

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`

2026/01/28, v0.14

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

Abstract

`eqnlines` is a L^AT_EX 2 ϵ package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of L^AT_EX 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	Debugging 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	18
4	Features	18
4.1	Punctuation	18
4.2	Math Classes at Alignment	21

4.3	Framed Cells	21
4.4	Single-Line Composition	22
4.5	Alternative Content Description	23
5	Equation Numbering	23
5.1	Supporting Definitions	23
5.2	Schemes	24
5.3	Interface	28
5.4	Integration	33
5.5	Positioning	35
5.6	Component Display	39
5.7	Tag Composition	42
6	Subequation Numbering	45
6.1	Definitions	45
6.2	Environment	46
6.3	Subequation Scheme	47
7	Display Equations Support	48
7.1	Display Breaks	48
7.2	Explicit Vertical Space	49
7.3	Default Vertical Spacing	51
7.4	Entry and Exit	52
7.5	Stack	57
8	Multi-Line Support	58
8.1	Measure Support	58
8.2	Line Breaks	60
8.3	Intertext	61
8.4	Line Marks	63
9	Column Placement	65
9.1	Supporting Definitions	65
9.2	Shape Schemes	65
9.3	Width Data	69
9.4	Best Line Selection	72
9.5	Tag Margin	73
9.6	Single Column	74
9.7	Multiple Columns	75
10	Single Column Arrangement	81
10.1	Supporting Definitions	81
10.2	Arrangement Methods	81
10.3	Central Alignment	83
10.4	Left Alignment	84
10.5	Right Alignment	85
11	Equations Box Environment	87
11.1	Line Breaks	87
11.2	Stacked Mode	88
11.3	Aligned Mode	89
11.4	Cases Mode	91
11.5	Main	92
11.6	Environment	94

12 Single-Line Equation	95
12.1 Native Mode	95
12.2 Print	95
13 Multi-Line with Single Column	97
13.1 Measure	97
13.2 Column Placement	98
13.3 Print	99
14 Multi-Line with Multiple Columns	100
14.1 Support	100
14.2 Transpose	101
14.3 Measure	103
14.4 Columns Placement	105
14.5 Print	106
15 Interface	109
15.1 Scanning the Equation Body	109
15.2 Options Processing	112
15.3 Single-Line Main	114
15.4 Multi-Line Main	116
15.5 Equations Environment	117
16 Options	120
16.1 Selection Tools	120
16.2 Options Declarations	121
16.3 Parameter Presets	133
16.4 Component Selection	135
16.5 Global and Package Options	146

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 Debugging Messages

The package offers a verbose mode for debugging purposes. It outputs extra information on the current location within the code in order to track progress: **TODO:** describe

```

1 \def\eql@verbose@on{%
2   \def\eql@verbose@info##1{\PackageInfo{eqnlines}{##1}}
3   \def\eql@verbose@infoarg##1##2{\eql@verbose@info{##1##2}}
4 }
5 \def\eql@verbose@off{%
6   \let\eql@verbose@info\@gobble
7   \let\eql@verbose@infoarg\@gobbletwo
8 }
9 \eql@verbose@off

```

TODO: describe

```

10 \def\eql@verbose@msg@enterenv{entering \@currentenv}
11 \def\eql@verbose@msg@leaveenv{ leaving \@currentenv}
12 \def\eql@verbose@msg@start#1{starting \string#1}
13 \def\eql@verbose@msg@end#1{ \space ending \string#1}
14 \def\eql@verbose@msg@within#1{ \space within \string#1}
15 \def\eql@verbose@msg@enter#1{entering \string#1}
16 \def\eql@verbose@msg@leave#1{ leaving \string#1}
17 \def\eql@verbose@msg@startline{starting line \the\eql@row@}

```

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:

```

18 \let\eql@false\@undefined
19 \let\eql@true\@empty

```

TODO: describe

```

20 \def\eql@append#1#2{\edef#1{\unexpanded\expandafter{#1#2}}}
21 \def\eql@appendexpand#1#2{\edef#1{\unexpanded\expandafter{#1}#2}}
22 \def\eql@appendmacro#1#2{\eql@appendexpand#1{\unexpanded\expandafter{#2}}}
23 \def\eql@letcs#1{\expandafter\let\csname#1\endcsname}

```

2.3 Dollardollar Abstraction

`\dollar@dollar@begin` As of 2025 L^AT_EX defines `\dollar@dollar@begin` and `\dollar@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:

```

24 \ifdefined\dollar@dollar@begin
25   \def\eql@dollar@dollar@begin{\dollar@dollar@begin}
26   \def\eql@dollar@dollar@end{\dollar@dollar@end}
27 \else
28   \def\eql@dollar@dollar@begin{$$$}
29   \def\eql@dollar@dollar@end{$$$}
30 \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.

`\eq@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 `\eq@ifnextchar@loose`. We replicate the amsgen version `\new@ifnextchar` that does not skip over spaces as `\eq@ifnextchar@loose`. The space before #1 allows to look-ahead for spaces as well:

```
31 \let\eq@ifnextchar@loose\kernel@ifnextchar
32 \long\def\eq@ifnextchar@tight#1#2#3{%
33   \let\reserved@a=#1%
34   \def\reserved@a{#2}%
35   \def\reserved@b{#3}%
36   \futurelet\@let@token\eq@ifnch@tight
37 }
38 \def\eq@ifnch@tight{%
39   \ifx\@let@token\reserved@a
40     \let\reserved@b\reserved@a
41   \fi
42   \reserved@b
43 }
```

`\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:

```
44 \let\eq@atxi=@
45 \begingroup
46   \makeatother
47   \let\tmp=@%
48   \makeatletter
49   \global\let\eq@atxii\tmp
50 \endgroup
```

`\eq@ifnextgobble@...` We introduce a collection of look-ahead macros which do or do not skip over spaces. The macros `\eq@ifstar@...` and `\eq@testopt@...` replicate the L^AT_EX counterparts `\@ifstar` and `\@testopt`. The macros `\eq@ifnextgobble@...` work like `\@ifnextchar`, 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 [...]:

```
51 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
52 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
53 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
54 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
55 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{#2}}
56 \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}
57 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{#2}}
58 \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}
59 \long\def\eq@testopt@loose#1#2{\eq@ifnextchar@loose[{#1}]{#1[{#2}]}}
60 \long\def\eq@testopt@tight#1#2{\eq@ifnextchar@tight[{#1}]{#1[{#2}]}}
61 \long\def\eq@teststaropt@loose#1#2#3{%
62   \eq@ifstar@loose{\eq@testopt@loose{#1}{#3}}{\eq@testopt@loose{#2}{#3}}}
63 \long\def\eq@teststaropt@tight#1#2#3{%
64   \eq@ifstar@tight{\eq@testopt@tight{#1}{#3}}{\eq@testopt@tight{#2}{#3}}}
65 \long\def\eq@teststaroropt@loose#1#2#3{%
66   \eq@ifstar@loose{#1}{\eq@testopt@loose{#2}{#3}}}
67 \long\def\eq@teststaroropt@tight#1#2#3{%
```

```

68 \eq@ifstar@tight{#1}{\eq@testopt@tight{#2}{#3}}
69 \long\def\eq@gobbleopt[#1]{}
70 \long\def\eq@gobbleoptone[#1]#2{}

```

TODO: describe

```

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

```

TODO: describe

```

72 \let\eq@parseopt@warn@main\@empty
73 \let\eq@parseopt@warn@aux\@empty

```

TODO: describe

```

74 \def\eq@parseopt@main{%
75   \let\eq@parseopt@warn\eq@parseopt@warn@main\eq@parseopt}
76 \def\eq@parseopt@aux{%
77   \let\eq@parseopt@warn\eq@parseopt@warn@aux\eq@parseopt}

```

TODO: describe

```

78 \def\eq@parseopt#1#2{%
79   \def\eq@parseopt@case{#1}%
80   \def\eq@parseopt@end{#2}%
81   \eq@parseopt@peek
82 }
83 \def\eq@parseopt@peek{%
84   \futurelet\eq@parseopt@token\eq@parseopt@select
85 }
86 \def\eq@parseopt@select{%
87   \let\eq@parseopt@next\eq@parseopt@other
88   \ifx\eq@parseopt@token\@sptoken
89     \let\eq@parseopt@next\eq@parseopt@end
90   \fi
91   \eq@parseopt@case
92   \eq@parseopt@next
93 }
94 \def\eq@parseopt@other{\eq@parseopt@warn\eq@parseopt@end}
95 \def\eq@parseopt@gobble#1{\eq@parseopt@peek}

```

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

```

96 \def\eq@spbggroup{\iffalse{\fi\ifnum0='}\fi}
97 \def\eq@speggroup{\ifnum0='{ \fi\iffalse}\fi}
98 \def\eq@srbgroup{\relax\iffalse{\fi\ifnum0='}\fi}
99 \def\eq@sregroup{\let\@let@token\relax\ifnum0='{ \fi\iffalse}\fi}

```

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

```

100 \long\def\eq@ampprotect#1#2{\eq@srbgroup#1{\eq@sregroup#2}}
101 \long\def\eq@ampprotecttwo#1#2#3{%
102   \eq@srbgroup#1{\eq@sregroup#2}{\eq@sregroup#3}}

```

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>.

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

```

103 \def\eq@ifnextchar@loose@ampsafe#1{%
104   \eq@ampprotecttwo{\eq@ifnextchar@loose#1}}
105 \def\eq@ifnextchar@tight@ampsafe#1{%
106   \eq@ampprotecttwo{\eq@ifnextchar@tight#1}}
107 \def\eq@ifstar@loose@ampsafe{\eq@ampprotecttwo\eq@ifstar@loose}
108 \def\eq@ifstar@tight@ampsafe{\eq@ampprotecttwo\eq@ifstar@tight}
109 \def\eq@testopt@loose@ampsafe{\eq@ampprotect\eq@testopt@loose}
110 \def\eq@testopt@tight@ampsafe{\eq@ampprotect\eq@testopt@tight}
111 \def\eq@teststaropt@loose@ampsafe{\eq@ampprotecttwo\eq@teststaropt@loose}
112 \long\def\eq@teststaropt@tight@ampsafe{%
113   \eq@ampprotecttwo\eq@teststaropt@tight}

```

\eq@amproof We may want to replace L^AT_EX’s definitions \@ifnextchar, \@ifstar and \@testopt to respect ‘&’ characters within aligned equations. This might make unrelated definitions with optional arguments and starred variants more robust in this context. The macro \eq@amproof overwrites the original definitions, and \eq@amprevert reverts the changes:

