let\eql@tmp\@empty
5498 \fi
5499 \ifx\eql@tmp\@empty
5500 \eql@provide@onlyonce{eqref}{}%
5501 \else
5502 \def\eql@tmpa{*}%
5503 \ifx\eql@tmp\eql@tmpa
5504 \def\eql@tmp{eqref}%
5505 \else
5506 \edef\eql@tmp{\expandafter\@gobble\string#1}%
5507 \fi
5508 \fi
5509 \ifdefined\eql@tmp
5510 \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
5511 \else
5512 \eql@amsmath@after{%
5513 \eql@provide@movecmd{eqref}{eql@eqref}%
5514 }%
5515 \AddToHook{package/mathtools/after}{%
5516 \eql@provide@movecmd{eqref}{eql@eqref}%
5517 }%
5518 \eql@provide@movecmd{eqref}{eql@eqref}%
5519 \eql@amsmath@undefine\eqref
5520 \fi
5521 }

```

\eql@provide@amsenv Provide one of the `amsmath` environments. Copy into place, and copy again when `amsmath` is loaded. Remove definition before `amsmath` is loaded in the future to avoid an error:

```

5522 \def\eql@provide@amsenv#1#2{%
5523 \eql@provide@onlyonce{#1}{#2}%
5524 \ifdefined\eql@tmp
5525 \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5526 \eql@tagging@register@luamml{\eql@tmp}%

```

```

5527 \eq\markline@amsthm@register{\eq\@tmp}%
5528 \else
5529 \eq\amsmath@after{%
5530 \eq\provide@moveenv{#1}{eq\@#1}%
5531 \eq\markline@amsthm@register{#1}%
5532 }%
5533 \AddToHook{package/mathtools/after}{%
5534 \eq\provide@moveenv{#1}{eq\@#1}%
5535 \eq\markline@amsthm@register{#1}%
5536 }%
5537 \eq\provide@moveenv{#1}{eq\@#1}%
5538 \eq\markline@amsthm@register{#1}%
5539 \eq\amsmath@before{\eq\provide@undefineenv{#1}}%
5540 \fi
5541 }

```

`\eq\provide@amssub` Provide one of the amsmath subequation structures. Copy into place, and copy again when amsmath is loaded. Remove definition before amsmath is loaded in the future to avoid an error:

```

5542 \def\eq\provide@amssub#1#2{%
5543 \eq\provide@onlyonce{#1}{#2}%
5544 \ifdefined\eq\@tmp
5545 \eq\provide@moveenv{\eq\@tmp}{eq\@#1}%
5546 \else
5547 \eq\amsmath@after{%
5548 \eq\provide@moveenv{#1}{eq\@#1}}%
5549 \AddToHook{package/mathtools/after}{%
5550 \eq\provide@moveenv{#1}{eq\@#1}}%
5551 \eq\provide@moveenv{#1}{eq\@#1}%
5552 \eq\amsmath@before{\eq\provide@undefineenv{#1}}%
5553 \fi
5554 }

```

`\eq\provide@multlined` Provide mathtools environment multlined. Copy into place, and copy again when mathtools is loaded. Remove definition before mathtools is loaded in the future to avoid an error:

```

5555 \def\eq\provide@multlined#1{%
5556 \eq\provide@onlyonce{multlined}{#1}%
5557 \ifdefined\eq\@tmp
5558 \eq\provide@moveenv{\eq\@tmp}{eq\@multlined}%
5559 \else
5560 \AddToHook{package/mathtools/after}{%
5561 \eq\provide@moveenv{multlined}{eq\@multlined}}%
5562 \eq\provide@moveenv{multlined}{eq\@multlined}%
5563 \@ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5564 \eq\provide@undefineenv{multlined}}}%
5565 \fi
5566 }

```

`\eq\provide@equation` Provide the environment equation. Copy into place, and copy again when amsmath or hyperref are loaded. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

5567 \def\eq\provide@equation#1{%
5568 \eq\provide@onlyonce{equation}{#1}%
5569 \ifdefined\eq\@tmp