```

114 \let\eq@ifnextchar@org\@ifnextchar
115 \let\eq@ifstar@org\@ifstar
116 \let\eq@testopt@org\@testopt
117 \def\eq@amprevert{%
118   \let\@ifnextchar\eq@ifnextchar@org
119   \let\@testopt\eq@testopt@org
120   \let\@ifstar\eq@ifstar@org
121 }
122 \def\eq@amproof{%
123   \let\@ifnextchar\eq@ifnextchar@loose@ampsafe
124   \let\@testopt\eq@testopt@loose@ampsafe
125   \let\@ifstar\eq@ifstar@loose@ampsafe
126 }

```

2.5 Error Messages

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

```

127 \def\eq@error#1{\PackageError{eqnlines}{#1}{}}
128 \def\eq@warning{\PackageWarning{eqnlines}}

```

\eq@error@mathmode Error messages concerning math mode:

```

129 \def\eq@warn@here#1{\eq@warning{\string#1 not allowed outside equations}}
130 \def\eq@error@mathmode#1{\eq@error{#1 allowed only in paragraph mode}}

```

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

```

\eq@warn@label@multiple
\eq@warn@tag@unused
\eq@warn@tag@multiple
\eq@warn@name@unused
\eq@warn@name@multiple
\eq@warn@ref@unused
\eq@warn@ref@multiple
131 \def\eq@warn@tags@unused#1#2{\eq@warning{Unused equation #1:
132   #2 will be lost}}
133 \def\eq@warn@tags@multiple#1#2#3{\eq@warning{Multiple equation #1:
134   previous #2 will be lost#3}}
135 \def\eq@warn@label@unused{\eq@warn@tags@unused{\string\label}}
136   {label '\eq@tags@label'}}
137 \def\eq@warn@label@multiple#1{\eq@warn@tags@multiple{\string\label's}
138   {label '\eq@tags@label'}}{ and replaced by '#1'}}
139 \def\eq@warn@name@unused{\eq@warn@tags@unused{label name}
140   {name declaration}}
141 \def\eq@warn@name@multiple{\eq@warn@tags@multiple{label names}

```

```

142 {name declaration}{}
143 \def\eq@warn@tag@unused{\eq@warn@tags@unused{\string\tag}
144 {tag declaration}}
145 \def\eq@warn@tag@multiple{\eq@warn@tags@multiple{\string\tag's}
146 {tag declaration will be lost}{}
147 \def\eq@warn@ref@unused{\eq@warn@tags@unused{tag label}
148 {tag label declaration}}
149 \def\eq@warn@ref@multiple{\eq@warn@tags@multiple{tag labels}
150 {tag label declaration}{}

151 \def\eq@warn@parseopt{\eq@warning{Unknown modifier token:
152 modifier parsing stopped}}
153 \def\eq@warn@parseopt@verbose{\eq@warning{Unknown modifier token:
154 '\meaning\eq@parseopt@token'}}

```

2.6 amsmath Integration

`\eq@amsmath@after` We need to overwrite certain macros from `amsmath`. The method `\eq@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 `\eq@amsmath@before` will execute argument #1 before any future loading of `amsmath`. `\eq@amsmath@undefine` undefines a macro in this way and `\eq@amsmath@let` overwrites a macro of `\amsmath/`:

```

155 \def\eq@amsmath@after#1{\AddToHook{package/amsmath/after}{#1}}
156 \def\eq@amsmath@before#1{%
157 \ifpackage@loaded{amsmath}{}{\AddToHook{package/amsmath/before}{#1}}
158 \def\eq@amsmath@undefine#1{\eq@amsmath@before{\let#1\undefined}}
159 \def\eq@amsmath@let#1#2{\eq@amsmath@undefine#1\let#1#2}

```

TODO: temporary fix for development stages

```

160 \ifpackage@loaded{amsmath}{}{
161 \DeclareHookRule{package/amsmath/after}
162 {eqnlines}{after}{latex-lab-testphase-math}}

```

2.7 PDF Tagging Support

`\eq@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

```

163 \let\eq@tagging@on\eq@false
164 \IfFormatAtLeastTF{2025-06-01}{%
165 \csname tag_if_active:T\endcsname{\let\eq@tagging@on\eq@true}}{}
166 \ifdefined\eq@tagging@on
167 \def\eq@tagging@mathsave{%
168 \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
169 \def\eq@tagging@mathaddlast{%
170 \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
171 \def\eq@tagging@tagbegin{%
172 \UseTaggingSocket{math/display/tag/begin}}
173 \def\eq@tagging@tagend{%
174 \UseTaggingSocket{math/display/tag/end}}
175 \def\eq@tagging@tagsave{%
176 \UseTaggingSocket{math/luamml/mtable/tag/save}}

```

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


```

177 \def\eql@tagging@tagaddbox{%
178   \setbox\z@\copy\eql@tagbox@%
179   \UseTaggingSocket{math/luamml/mtable/tag/set}}
180 \def\eql@tagging@tablesaveinner{%
181   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
182 \def\eql@tagging@tableaddinner{%
183   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
184 \def\eql@tagging@tablesavelines{%
185   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
186 \def\eql@tagging@tablesavealign{%
187   \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
188 \def\eql@tagging@alignleft{%
189   \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
190 \def\eql@tagging@aligncenter{%
191   \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
192 \def\eql@tagging@alignright{%
193   \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:

```

194 \let\eql@single@doscan\eql@true
195 \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:

```

196 \def\eql@tagging@start{%
197   \csname bool_if:N\expandafter\endcsname
198     \csname l__math_collected_bool\endcsname{%
199     \edef\eql@tmp{\@currenvir}{\unexpanded\expandafter{\eql@tagging@opt}}}%
200     \the\eql@scan@reg@}}%
201   \csname __math_process:nn\expandafter\endcsname\eql@tmp
202   \@kernel@math@registered@begin
203   \csname bool_set_true:N\expandafter\endcsname
204     \csname l__math_collected_bool\endcsname
205   }%
206 }
207 \def\eql@tagging@end{}
208 \def\eql@tagging@register@luamml#1{%
209   \AddToHook{package/luamml/after}{%
210     \eql@letcs{c__luamml_label_#1_tl}{\@empty}}
211 \def\eql@tagging@register@env{\csname math_register_env:n\endcsname}

```

When tagging is deactivated, provide empty definitions:

```

212 \else
213   \let\eql@tagging@mathsave\@empty
214   \let\eql@tagging@mathaddlast\@empty
215   \let\eql@tagging@tagbegin\@empty
216   \let\eql@tagging@tagend\@empty
217   \let\eql@tagging@tagsave\@empty
218   \let\eql@tagging@tagaddbox\@empty
219   \let\eql@tagging@tablesaveinner\@empty
220   \let\eql@tagging@tableaddinner\@empty

```

```

221 \let\eql@tagging@tablesavelines\@empty
222 \let\eql@tagging@tablesalign\@empty
223 \let\eql@tagging@alignleft\@empty
224 \let\eql@tagging@aligncenter\@empty
225 \let\eql@tagging@alignright\@empty
226 \let\eql@tagging@start\@empty
227 \let\eql@tagging@end\@empty
228 \let\eql@tagging@register@luamml\@gobble
229 \let\eql@tagging@register@env\@gobble
230 \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:

```
231 \RequirePackage{keyval}
```

Value Selection.

`\eql@decide@select` Some parameter values take values in a given set, e.g. `true` vs. `false` or `left` vs. `right`. The macro `\eql@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:

```

232 \def\eql@decide@relax{\eql@tmpb:=\relax}
233 \def\eql@decide@select#1#2#3#4{%
234   \def\eql@tmpa{#3}%
235   \let\eql@tmpd\@undefined
236   \@for\eql@tmpc:=#4\do{%
237     \ifdefined\eql@tmpd\else
238       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
239       \ifx\eql@tmpb\eql@decide@relax
240         \def\eql@tmpa{\relax}%
241       \fi
242       \expandafter\@for\eql@tmpb\do{%
243         \ifx\eql@tmpa\eql@tmpb
244           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
245             \expandafter\@secondoftwo\eql@tmpc}}}%
246         \fi
247       }%
248     \fi
249   }%
250   \ifdefined\eql@tmpd
251     \eql@tmpd
252   \else
253     \eql@error{undefined value '#3' for option '#2' of '#1'}%
254   \fi
255 }

```

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

```

256 \def\eql@decide@true{on,true,yes,enabled}
257 \def\eql@decide@false{off,false,no,disabled}

```

`\eql@decide@if`

```

258 \def\eql@decide@if#1#2#3#4#5{%
259   \eql@decide@select{#1}{#2}{#3}{%
260     {\eql@decide@true{#4}},%
261     {\eql@decide@false{#5}}}%

```

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

```

262 \def\eql@decide@bool#1#2#3#4{%
263   \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:

```

264 \def\eql@define@key#1#2{%
265   \eql@ifnextchar@loose[%
266     {\eql@definekey@opt{#1}{#2}}%
267     {\eql@definekey@noopt{#1}{#2}}%
268 }
269 \def\eql@definekey@noopt#1#2#3{\eql@definekey@for{#1}{#2}{#3}}
270 \def\eql@definekey@opt#1#2[#3]#4{\eql@definekey@for{#1}{#2}{[#3]{#4}}}
271 \def\eql@definekey@for#1#2#3{%
272   \def\eql@for@fn##1##2##3{\define@key{eql@##3}{##2}{#3}%
273     \edef\eql@for@vara{\noexpand\eql@for@vara:=#1}%
274     \expandafter\@for\eql@for@vara\do{%
275       \edef\eql@for@varb{\noexpand\eql@for@varb:=#2}%
276       \expandafter\@for\eql@for@varb\do{%
277         \edef\eql@for@call##1{%
278           \noexpand\eql@for@fn{##1}{\eql@for@varb}{\eql@for@vara}}%
279         \eql@for@call{##1}%
280       }%
281     }%
282 }

```

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