```

```

5570 \eql@provide@moveenv{\eql@tmp}{\eql@equation}%
5571 \eql@tagging@register@luamml{\eql@tmp}%
5572 \eql@markline@amsthm@register{\eql@tmp}%
5573 \else
5574 \eql@amsmath@after{%
5575 \eql@provide@moveenv{equation}{\eql@equation}%
5576 \eql@markline@amsthm@register{equation}%
5577 }%
5578 \AddToHook{package/hyperref/after}{%
5579 \ifpackageloaded{amsmath}{}%
5580 \eql@provide@moveenv{equation}{\eql@equation}%
5581 \eql@markline@amsthm@register{equation}%
5582 }%
5583 }%
5584 \eql@provide@moveenv{equation}{\eql@equation}%
5585 \eql@markline@amsthm@register{equation}%
5586 \ifdefined\eql@tagging@on
5587 \AddToHook{begindocument/end}{%
5588 \eql@provide@moveenv{equation}{\eql@equation}%
5589 \eql@markline@amsthm@register{equation}%
5590 }%
5591 \fi
5592 \fi
5593 }

```

`\provide@equationstar` Provide the environment `equation*`. Copy into place, and copy again when `amsmath` or `hyperref` are loaded. Remove definition of `equation*` before `amsmath` is loaded in the future to avoid an error. When PDF tagging is active, the environment is modified at `\begin{document}` in an undesirable fashion, so copy the definition again:

```

5594 \def\eql@provide@equationstar#1{%
5595 \eql@provide@onlyonce{equation*}{#1}%
5596 \ifdefined\eql@tmp
5597 \eql@provide@moveenv{\eql@tmp}{\eql@equation*}%
5598 \eql@tagging@register@luamml{\eql@tmp}%
5599 \eql@markline@amsthm@register{\eql@tmp}%
5600 \else
5601 \eql@amsmath@after{%
5602 \eql@provide@moveenv{equation*}{\eql@equation*}%
5603 \eql@markline@amsthm@register{equation*}%
5604 }%
5605 \eql@provide@moveenv{equation*}{\eql@equation*}%
5606 \eql@markline@amsthm@register{equation*}%
5607 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5608 \ifdefined\eql@tagging@on
5609 \AddToHook{begindocument/end}{%
5610 \eql@provide@moveenv{equation*}{\eql@equation*}%
5611 \eql@markline@amsthm@register{equation*}%
5612 }%
5613 \fi
5614 \fi
5615 }

```

`\@provide@displaymath` **TODO:** describe

```

5616 \def\eql@provide@displaymath#1{%
5617 \eql@provide@onlyonce{displaymath}{#1}%
5618 \ifdefined\eql@tmp
5619 \eql@provide@moveenv{\eql@tmp}{\eql@displaymath}%

```

```

5620 \eq\markline@amsthm@register{\eq\tmp}%
5621 \eq\tagging@register@luamml{\eq\tmp}%
5622 \else
5623 \eq\provide@moveenv{displaymath}{eq\displaymath}%
5624 \eq\markline@amsthm@register{displaymath}%
5625 \ifdefined\eq\tagging@on
5626 \AddToHook{begindocument/end}{%
5627 \eq\provide@moveenv{displaymath}{eq\displaymath}}%
5628 \fi
5629 \fi
5630 }

```

provide@subequations Provide the amsmath environment `subequations`. Copy into place, and copy again when `amsmath` is loaded. `hyperref` adds a hook to the command which messes up the parsing of optional arguments (even if the hook is emptied). The hook placement happens at `\begin{document}`, so we copy the environment again afterwards. We also remove the hook (after adding an empty hook to avoid errors). Remove definition before `amsmath` is loaded in the future to avoid an error:

```

5631 \def\eq\provide@subequations#1{%
5632 \eq\provide@onlyonce{subequations}{#1}%
5633 \ifdefined\eq\tmp
5634 \eq\provide@moveenv{\eq\tmp}{eq\subequations}%
5635 \else
5636 \eq\amsmath@after{%
5637 \eq\provide@moveenv{subequations}{eq\subequations}%
5638 }%
5639 \AddToHook{package/hyperref/after}{%
5640 \AddToHook{cmd/subequations/before}[hyperref]{}%
5641 \AddToHook{cmd/subequations/after}[hyperref]{}%
5642 \RemoveFromHook{cmd/subequations/before}[hyperref]%
5643 \RemoveFromHook{cmd/subequations/after}[hyperref]%
5644 \AddToHook{begindocument/end}{%
5645 \eq\provide@moveenv{subequations}{eq\subequations}}%
5646 }%
5647 \eq\provide@moveenv{subequations}{eq\subequations}%
5648 \eq\amsmath@before{\eq\provide@undefineenv{subequations}}%
5649 \fi
5650 }

```

\eq\provide@sqr Provide the symbolic environment `\[...\]`. Copy into place, and copy again when `amsmath` is loaded. If PDF tagging is active, some undesired modifications happen at `\begin{document}`, so copy again afterwards:

```

5651 \def\eq\provide@sqr{%
5652 \let\[ \eq\sqr@open
5653 \let\] \eq\sqr@close
5654 \eq\amsmath@after{%
5655 \let\[ \eq\sqr@open
5656 \let\] \eq\sqr@close
5657 }%
5658 \ifdefined\eq\tagging@on
5659 \AddToHook{begindocument/end}{%
5660 \let\[ \eq\sqr@open
5661 \let\] \eq\sqr@close
5662 }%
5663 \fi
5664 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```
5665 \def\eql@provide@ang{%
5666   \let\<\eql@ang@open
5667   \let\>\eql@ang@close
5668 }
```

Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```
5669 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
5670 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
5671 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
5672 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
5673 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
5674 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
5675 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
5676 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
5677 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
5678 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
5679 \eql@define@key{provide}{aligned}[]{\eql@provide@amssub{aligned}{#1}}
5680 \eql@define@key{provide}{alignedat}[]{\eql@provide@amssub{alignedat}{#1}}
5681 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
5682 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
5683 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
5684 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}
5685 \eql@define@key{provide}{alignat*}[]{\eql@provide@amsenv{alignat*}{#1}}
5686 \eql@define@key{provide}{xalignat*}[]{\eql@provide@amsenv{xalignat*}{#1}}
5687 \eql@define@key{provide}{gathered}[]{\eql@provide@amssub{gathered}{#1}}
5688 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
5689 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
5690 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
5691 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
5692 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
5693 \eql@define@key{provide}{tagform}[]{%
5694   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}%
5695 \eql@define@key{provide}{maketag}[]{%
5696   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}
```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```
5697 \newcommand{\eqnlinesprovide}[1]{%
5698 (dev)\eql@dev@start\eqnlinesprovide
5699   \eql@setkeys{provide}{#1}%
5700   \ignorespaces
5701 }
```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
5702 \let\eql@error@packageoption\gobble
```

Declare math layout options `leqno` and `fleqn` for common L^AT_EX classes:

```
5703 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
5704 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
5705 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
5706 \eql@defaults@eqnlines
5707 \eql@mode@columns
5708 \eql@mode@aligned
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the T_EX conditional parsing mechanism:

```
5709 \ifdefined\tagsleft@true\else
5710   \expandafter\newif\csname iftagsleft@\endcsname
5711 \fi
5712 \ifdefined\@fleqntrue\else
5713   \expandafter\newif\csname if@fleqn\endcsname
5714 \fi
```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
5715 \eql@amsmath@after{%
5716   \ifnum\eql@provide@opt@env=\tw@
5717     \iftagsleft@
5718       \eqnlineset{tags=left}%
5719     \else
5720       \eqnlineset{tags=right}%
5721     \fi
5722   \if@fleqn
5723     \eqnlineset{layout=left}%
5724   \else
5725     \eqnlineset{layout=center}%
5726   \fi
5727 \fi
5728 }
```

Process package options:

```
5729 \ProcessOptions
```

`@error@packageoption` Enable error message for exclusive package options:

```
5730 \def\eql@error@packageoption#1{%
5731   \eql@error{may only use '#1' as a package option}%
5732 }
```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```
5733 \ifdefined\eql@provide@opt@amsmathends\eql@amsmath@fixends\fi
```

Backup all `amsmath` environments that may be overwritten to `ams...`. This will happen before any replacements:

```
5734 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi
```

Provide native L^AT_EX environment `equation` and symbolic shortcut `\[...\]` if desired:

```

5735 \ifnum\eq1@provide@opt@env>\z@
5736   \eqnlinesprovide{equation,equation*,sqr,displaymath}
5737 \fi

```

Provide amsmath equation environments if desired:

```

5738 \ifnum\eq1@provide@opt@env=\tw@
5739   \eqnlinesprovide{%
5740     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
5741     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
5742     multlined,gathered,aligned,alignedat,%
5743     subequations}
5744 \fi

```

Provide symbolic shortcut \<...> if desired:

```

5745 \ifdefined\eq1@provide@opt@ang\eqnlinesprovide{ang}\fi

```

Provide equation reference \eqref if desired:

```

5746 \ifdefined\eq1@provide@opt@eqref\eqnlinesprovide{eqref}\fi

```