```

283 \def\eql@setkeys#1#2{%
284   \def\eql@tmp{\setkeys{eql@#1}}%
285   \expandafter\eql@tmp\expandafter{#2}%
286 }

```

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:

```

287 \let\eql@nextopt\@empty
288 \def\eql@nextopt@process#1{%
289   \eql@setkeys{#1}\eql@nextopt

```

```

290 \let\eql@tagging@opt\eql@nextopt
291 \global\let\eql@nextopt\@empty
292 }

```

`\eqnaddopt`

```

293 \newcommand{\eqnaddopt}[1]{%
294   \ifx\eql@nextopt\@empty
295     \eql@append\eql@nextopt{#1}%
296   \else
297     \eql@append\eql@nextopt{, #1}%
298   \fi
299 }

```

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

```

300 \newcommand{\eqnlineset}[1]{%
301   \eql@setkeys{setup}{#1}%
302   \ignorespaces
303 }

```

`\eql@control@default`

```

304 \protected\def\eql@control@default{%
305   \eql@warn@here\eqncontrol
306   \@gobble
307 }
308 \let\eqncontrol\eql@control@default

```

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

```

309 \newcommand{\eql@control}[1]{%
310   \relax
311   \eql@setkeys{control}{#1}%
312   \ignorespaces
313 }

```

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:

```

314 \let\eql@tagsleft\eql@false

```

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

```

315 \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`:

```
316 \def\eql@layoutleftmargin{\leftmargini}
317 \def\eql@layoutleftmarginmax{.5\maxdimen}
318 \def\eql@layoutleftmarginmin{\z@}
```

`\eql@mathstyle` The math style to be used within cells is specified by `\eql@mathstyle`:

```
319 \let\eql@mathstyle\displaystyle
```

`\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@ratio@` (*dimen*) `\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

```
320 \newdimen\eql@tagmargin@
321 \let\eql@tagmargin@val\undefined
322 \newdimen\eql@tagmargin@ratio@
323 \eql@tagmargin@ratio@\p@
324 \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:

```
325 \newdimen\eql@indent@
326 \def\eql@indent@val{2em}
```

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

```
327 \newdimen\eql@paddingleft@
328 \newdimen\eql@paddingright@
329 \let\eql@paddingleft@val\undefined
330 \let\eql@paddingright@val\undefined
```

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

```
331 \let\eql@display@linewidth\undefined
332 \let\eql@display@marginleft\undefined
333 \let\eql@display@marginright\undefined
```

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

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

```
\eql@box@condsep
334 \def\eql@box@colsep{2em}
335 \def\eql@box@shortsep{1em}
336 \def\eql@box@condsep{\eql@box@shortsep}
```

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

```
\eql@break@line@shortsep
337 \def\eql@break@line@sep{2em minus 1em}
\eql@break@col@sep
338 \def\eql@break@line@shortsep{1em}
1@break@col@shortsep
339 \def\eql@break@col@sep{2em minus 1em}
340 \def\eql@break@col@shortsep{1em}
```

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

```
341 \let\eql@spread@reset\eql@false
342 \def\eql@spread@val{\jot}
343 \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 T_EX, nevertheless: **TODO**: probably do not need this due to fixed point arithmetic.

```
344 \newdimen\eql@tagfuzz@
345 \eql@tagfuzz@16sp\relax
```

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

```
346 \def\eql@display@height\@undefined
347 \def\eql@display@depth\@undefined
```

`\eql@skip@mode@short` The setting `\eql@skip@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:

```
348 \def\eql@skip@mode@short{2}

349 \def\eql@skip@mode@cont@above{2}
350 \def\eql@skip@mode@cont@below{0}

351 \def\eql@skip@mode@par@above{3}
352 \def\eql@skip@mode@par@below{0}

353 \def\eql@skip@mode@top@above{4}
354 \def\eql@skip@mode@top@below{0}

355 \newcount\eql@skip@mode@leave@
356 \let\eql@skip@force@leave\@undefined
```

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

`\eql@skip@force@below`
`\mode@above@` (*counter*)
`\mode@below@` (*counter*)

```
357 \newcount\eql@skip@mode@above@
358 \newcount\eql@skip@mode@below@
359 \let\eql@skip@force@above\@undefined
360 \let\eql@skip@force@below\@undefined
361 \let\eql@skip@custom@above\@undefined
362 \let\eql@skip@custom@below\@undefined
```

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

`\eql@skip@top@above` The glue when an equation is at the top of a vertical list is specified by `\eql@skip@top@above` and `\eql@skip@top@below`:

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

`\eql@skip@med@above` The surrounding glue for an equation with reduced spacing is given by `\eql@skip@med@above` and `\eql@skip@med@below`:

```

363 \def\eq@skip@long@above{\abovedisplayskip}
364 \def\eq@skip@long@below{\belowdisplayskip}
365 \def\eq@skip@short@above{\abovedisplayshortskip}
366 \def\eq@skip@short@below{\belowdisplayshortskip}
367 \def\eq@skip@cont@above{\eq@skip@short@above}
368 \def\eq@skip@cont@below{\eq@skip@short@below}
369 \def\eq@skip@par@above{\eq@skip@long@above}
370 \def\eq@skip@par@below{\eq@skip@long@below}
371 \def\eq@skip@top@above{\eq@skip@long@above}
372 \def\eq@skip@top@below{\eq@skip@long@below}
373 \def\eq@skip@med@above{\abovedisplayskip/2}
374 \def\eq@skip@med@below{\belowdisplayskip/2}
375 \def\eq@skip@tag@above{\z@skip}
376 \def\eq@skip@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:

```

377 \newdimen\eq@colsepmin@
378 \def\eq@colsepmin@val{1em}
379 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

380 \newdimen\eq@tagwidthmin@
381 \eq@tagwidthmin@\z@

```

`\eq@tagsepmin@` (*dimen*) The minimum separation between an equation and its tag is given by `\eq@tagsepmin@`. $\mathrm{T}_{\mathrm{E}}\mathrm{X}$'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.

```

382 \newdimen\eq@tagsepmin@
383 \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
384 \def\eq@equations@sqr@opt{equation,nonumber}
385 \def\eq@equations@ang@opt{align,nonumber}
386 \def\eq@box@ang@opt{align}

```

Multi-Line Align Mode.

```

387 \let\eq@columns@fulllength\eq@false

```

3.2 Registers

TODO: describe

General. **TODO:** describe

```

388 \newcount\eq@count@
389 \newdimen\eq@dimen@
390 \newskip\eq@skip@

```

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

TODO: describe

```
391 \let\eqldisplay@container\@empty
```

$\backslash\mathrm{eqldcellbox@}$ (*box*) The box $\backslash\mathrm{eqldcellbox@}$ holds the present alignment component and $\backslash\mathrm{eqldtagbox@}$ the tag for the present line. The corresponding dimensions $\backslash\mathrm{eqldcellwidth@}$ and $\backslash\mathrm{eqldtagwidth@}$ hold their widths. $\backslash\mathrm{eqldprevwidth@}$ holds the width of the previous alignment component: **TODO:** adjust

```

\eqldtagwidth@ (dimen) 392 \newbox\eqldcellbox@
\eqldprevdepth@ (dimen) 393 \newbox\eqldtagbox@
\eqldprevgraf@ (counter) 394 \newdimen\eqldcellwidth@
395 \newdimen\eqldprevwidth@
396 \newdimen\eqldtagwidth@
397 \newdimen\eqldprevdepth@
398 \newcount\eqldprevgraf@
```

```

\eqldtotalwidth@ (dimen)
\eqldtagwidth@max@ (dimen) 399 \newdimen\eqldtotalwidth@
\eqldtotalheight@ (dimen) 400 \newdimen\eqldtagwidth@max@
401 \newdimen\eqldtotalheight@
402 \newdimen\eqldtopheight@
403 \newdimen\eqldbottomdepth@
```

$\backslash\mathrm{eqldline@height@}$ (*dimen*) The dimension registers $\backslash\mathrm{eqldline@height@}$ and $\backslash\mathrm{eqldline@depth@}$ keep track of the height and depth of the present line in an alignment:

```

404 \newdimen\eqldline@height@
405 \newdimen\eqldline@depth@
```

```

\eqldline@width@ (dimen)
\eqldline@avail@ (dimen) 406 \newdimen\eqldline@width@
\eqldline@pos@ (dimen) 407 \newdimen\eqldline@avail@
\eqldwidthsep@ (counter) 408 \newdimen\eqldline@pos@
\eqldavailsep@ (counter) 409 \newcount\eqldline@availsep@
\eqldpossep@ (counter) 410 \newcount\eqldline@widthsep@
\eqldline@offset@ (dimen) 411 \newcount\eqldline@possep@
\eqldprevdepth@ (dimen) 412 \newdimen\eqldline@offset@
\eqldinterline@ (dimen) 413 \newdimen\eqldline@prevdepth@
414 \newdimen\eqldline@interline@
```

Rows and Columns.

$\backslash\mathrm{eqldrow@}$ (*counter*) **TODO:** tagrows $\backslash\mathrm{eqldrow@}$ counts the present row (1-based) and $\backslash\mathrm{eqldtotalrows@}$ holds the total number of rows:

```

\eqldtagrows@ (counter) 415 \newcount\eqldrow@
416 \newcount\eqldtotalrows@
417 \newcount\eqldtagrows@
```

```

\eqldcolumn@
\eqldtotalcolumns@ 418 \newcount\eqldcolumn@
419 \newcount\eqldtotalcolumns@
```


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

```
420 \newdimen\eq@colsep@
```

`\intercolumns@` (*counter*)

```
421 \newcount\eq@intercolumns@
```

Vertical Spacing Adjustments.

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

```
422 \newdimen\eq@display@firstavail@
423 \def\eq@display@firstavail@set#1{%
424   \ifnum\eq@row@=\@one
425     \global\eq@appendexpand\eq@display@container{%
426       \eq@display@firstavail@the#1\relax}%
427   \fi
428 }
```

The counter stores whether the tag on 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

```
429 \newdimen\eq@display@aboveextend@
430 \newdimen\eq@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:

```
431 \ifdefined\measuring@true\else
432   \expandafter\newif\csname ifmeasuring@\endcsname
433 \fi
434 \AddToHook{package/hyperref/after}{
435   \ifdefined\Hy@ifnotmeasuring
436     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
437   \fi
438 }
```

`\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:

```
439 \ifdefined\@displaytrue\else
440   \expandafter\newif\csname if@display\endcsname
441   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
442 \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

```
443 \let\eql@hook@blockbefore\@empty
444 \let\eql@hook@blockafter\@empty
445 \let\eql@hook@blockin\@empty
446 \let\eql@hook@blockout\@empty
447 \let\eql@hook@linein\@empty
448 \let\eql@hook@lineout\@empty
449 \let\eql@hook@colin\@empty
450 \let\eql@hook@colout\@empty
451 \let\eql@hook@eqin\@empty
452 \let\eql@hook@eqout\@empty
453 \let\eql@hook@innerleft\@empty
454 \let\eql@hook@innerright\@empty
455 \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. An undefined value indicates that the punctuation should be handed down to the next lower level:

`\eql@punct@block` **TODO:** update

```
\eql@punct@next
\eql@punct@term
\eql@punct@main
456 \let\eql@punct@col\@empty
457 \let\eql@punct@line\@undefined
458 \let\eql@punct@block\@undefined
459 \let\eql@punct@next\@undefined
460 \let\eql@punct@term\@undefined
461 \let\eql@punct@main\@undefined
```

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

```
462 \let\eql@punct@sep\@empty
```

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

`\eql@punct@term@stop`

```
463 \def\eql@punct@term@set{%
464   \let\eql@punct@term\eql@punct@block}
465 \def\eql@punct@term@stop{%
466   \global\let\eql@punct@term\relax}
```

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

```
467 \def\eql@punct@relax{\relax}
468 \def\eql@punct@tilde{~}
469 \def\eql@punct@set#1#2{%
470   \def#1{#2}%
471   \ifx#1\eql@punct@relax
```

```

472 \let#1\@undefined
473 \fi
474 \ifx#1\eql@punct@tilde
475 \let#1\@empty
476 \fi
477 }
478 \def\eql@punct@clear{%
479 \let\eql@punct@col\@empty
480 \let\eql@punct@line\@empty
481 \let\eql@punct@main\@empty
482 }

```

\eqnpunct Set the punctuation 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: **TODO**: check operation of starred form, also for control

```

483 \newcommand{\eqnpunct}{%
484 \eql@ifstar@tight\eql@punct@next@setrelax\eql@punct@next@set}
485 \def\eql@punct@next@set#1{%
486 \ifmmode
487 \eql@punct@set\eql@punct@next{#1}%
488 \else
489 \eqnaddopt{punct={#1}}%
490 \fi
491 \ignorespaces}
492 \def\eql@punct@next@setrelax{%
493 \ifmmode
494 \let\eql@punct@block\@undefined
495 \else
496 \eqnaddopt{punct*}%
497 \fi
498 \ignorespaces}

```

\eql@punct@fill@next Fill the next punctuation:

```

499 \def\eql@punct@fill@next#1{%
500 \ifdefined\eql@punct@next\else
501 \ifx\eql@punct@term\relax\else
502 \let\eql@punct@next#1%
503 \fi
504 \fi
505 }

```

\eql@punct@output@next Output the next punctuation. If non-empty, prepend some separation:

```

506 \def\eql@punct@output@next{%
507 \ifx\eql@punct@next\@empty\else
508 \eql@punct@sep
509 \eql@punct@next
510 \fi
511 \let\eql@punct@next\@undefined
512 }

```

\eql@punct@print@next Print the next punctuation if available:

```

513 \def\eql@punct@print@next{%
514 \ifdefined\eql@punct@next

```

```

515 \eql@punct@output@next
516 \fi
517 }

```

`\eql@punct@apply@next` Print the next punctuation if available. Stop further punctuation within the current group:

```

518 \def\eql@punct@apply@next{%
519 \ifdefined\eql@punct@next
520 \eql@punct@output@next
521 \let\eql@punct@term\relax
522 \fi
523 }

```

`\eql@punct@print@col` Print the punctuation for the present column:

```

524 \def\eql@punct@print@col{%
525 \eql@punct@fill@next\eql@punct@col
526 \eql@punct@print@next
527 }

```

`\eql@punct@apply@col` Output the punctuation for the present column. Stop further punctuation within the current group:

```

528 \def\eql@punct@apply@col{%
529 \eql@punct@fill@next\eql@punct@col
530 \eql@punct@apply@next
531 }

```

Output the punctuation for the present line unless disabled:

`\eql@punct@print@line`

```

532 \def\eql@punct@print@line{%
533 \eql@punct@fill@next\eql@punct@line
534 \eql@punct@print@next
535 }

```

Output the punctuation for the present line unless disabled. Stop further punctuation within the current group:

`\eql@punct@apply@line`

```

536 \def\eql@punct@apply@line{%
537 \eql@punct@fill@next\eql@punct@line
538 \eql@punct@apply@next
539 }

```

`\eql@punct@apply@block` Output the punctuation for the present block unless disabled. Stop further punctuation within the current group:

```

540 \def\eql@punct@apply@block{%
541 \eql@punct@fill@next\eql@punct@block
542 \eql@punct@apply@next
543 }

```

`\eqnpunctapply` Output the terminal punctuation unless disabled. Stop further punctuation globally:

```

544 \newcommand{\eqnpunctapply}{%
545 \ifmmode\else\unskip\fi

```

```

546 \eqlopunct@fill@next\eqlopunct@term
547 \eqlopunct@print@next
548 \global\let\eqlopunct@term\relax
549 }

```

4.2 Math Classes at Alignment

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

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

```

550 \def\eqlopclass@innerright@sel@{%
551   \ifdim\eqlop@prevwidth@=\z@
552     \eqlopclass@innerlead
553   \else
554     \eqlopclass@innerright
555   \fi
556 }

```

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

`\eqlopclass@innerright@set`

`\eqlopclass@innerlead@set`

```

557 \def\eqlopclass@innerleft@set#1{%
558   \def\eqlopclass@innerleft{#1}%
559 }
560 \def\eqlopclass@innerright@set#1{%
561   \def\eqlopclass@innerright{#1}%
562   \let\eqlopclass@innerright@sel\eqlopclass@innerright
563 }
564 \def\eqlopclass@innerlead@set#1{%
565   \def\eqlopclass@innerlead{#1}%
566   \let\eqlopclass@innerright@sel\eqlopclass@innerright@sel@
567 }

```

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

```

568 \def\eqlopclass@ampeq{%
569   \eqlopclass@innerleft@set{}%
570   \eqlopclass@innerright@set{{}}%
571 }
572 \def\eqlopclass@eqamp{%
573   \eqlopclass@innerleft@set{\mathrel{}}%
574   \eqlopclass@innerright@set{\mathrel{}}%
575   \eqlopclass@innerlead@set{{}}%
576 }
577 \eqlopclass@ampeq

```

4.3 Framed Cells

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

```

578 \let\eqlopframe@cmd\@undefined
579 \newdimen\eqlopframe@margin@
580 \def\eqlopframe@set[#1]{%
581   \global\eqlopappend\eqlop@cell@container{\def\eqlopframe@cmd{#1}}

```

```

582 \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbx}
583 \def\eql@frame@measure{%
584   \setbox\z@\hbox{\eql@frame@cmd{}}}%
585   \eql@frame@margin@.5\wd\z@
586 }
587 \def\eql@frame@print{%
588   \setbox\eql@cellbox@\hbox{%
589     \eql@frame@cmd{\unhbox\eql@cellbox@}%
590   }%
591 }
592 \def\eql@frame@adjust{%
593   \setbox\eql@cellbox@\hbox{%
594     \eql@frame@cmd{%
595       \unhbox\eql@cellbox@
596       \unkern
597       \unskip
598     }%
599     \hfil
600     \kern\z@
601   }%
602 }

```

4.4 Single-Line Composition

TODO: describe

```

\eql@break@line
\eql@break@col
603 \def\eql@break@line{%
604   \let\eql@break@sep\eql@break@line@sep
605   \let\eql@break@shortsep\eql@break@line@shortsep
606   \let\eql@break@print\eql@punct@print@line
607   \eql@ampprotect\eql@break@testall\eql@break@process}
608 \def\eql@break@col{%
609   \let\eql@break@sep\eql@break@col@sep
610   \let\eql@break@shortsep\eql@break@col@shortsep
611   \let\eql@break@print\eql@punct@print@col
612   \eql@ampprotect\eql@break@testall\eql@break@process}

```

```

\eql@break@testall TODO: describe
\eql@break@parse
613 \def\eql@break@testall{\eql@parseopt@aux\eql@break@parse}
614 \def\eql@break@parse{%
615   \ifx\eql@parseopt@token[%
616     \let\eql@parseopt@next\eql@break@parse@val
617     \fi
618     \ifx\eql@parseopt@token*%
619       \let\eql@parseopt@next\eql@break@parse@star
620       \fi
621       \ifx\eql@parseopt@token.%
622         \let\eql@parseopt@next\eql@parseopt@punctpass
623         \fi
624         \ifx\eql@parseopt@token,%
625           \let\eql@parseopt@next\eql@parseopt@punctpass
626           \fi
627           \ifx\eql@parseopt@token~%
628             \let\eql@parseopt@next\eql@parseopt@punctpass
629             \fi

```

```

630 \ifx\eql@parseopt@token'%
631 \let\eql@parseopt@next\eql@parseopt@punctnext
632 \fi
633 }
634 \def\eql@break@parse@val[#1]{\def\eql@break@sep{#1}\eql@parseopt@peek}
635 \def\eql@break@parse@star#1{%
636 \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}

```

`\eql@break@process`

```

637 \def\eql@break@process{%
638 \eql@break@print
639 \hspace{\glueexpr\eql@break@sep\relax}}

```

`\eql@break@join`

```

640 \def\eql@break@join{\eql@srbgroup
641 \eql@ifstar@tight
642 {\eql@break@join@opt[\eql@break@col@shortsep]}%
643 {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
644 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
645 \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

`\eqnsep` **TODO:** expand to lines and columns mode
`\eqnbreak`
`\eqnjoin`

```

646 \newcommand{\eqnsep}{\eql@break@col}
647 \newcommand{\eqnbreak}{\eql@break@line}
648 \newcommand{\eqnjoin}{\eql@break@join}

```

4.5 Alternative Content Description

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

```

649 \DeclareRobustCommand{\eqnalt}[2][\relax]{}

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```

650 \let\eql@tags@autolabel\eql@false
651 \let\eql@tags@autotag\eql@true
652 \let\eql@tags@warn\eql@true

653 \def\eql@tags@name@generic[equation]}

654 \let\eql@tagpos@doconvert\eql@false

655 \def\eql@tagpos@snap{4pt}

```

Registers.

```
656 \let\eql@numbering@mode\@undefined

657 \let\eql@numbering@active\eql@true
658 \let\eql@numbering@multi\eql@true

659 \let\eql@tags@container\@undefined
660 \def\eql@tags@container@clear{%
661   \let\eql@tags@label\@undefined
662   \let\eql@tags@name\@undefined
663   \let\eql@tags@tag\@undefined
664   \let\eql@tags@ref\@undefined
665   \let\eql@tags@anchor\@empty
666   \eql@tagpos@shift@z@
667   \eql@tagpos@smashup@z@
668   \eql@tagpos@smashdown@z@
669   \let\eql@tagpos@reserve\eql@true
670 }

671 \let\eql@tags@label\@undefined
672 \let\eql@tags@name\@undefined
673 \let\eql@tags@tag\@undefined
674 \let\eql@tags@ref\@undefined
675 \let\eql@tags@frame@cmd\@firstofone
```

tags@glabel@ (*counter*)

```
676 \newcount\eql@tags@glabel@
677 \eql@tags@glabel@z@
678 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
679 \def\eql@tags@glabel@step{\global\advance\eql@tags@glabel@\@ne}

680 \let\eql@tagpos@continuous\eql@false

681 \newcount\eql@tagpos@row@
682 \newcount\eql@tagpos@prevrow@
683 \newdimen\eql@tagpos@shift@
684 \newdimen\eql@tagpos@smashup@
685 \newdimen\eql@tagpos@smashdown@
686 \newdimen\eql@tagpos@current@
687 \newdimen\eql@tagpos@plain@
688 \newdimen\eql@tagpos@raised@
689 \newdimen\eql@tagpos@target@
690 \newdimen\eql@tagpos@headroom@
691 \newdimen\eql@tagpos@footroom@
```

5.2 Schemes

TODO: describe

Table.

```
692 \def\eql@numbering@tab@sub{sub}
693 \def\eql@numbering@tab@all{all}
694 \def\eql@numbering@tab@first{first}
695 \def\eql@numbering@tab@last{last}
696 \def\eql@numbering@tab@in{in}
```



```

697 \def\eql@numbering@tab@out{out}
698 \def\eql@numbering@tab@middle{middle}
699 \def\eql@numbering@tab@best{best}
700 \def\eql@numbering@tab@here{here}
701 \def\eql@numbering@tab@top{top}
702 \def\eql@numbering@tab@bottom{bottom}
703 \def\eql@numbering@tab@center{center}
704 \def\eql@numbering@tab@centerone{centerone}
705 \def\eql@numbering@tab@median{median}
706 \def\eql@numbering@tab@baseline{baseline}

707 \let\eql@numbering@mode\eql@numbering@tab@all
708 \let\eql@numbering@mode@multi\eql@numbering@tab@all
709 \let\eql@numbering@mode@single\eql@numbering@tab@out

```

TODO: describe

```

710 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
711 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
712 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
713 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
714 \let\eql@numbering@tab@outside\eql@numbering@tab@out
715 \let\eql@numbering@tab@inside\eql@numbering@tab@in
716 \let\eql@numbering@tab@within\eql@numbering@tab@in
717 \let\eql@numbering@tab@opt\eql@numbering@tab@best
718 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
719 \let\eql@numbering@tab@pick\eql@numbering@tab@here
720 \let\eql@numbering@tab@med\eql@numbering@tab@median
721 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
722 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone

```

TODO: describe

```

723 \let\eql@numbering@tab@a\eql@numbering@tab@all
724 \let\eql@numbering@tab@s\eql@numbering@tab@sub
725 \let\eql@numbering@tab@f\eql@numbering@tab@first
726 \let\eql@numbering@tab@l\eql@numbering@tab@last
727 \let\eql@numbering@tab@o\eql@numbering@tab@out
728 \let\eql@numbering@tab@i\eql@numbering@tab@in
729 \let\eql@numbering@tab@h\eql@numbering@tab@here
730 \let\eql@numbering@tab@t\eql@numbering@tab@top
731 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
732 \let\eql@numbering@tab@c\eql@numbering@tab@center
733 \let\eql@numbering@tab@m\eql@numbering@tab@median
734 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
735 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
736 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
737 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone

```

Implementations. **TODO:** describe

```

738 \def\eql@numbering@init@all{\let\eql@numbering@mode\eql@numbering@multi\eql@true}

```

TODO: describe

```

739 \def\eql@numbering@init@sub{%
740   \let\eql@numbering@mode\eql@numbering@multi\eql@true
741   \ifdefined\eql@subequations@active
742     \let\eql@numbering@mode\eql@numbering@tab@all
743   \else

```

```

744 \let\eq@numbering@subeq@use\eq@true
745 \fi
746 }

747 \def\eq@numbering@init@first{\eq@tagpos@row@\@ne}
748 \def\eq@numbering@init@last{\eq@tagpos@row@\@MM}
749 \def\eq@numbering@init@here{\eq@tagpos@row@\m@ne}

```

TODO: describe

```

750 \def\eq@numbering@init@in{%
751 \ifdefined\eq@tagsleft
752 \eq@numbering@init@last
753 \else
754 \eq@numbering@init@first
755 \fi
756 }

```

TODO: describe

```

757 \def\eq@numbering@init@out{%
758 \ifdefined\eq@tagsleft
759 \eq@numbering@init@first
760 \else
761 \eq@numbering@init@last
762 \fi
763 }

```

TODO: describe

```

764 \def\eq@tagpos@eval@middle{%
765 \ifnum\eq@tagpos@row@=\z@
766 \eq@tagpos@row@\numexpr(\eq@totalrows@
767 +\ifdefined\eq@tagsleft\z@\else\@ne\fi)/\tw@\relax
768 \fi
769 }

```

TODO: describe

```

770 \def\eq@tagpos@eval@best{%
771 \ifnum\eq@tagpos@row@=\z@
772 \let\eq@numbering@best@use\eq@true
773 \eq@numbering@init@out
774 \fi
775 }

```

TODO: describe

```

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

```

TODO: describe

```

777 \let\eq@numbering@init@top\eq@numbering@init@continuous
778 \def\eq@tagpos@eval@top{%
779 \eq@tagpos@current@\z@
780 }

```

TODO: describe

```

781 \let\eq@numbering@init@bottom\eq@numbering@init@continuous
782 \def\eq@tagpos@eval@bottom{%
783 \eq@tagpos@current@\dimexpr\eq@totalheight@
784 -\eq@tagheight@block@-\eq@tagdepth@block@\relax
785 }

```

TODO: describe

```
786 \let\eql@numbering@init@center\eql@numbering@init@continuous
787 \def\eql@tagpos@eval@center{%
788   \ifnum\eql@totalrows@=\@ne
789     \eql@tagpos@row@\@ne
790   \fi
791   \eql@tagpos@current@\dimexpr(\eql@totalheight@
792     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
793 }
```

TODO: describe

```
794 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
795 \def\eql@tagpos@eval@centerone{%
796   \eql@tagpos@current@\dimexpr(\eql@totalheight@
797     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
798 }
```

TODO: describe

```
799 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
800 \def\eql@tagpos@eval@baseline{%
801   \eql@tagpos@current@\dimexpr(\eql@totalheight@
802     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
803 }
```

TODO: describe

```
804 \let\eql@numbering@init@median\eql@numbering@init@continuous
805 \def\eql@tagpos@eval@median{%
806   \ifnum\eql@tagpos@row@=\z@
807     \ifodd\eql@totalrows@
808       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
809     \else
810       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
811       \eql@dimensions@get\eql@tagpos@row@
812       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
813         +(\eql@line@interline@-\eql@tagheight@block@
814         +\eql@tagdepth@block@)/\tw@\relax
815     \fi
816   \ifnum\eql@totalrows@=\@ne
817     \eql@tagpos@row@\@ne
818   \else
819     \eql@tagpos@adjust@eval@convert
820     \eql@tagpos@row@\z@
821   \fi
822 \fi
823 }
```

Selection.

```
824 \def\eql@numbering@set#1{%
825   \ifcsname eql@numbering@tab@#1\endcsname
826     \expandafter\let\expandafter\eql@numbering@mode
827       \csname eql@numbering@tab@#1\endcsname
828   \ifx\eql@numbering@mode\eql@numbering@tab@all
829     \let\eql@numbering@mode@multi\eql@numbering@mode
830   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
831     \let\eql@numbering@mode@multi\eql@numbering@mode
```

```

832 \else
833 \let\eql@numbering@mode@single\eql@numbering@mode
834 \fi\fi
835 \else
836 \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
837 \let\eql@numbering@mode\eql@numbering@tab@all
838 \fi
839 }

```

TODO: describe

```

840 \def\eql@numbering@init{%
841 \let\eql@numbering@multi\eql@false
842 \let\eql@tagpos@continuous\eql@false
843 \let\eql@numbering@subeq@use\eql@false
844 \let\eql@numbering@best@use\eql@false
845 \eql@tagpos@row@z@
846 \csname eql@numbering@init@\eql@numbering@mode\endcsname
847 \ifdefined\eql@numbering@active
848 \let\eql@numbering@eqnswinit\@eqnswtrue
849 \else
850 \let\eql@numbering@eqnswinit\@eqnswfalse
851 \fi
852 \let\eql@numbering@active\eql@false
853 }

```

5.3 Interface

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

```

854 \eql@amsmath@after{
855 \let\eql@print@eqnum@default\print@eqnum
856 \let\eql@incr@eqnum@default\incr@eqnum
857 }

```

TODO: describe

```

858 \protected\def\donumber{%
859 \if@eqnsw\else
860 \global\@eqnswtrue
861 \ifx\print@eqn\@empty
862 \global\let\print@eqn\eql@print@eqnum@default
863 \fi
864 \ifx\incr@eqn\@empty
865 \global\let\incr@eqn\eql@incr@eqnum@default
866 \fi
867 \fi
868 }

```

TODO: reconsider operation

\numberhere

```

869 \protected\def\eql@numberhere{%
870 \ifdefined\eql@numbering@multi
871 \global\@eqnswtrue
872 \else
873 \global\eql@tagpos@row@\eql@row@
874 \fi
875 }

```

TODO: describe

`\numbernext`

```
876 \protected\def\eq\@numbernext{%
877   \ifdefined\eq\@numbering@multi
878     \global\@eqnswfalse
879   \else
880     \ifdefined\eq\@tagpos@continuous\else
881       \ifnum\eq\@tagpos@row@=\eq\@row@
882         \global\advance\eq\@tagpos@row@\@ne
883       \fi
884     \fi
885   \fi
886 }
```

Activation Trigger.

```
887 \def\eq\@tags@autoenable{%
888   \global\@eqnswtrue
889   \ifnum\eq\@tagpos@row@=\m@ne
890     \numberhere
891   \fi
892 }
```

Labels. **TODO:** describe

`\eq\@label@org`

```
893 \let\eq\@label@org\label
```

TODO: describe

```
894 \def\eq\@label@gobble{\eq\@ampprotect\eq\@testopt@tight\eq\@gobbleoptone{}}
```

TODO: describe

```
895 \protected\def\eq\@label{%
896   \eq\@ampprotect\eq\@testopt@tight\eq\@tags@add@labelname\eq\@testopt@default
897 }
```

TODO: describe

```
898 \def\eq\@tags@add@labelname[#1]#2{%
899   \def\eq\@tmp{#1}%
900   \ifx\eq\@tmp\eq\@testopt@default\else
901     \eq\@tags@add@name{#1}%
902   \fi
903   \eq\@tags@add@label{#2}%
904 }
```

TODO: describe

```
905 \def\eq\@tags@set@label#1{%
906   \ifdefined\eq\@tags@warn
907     \ifdefined\eq\@tags@label
908       \eq\@warn@label@multiple{#1}%
909     \fi
910   \fi
911   \def\eq\@tags@label{#1}%
912 }
```

TODO: describe

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

TODO: describe

```
921 \def\eql@tags@add@label#1{%
922   \ifdefined\eql@tags@autolabel
923     \eql@tags@autoenable
924   \fi
925   \global\eql@appendexpand\eql@tags@container{%
926     \noexpand\eql@tags@set@label{#1}}%
927 }
```

TODO: describe

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

TODO: describe

```
932 \def\eql@tags@addblock@label#1{%
933   \eql@appendexpand\eql@tags@container@block{%
934     \noexpand\eql@tags@set@label{#1}}%
935 }
```

TODO: describe

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

Tags. **TODO:** describe

`\eql@tag@default`

```
940 \protected\def\eql@tag@default{%
941   \eql@warn@here\tag
942   \eql@tag@gobble
943 }
944 \let\tag\eql@tag@default
```

`\eql@tag@gobble`

```
945 \def\eql@tag@gobble{%
946   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

TODO: describe

```
947 \protected\def\eql@tag{%
948   \eql@ampprotecttwo\eql@teststaropt@tight
```

```

949     {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
950     \eql@testopt@default
951 }

```

\eql@tags@add@tagref

```

952 \def\eql@tags@add@tagref[#1]#2{%
953   \def\eql@tmp{#1}%
954   \ifx\eql@tmp\eql@testopt@default\else
955     \eql@tags@add@ref{#1}%
956   \fi
957   \eql@tags@add@tag{#2}%
958 }

```

TODO: describe

```

959 \def\eql@tags@set@tag#1{%
960   \ifdefined\eql@tags@warn
961     \ifdefined\eql@tags@tag
962       \eql@warn@tag@multiple
963     \fi
964   \fi
965   \def\eql@tags@tag{#1}%
966 }

```

TODO: describe

```

967 \def\eql@tags@set@ref#1{%
968   \ifdefined\eql@tags@warn
969     \ifdefined\eql@tags@ref
970       \eql@warn@ref@multiple
971     \fi
972   \fi
973   \def\eql@tags@ref{#1}%
974 }

```

TODO: describe

```

975 \def\eql@tags@add@tag#1{%
976   \ifdefined\eql@tags@autotag
977     \eql@tags@autoenable
978   \fi
979   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
980   \global\eql@appendmacro\eql@tags@container\eql@tmp
981 }

```

TODO: describe

```

982 \def\eql@tags@add@ref#1{%
983   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
984   \global\eql@appendmacro\eql@tags@container\eql@tmp
985 }

```

tags@add@tagform@off

```

986 \def\eql@tags@add@tagform@off{%
987   \global\eql@append\eql@tags@container{\let\eql@tags@tagform\@firstofone}%
988 }

```

TODO: describe

```

989 \def\eql@tags@addblock@tag#1{%

```

```

990 \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
991 \eql@appendmacro\eql@tags@container@block\eql@tmp
992 }

```

TODO: describe

```

993 \def\eql@tags@addblock@ref#1{%
994 \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
995 \eql@appendmacro\eql@tags@container@block\eql@tmp
996 }

```

TODO: describe

```

997 \def\eql@tags@addblock@tagform@off{%
998 \eql@append\eql@tags@container@block{\let\eql@tags@tagform\@firstofone}%
999 }

```

Raise Tags.

\raisetag

```

1000 \def\eql@raisetag@default{%
1001 \eql@warn@here\raisetag
1002 \eql@raisetag@gobble
1003 }

1004 \def\eql@raisetag@gobble{%
1005 \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble
1006 }

```

TODO: describe

```

1007 \eql@amsmath@let\raisetag\eql@raisetag@default

1008 \def\eql@raisetag{%
1009 \eql@ampprotecttwo\eql@ifstar@tight\eql@tags@add@raiseshift\eql@raisetag@test
1010 }

1011 \def\eql@raisetag@test#1{%
1012 \def\eql@tmpa{#1}\def\eql@tmpb{!}%
1013 \ifx\eql@tmpa\eql@tmpb
1014 \eql@tags@add@forceraise
1015 \else
1016 \eql@tags@add@raisesmash{#1}%
1017 \fi
1018 }

1019 \def\eql@tags@add@raiseshift#1{%
1020 \global\eql@appendexpand\eql@tags@container{%
1021 \advance\eql@tagpos@shift@the\glueexpr#1\relax\relax}%
1022 }

1023 \def\eql@tags@add@raisesmash#1{%
1024 \dimen@glueexpr#1\relax
1025 \ifdim\dimen@<\z@
1026 \global\eql@appendexpand\eql@tags@container{%
1027 \advance\eql@tagpos@smashdown@the\dimen@\relax}%
1028 \else
1029 \global\eql@appendexpand\eql@tags@container{%
1030 \advance\eql@tagpos@smashup@the\dimen@\relax}%
1031 \fi
1032 }

```



```

1033 \def\eql@tags@add@forceraise{%
1034   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
1035 }

```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```

1036 \def\eql@numbering@settools{%
1037   \let\label\eql@label
1038   \let\tag\eql@tag
1039   \let\raisetag\eql@raisetag
1040   \let\numberhere\eql@numberhere
1041   \let\numbernext\eql@numbernext
1042 }

```

TODO: not necessary anymore

```

1043 \def\eql@numbering@settools@gobble{%
1044   \let\label\eql@label@gobble
1045   \let\tag\eql@tag@gobble
1046   \let\raisetag\eql@raisetag@gobble
1047   \let\numberhere\relax
1048   \let\numbernext\relax
1049 }

```

```

1050 \def\eql@numbering@autoblock{%
1051   \begingroup
1052     \let\eql@tags@warn\eql@false
1053     \eql@tags@container@block
1054     \ifdefined\eql@tags@autolabel
1055       \ifdefined\eql@tags@label
1056         \global\@eqnswtrue
1057       \fi
1058     \fi
1059     \ifdefined\eql@tags@autotag
1060       \ifdefined\eql@tags@tag
1061         \global\@eqnswtrue
1062       \fi
1063     \fi
1064   \endgroup
1065 }

```

```

1066 \def\eql@numbering@warnunused{%
1067   \ifdefined\eql@tags@label
1068     \eql@warn@label@unused
1069   \fi
1070   \ifdefined\eql@tags@name
1071     \eql@warn@name@unused
1072   \fi
1073   \ifdefined\eql@tags@tag
1074     \eql@warn@tag@unused
1075   \fi
1076   \ifdefined\eql@tags@erf
1077     \eql@warn@ref@unused
1078   \fi

```

1079 }

Single Line. TODO: describe

```
1080 \def\eql@numbering@single@init{%
1081   \let\eql@numbering@multi\eql@false
1082   \eql@numbering@settools
1083   \eql@numbering@eqnswinit
1084   \eql@numbering@autoblock
1085   \global\let\eql@tags@container\eql@tags@container@block
1086   \let\eql@tags@warn\eql@true
1087 }

1088 \def\eql@numbering@single@eval{%
1089   \ifnum\eql@tagpos@row@=\m@ne
1090     \eqnswfalse
1091   \fi
1092 }
```

Multi-Line Measuring Pass. TODO: describe

```
1093 \def\eql@numbering@measure@init{%
1094   \eql@numbering@settools
1095   \ifdefined\eql@numbering@multi\else
1096     \eql@numbering@eqnswinit
1097     \eql@numbering@autoblock
1098   \fi
1099   \global\let\eql@tags@container\eql@tags@container@block
1100   \let\eql@tags@warn\eql@true
1101 }
```

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

```
1102 \def\eql@numbering@measure@line@begin{%
1103   \ifdefined\eql@numbering@multi
1104     \global\eql@numbering@eqnswinit
1105   \fi
1106 }
```

TODO: describe

```
1107 \def\eql@numbering@measure@blocktag{%
1108   \ifdefined\eql@numbering@multi
1109     \eqnswfalse
1110   \else
1111     \ifnum\eql@tagpos@row@=\m@ne
1112       \eqnswfalse
1113     \fi
1114     \ifnum\eql@totalrows@=\z@
1115       \eqnswfalse
1116     \fi
1117   \fi
1118 }
```

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!

```

1119 \def\eql@numbering@print@init{%
1120   \let\eql@tags@warn\eql@false
1121   \ifdefined\eql@numbering@multi
1122     \eql@numbering@settools
1123     \global\let\eql@tags@container\eql@tags@container@block
1124   \else
1125     \let\eql@tags@container@block\eql@tags@container
1126     \eql@numbering@settools@gobble
1127   \fi
1128 }

```

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

```

1129 \def\eql@numbering@print@block@begin{%
1130   \ifdefined\eql@numbering@multi\else
1131     \ifnum\eql@tagpos@row@>\z@
1132       \eql@tags@makeblockanchor
1133       \global\eql@appendexpand\eql@tags@container@block{%
1134         \def\noexpand\eql@tags@anchor{%
1135           \unexpanded\expandafter{\eql@tags@anchor}}}%
1136     \fi
1137   \fi
1138   \ifdefined\eql@numbering@subeq@use
1139     \eql@tags@printsubeqlabel
1140   \fi
1141 }

```

TODO: describe

```

1142 \def\eql@numbering@print@line@begin{%
1143   \ifdefined\eql@numbering@multi
1144     \global\eql@numbering@eqnswinit
1145   \fi
1146 }

```

TODO: describe

```

1147 \def\eql@numbering@print@line@eval{%
1148   \ifdefined\eql@numbering@multi
1149     \if@eqnsw
1150       \eql@tags@container
1151     \fi
1152   \else
1153     \ifnum\eql@tagpos@row@=\eql@row@
1154       \@eqnswtrue
1155       \eql@tags@container@block
1156     \else
1157       \@eqnswfalse
1158     \fi
1159   \fi
1160 }

```

5.5 Positioning

TODO: describe

```

1161 \def\eql@tagpos@single@eval{%
1162   \if@eqnsw
1163     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1164     \ifnum\eql@tagpos@row@>\@ne

```

```

1165 \eql@tagpos@row@\@ne
1166 \fi
1167 \ifdefined\eql@tagpos@doconvert
1168 \let\eql@tagpos@continuous\eql@true
1169 \fi
1170 \ifdefined\eql@tagpos@continuous
1171 \eql@tagpos@single@eval@continuous
1172 \fi
1173 \else
1174 \eql@tagpos@row@\z@
1175 \fi
1176 \eql@tagpos@prevrow@\z@
1177 \eql@tagpos@headroom@\z@
1178 \eql@tagpos@footroom@\z@
1179 }

```

TODO: describe

```

1180 \def\eql@tagpos@single@eval@continuous{%
1181 \ifnum\eql@tagpos@row@>\z@
1182 \eql@tagpos@target@\eql@tagpos@shift@
1183 \else
1184 \eql@tagpos@target@\dimexpr\eql@line@height@
1185 -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1186 \fi
1187 \eql@tagpos@row@\@ne
1188 \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1189 \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1190 \eql@tagpos@target@\z@
1191 \fi
1192 }

```

TODO: describe

```

1193 \def\eql@tagpos@adjust@eval{%
1194 \if@eqnsw
1195 \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1196 \ifnum\eql@tagpos@row@>\eql@totalrows@
1197 \eql@tagpos@row@\eql@totalrows@
1198 \fi
1199 \ifdefined\eql@tagpos@doconvert
1200 \let\eql@tagpos@continuous\eql@true
1201 \fi
1202 \ifdefined\eql@tagpos@continuous
1203 \ifnum\eql@tagpos@row@>\z@
1204 \eql@tagpos@adjust@eval@convert
1205 \fi
1206 \eql@tagpos@adjust@eval@continuous
1207 \fi
1208 \else
1209 \eql@tagpos@row@\z@
1210 \eql@tagpos@prevrow@\z@
1211 \fi
1212 }

```

TODO: describe

```

1213 \def\eql@tagpos@adjust@eval@convert{%
1214 \eql@tagpos@current@\z@
1215 \eql@dimensions@for{%
1216 \ifnum\eql@row@<\eql@tagpos@row@

```

```

1217     \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1218     +\eql@line@height@+\eql@line@depth@\relax
1219   \fi
1220   \ifnum\eql@row@=\eql@tagpos@row@
1221     \advance\eql@tagpos@current@\dimexpr\eql@line@interline@
1222     +\eql@line@height@-\eql@tagheight@block@\relax
1223   \fi
1224 }%
1225 }

```

TODO: describe

```

1226 \def\eql@tagpos@adjust@eval@continuous{%
1227   \dimen@\dimexpr\eql@tagpos@current@-\eql@tagpos@shift@\relax
1228   \eql@tagpos@row@\eql@totalrows@
1229   \eql@tagpos@prevrow@\z@
1230   \eql@tagpos@headroom@\z@
1231   \eql@tagpos@footroom@\z@
1232   \eql@dimensions@for{%
1233     \ifnum\eql@tagpos@row@=\eql@totalrows@
1234       \eql@tagpos@headroom@\eql@line@interline@
1235       \eql@tagpos@target@\dimexpr\eql@line@interline@
1236       +\eql@line@height@-\dimen@-\eql@tagheight@block@\relax
1237     \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1238       \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1239       \advance\dimen@\eql@tagpos@target@
1240       \eql@tagpos@target@\z@
1241     \fi
1242     \ifdim\eql@tagpos@target@>%
1243       \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1244       \eql@tagpos@row@\eql@row@
1245       \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1246     \fi
1247     \advance\dimen@-\dimexpr\eql@line@interline@
1248     +\eql@line@depth@+\eql@line@height@\relax
1249   \fi
1250   \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1251     \eql@tagpos@footroom@\eql@line@interline@
1252   \fi
1253 }%
1254 }

```

TODO: describe

```

1255 \def\eql@tagpos@print@line@eval{%
1256   \ifdefined\eql@tagpos@continuous
1257     \eql@tagpos@print@line@eval@continuous
1258   \else
1259     \eql@tagpos@print@line@eval@discrete
1260   \fi
1261 }

```

TODO: describe

```

1262 \def\eql@tagpos@print@line@eval@continuous{%
1263   \if@eqnsw
1264     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1265     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1266     \eql@tagpos@plain@\eql@tagpos@target@
1267     \@tempdima\dimexpr\eql@line@height@+\eql@tagpos@headroom@
1268     -\ht\eql@tagbox@\relax

```

```

1269 \@tempdimb\dimexpr-\eql@line@depth@-\eql@tagpos@footroom@
1270 +\dp\eql@tagbox@\relax
1271 \ifnum\eql@row@=\@ne
1272 \@tempdima.5\maxdimen
1273 \fi
1274 \ifnum\eql@row@=\eql@totalrows@
1275 \@tempdimb-.5\maxdimen
1276 \fi
1277 \ifdim\eql@tagpos@plain@>\@tempdima
1278 \ifdim\eql@tagpos@plain@>\@tempdimb
1279 \ifdim\@tempdima>\@tempdimb
1280 \eql@tagpos@plain@\@tempdima
1281 \else
1282 \eql@tagpos@plain@\@tempdimb
1283 \fi
1284 \fi
1285 \else
1286 \ifdim\eql@tagpos@plain@<\@tempdimb
1287 \ifdim\@tempdima>\@tempdimb
1288 \eql@tagpos@plain@\@tempdimb
1289 \else
1290 \eql@tagpos@plain@\@tempdima
1291 \fi
1292 \fi
1293 \fi
1294 \ifnum\eql@tagpos@prevrow@>\z@
1295 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1296 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1297 \eql@tagpos@raised@\eql@tagpos@plain@
1298 \let\eql@tagpos@reserve\eql@false
1299 \fi
1300 \else
1301 \ifdim\eql@tagpos@target@>%
1302 \ifdefined\eql@tagsleft-1sp\relax\else\z@\fi
1303 \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1304 \ifdim\eql@tagpos@raised@>\eql@tagpos@plain@\else
1305 \eql@tagpos@raised@\eql@tagpos@plain@
1306 \let\eql@tagpos@reserve\eql@false
1307 \fi
1308 \else
1309 \eql@tagpos@raised@\dimexpr-\eql@line@depth@
1310 -\ht\eql@tagbox@\relax
1311 \ifdim\eql@tagpos@raised@<\eql@tagpos@plain@\else
1312 \eql@tagpos@raised@\eql@tagpos@plain@
1313 \let\eql@tagpos@reserve\eql@false
1314 \fi
1315 \fi
1316 \fi
1317 \else
1318 \ifnum\eql@tagpos@prevrow@=\eql@row@
1319 \eql@tagwidth@\eql@tagwidth@block@
1320 \else
1321 \let\eql@tagpos@reserve\eql@false
1322 \fi
1323 \fi
1324 }

```

TODO: describe

```

1325 \def\eq@tagpos@print@line@eval@discrete{%
1326   \if@eqnsw
1327     \ht\eq@tagbox@dimexpr\ht\eq@tagbox@-\eq@tagpos@smashup@relax
1328     \dp\eq@tagbox@dimexpr\dp\eq@tagbox@-\eq@tagpos@smashdown@relax
1329     \eq@tagpos@plain@eq@tagpos@shift@
1330     \eq@tagpos@headroom@z@
1331     \eq@tagpos@footroom@z@
1332     \ifdim\eq@tagpos@shift@>%
1333       \ifdefined\eq@tagsleft-1sprelaxelsez@fi
1334       \eq@tagpos@raised@dimexpr\eq@line@height@+\dp\eq@tagbox@relax
1335     else
1336       \eq@tagpos@raised@dimexpr-\eq@line@depth@-\ht\eq@tagbox@relax
1337     fi
1338   \else
1339     \let\eq@tagpos@reserve\eq@false
1340   fi
1341 }

```

TODO: describe

```

1342 \def\eq@tagpos@print@line@end{%
1343   \ifdefined\eq@tagpos@continuous
1344     \ifnum\eq@tagpos@prevrow=\eq@row@
1345       \ifdefined\eq@tagpos@reserve
1346         \global\eq@appendexpand\eq@tags@container@block{%
1347           \advance\eq@tagpos@headroom@the\dimexpr\eq@line@height@
1348             +\eq@line@depth@relax}%
1349         \eq@displaybreak@star\M
1350       fi
1351     fi
1352   fi
1353 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1354 \let\eq@SK@loaded\eq@false
1355 \let\eq@SK@label\@gobble
1356 \let\eq@SK@clearlabel\@empty
1357 \let\eq@SK@setlabel\@gobble
1358 \let\eq@SK@printlabel@right\@empty
1359 \let\eq@SK@printlabel@left\@empty
1360 \let\eq@SK@printlabel@line\@empty
1361 \def\eq@label@clean{\eq@label@org}
1362 \AddToHook{package/showkeys/after}{
1363   \let\eq@SK@loaded\eq@true
1364   \def\eq@SK@label#1{\SK@SK@label#1}
1365   \def\eq@SK@clearlabel{\let\eq@SK@lab\relax}
1366   \eq@SK@clearlabel
1367   \def\eq@SK@@label#1>#2\SK@{%
1368     \def\eq@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1369   }
1370   \def\eq@SK@setlabel#1{\SK@\eq@SK@@label#1}
1371   \def\eq@SK@printlabel@right{%
1372     \ifx\eq@SK@lab\relaxelse
1373       \rlap{\kern\marginparsep\eq@SK@lab}%
1374     \eq@SK@clearlabel

```

```

1375 \fi
1376 }
1377 \def\eq@SK@printlabel@left{%
1378 \ifx\eq@SK@lab\relax\else
1379 \llap{\eq@SK@lab\kern\marginparsep}%
1380 \eq@SK@clearlabel
1381 \fi
1382 }
1383 \def\eq@SK@printlabel@line{%
1384 \ifx\eq@SK@lab\relax\else
1385 \dimen@ \prevdepth
1386 \nointerlineskip
1387 \ifdefined\eq@tags@left
1388 \llap{%
1389 \eq@SK@lab
1390 \kern\marginparsep
1391 }%
1392 \eq@SK@clearlabel
1393 \else
1394 \rlap{%
1395 \dimen@ \displaywidth
1396 \advance\dimen@ \marginparsep
1397 \kern\dimen@
1398 \eq@SK@lab
1399 }%
1400 \fi
1401 \eq@SK@clearlabel
1402 \prevdepth\dimen@
1403 \fi
1404 }
1405 \let\eq@label@org\label
1406 \def\eq@label@clean{\let\SK@\@gobbletwo\eq@label@org}
1407 }

```

Labels.

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

```

1408 \def\eq@composetag@label{%
1409 \eq@SK@clearlabel
1410 \ifdefined\eq@tags@label
1411 \eq@SK@setlabel\eq@tags@label
1412 \ifdefined\eq@tags@name
1413 \let\@currentlabelname\eq@tags@name
1414 \else
1415 \let\@currentlabelname\eq@tags@name@generic
1416 \fi
1417 \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1418 \fi
1419 }

```

TODO: describe

```

1420 \def\eq@tags@printsubeq@label{%
1421 \eq@tags@container@parent
1422 \ifdefined\eq@tags@label
1423 \eq@tags@makeblockanchor
1424 \eq@SK@setlabel\eq@tags@label
1425 \begingroup

```



```

1426     \def\@currentcounter{equation}%
1427     \eq@tags@anchor
1428     \let\@currentlabelname\eq@tags@name@generic
1429     \protected@edef\@currentlabel{\p@equation\theparentequation}%
1430     \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1431     \endgroup
1432     \eq@SK@printlabel@line
1433   \fi
1434 }

```

Hyperref Anchors. **TODO:** describe

```

1435 \let\eq@Hy@anchor\@gobble
1436 \AddToHook{package/hyperref/after}{
1437   \def\eq@Hy@anchor#1{%
1438     \Hy@raisedlink{\hyper@anchor{#1}}%
1439   }%
1440 }

```

TODO: describe

```

1441 \def\eq@tags@makeblockanchor{%
1442   \eq@tags@glabel@step
1443   \eq@Hy@anchor\eq@tags@glabel
1444   \edef\eq@tags@anchor{%
1445     \def\noexpand\thepage{\thepage}%
1446     \def\noexpand\@currentHref{\eq@tags@glabel}%
1447   }%
1448 }

```

TODO: describe

ql@composetag@anchor

```

1449 \def\eq@composetag@anchor{%
1450   \ifdefined\eq@tags@tag
1451     \def\@currentcounter{equation}%
1452     \ifdefined\eq@tags@ref
1453       \let\@currentlabel\eq@tags@ref
1454     \else
1455       \protected@edef\@currentlabel{\p@equation\eq@tags@tag}%
1456     \fi
1457     \eq@tags@glabel@step
1458     \edef\@currentHref{\eq@tags@glabel}%
1459     \eq@Hy@anchor\@currentHref
1460   \else
1461     \refstepcounter{equation}%
1462     \protected@edef\eq@tags@tag{\theequation}%
1463   \fi
1464   \eq@tags@anchor
1465 }

```

Tag Layout. **TODO:** describe

```

1466 \def\eq@tags@taglayout@set@direct#1{%
1467   \def\eq@tags@taglayout##1{#1}%
1468 }
1469 \def\eq@tags@taglayout@set#1{%

```

```

1470 \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1471 }

```

TODO: describe

```

1472 \def\eql@tags@tagform@set@direct#1{%
1473   \def\eql@tags@tagform##1{#1}%
1474 }
1475 \def\eql@tags@tagform@set#1#2#3{%
1476   \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@italiccorr#3}%
1477 }

1478 \eql@tags@taglayout@set{#1}
1479 \eql@tags@tagform@set({#1})
1480 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1481 \protected\def\tagform{\eql@tags@tagform}
1482 \protected\def\tagbox{\eql@tags@taglayout}
1483 \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:

```

1484 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1485 \def\eql@composetag@tag{%
1486   \eql@tagging@tagbegin
1487   \eql@tags@frame@cmd{%
1488     \eql@tags@taglayout{%
1489       \eql@tags@tagform\eql@tags@tag
1490       \eql@tagging@tagsave
1491     }%
1492   }%
1493   \eql@tagging@tagend
1494 }

```

5.7 Tag Composition

TODO: describe

```

1495 \def\eql@composetag@measure{%
1496   \ifdefined\eql@tags@tag\else
1497     \stepcounter{equation}%
1498     \let\eql@tags@tag\theequation
1499   \fi
1500   \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1501   \ifdefined\eql@numbering@multi
1502     \global\let\eql@tags@container\eql@tags@container@clear
1503   \fi
1504 }

```

TODO: describe

```

1505 \def\eql@composetag@print{%
1506   \eql@composetag@anchor
1507   \eql@composetag@label
1508   \ifdefined\eql@tagsleft

```

```

1509 \eq@SK@printlabel@left
1510 \eq@composetag@tag
1511 \else
1512 \eq@composetag@tag
1513 \eq@SK@printlabel@right
1514 \fi
1515 \global\let\eq@tags@container\eq@tags@container@clear
1516 }

```

TODO: describe

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

```

1517 \def\eq@tagbox@make#1{%
1518 \setbox\eq@tagbox@\hbox{\eq@strut@tag\@lign#1}%
1519 \eq@tagwidth@\wd\eq@tagbox@
1520 \ifdim\eq@tagwidth@<\eq@tagwidthmin@
1521 \eq@tagwidth@\eq@tagwidthmin@
1522 \fi
1523 \advance\eq@tagwidth@\eq@tagsepmin@
1524 }

```

TODO: describe

```

1525 \def\eq@tagbox@print@adjustheadroom{%
1526 \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1527 \ifdim\dimen@>\z@
1528 \ifdim\dimen@>\eq@tagpos@headroom@
1529 \ht\eq@tagbox@\dimexpr\ht\eq@tagbox@-\eq@tagpos@headroom@\relax
1530 \else
1531 \ht\eq@tagbox@\dimexpr\eq@line@height@-\eq@tagpos@current@\relax
1532 \fi
1533 \fi
1534 }

```

TODO: describe

```

1535 \def\eq@tagbox@print@adjustfootroom{%
1536 \dimen@\dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax
1537 \ifdim\dimen@>\z@
1538 \ifdim\dimen@>\eq@tagpos@footroom@
1539 \dp\eq@tagbox@\dimexpr\dp\eq@tagbox@-\eq@tagpos@footroom@\relax
1540 \else
1541 \dp\eq@tagbox@\dimexpr\eq@line@depth@+\eq@tagpos@current@\relax
1542 \fi
1543 \fi
1544 }

```

TODO: describe

```

1545 \def\eq@tagbox@print@extendabove{%
1546 \dimen@\dimexpr\ht\eq@tagbox@+\eq@tagpos@current@-\eq@line@height@\relax
1547 \ifdim\dimen@>\z@
1548 \global\eq@appendexpand\eq@display@container{%
1549 \eq@display@aboveextend@the\dimen@\relax}%
1550 \fi
1551 }

```

TODO: describe

```

1552 \def\eq@tagbox@print@extendbelow{%
1553 \dimen@\dimexpr\dp\eq@tagbox@-\eq@tagpos@current@-\eq@line@depth@\relax

```

```

1554 \ifdim\dimen@>\z@
1555   \global\eql@appendexpand\eql@display@container{%
1556     \eql@display@belowextend@ \the\dimexpr\dimen@ \relax}%
1557 \fi
1558 }

```

TODO: describe

```

1559 \def\eql@tagbox@print@prepare{%
1560   \ifdefined\eql@tagpos@reserve
1561     \eql@tagpos@current@\eql@tagpos@plain@
1562   \else
1563     \eql@tagpos@current@\eql@tagpos@raised@
1564   \fi
1565   \ifdim\eql@tagpos@headroom@>\z@
1566     \eql@tagbox@print@adjustheadroom
1567   \fi
1568   \ifdim\eql@tagpos@footroom@>\z@
1569     \eql@tagbox@print@adjustfootroom
1570   \fi
1571   \ifnum\eql@row@=\@ne
1572     \eql@tagbox@print@extendabove
1573   \fi
1574   \ifnum\eql@row@=\eql@totalrows@
1575     \eql@tagbox@print@extendbelow
1576   \fi
1577 }

```

TODO: describe

```

1578 \def\eql@tagbox@print@tagsright{%
1579   \eql@tagbox@print@prepare
1580   \kern-\wd\eql@tagbox@
1581   \raise\eql@tagpos@current@\box\eql@tagbox@
1582 }

```

TODO: describe

```

1583 \def\eql@tagbox@print@tagsleft{%
1584   \eql@display@firstavail@set\z@
1585   \eql@tagbox@print@prepare
1586   \wd\eql@tagbox@\z@
1587   \raise\eql@tagpos@current@\box\eql@tagbox@
1588 }

```

$\eql@tagbox@print@cell$

```

1589 \def\eql@tagbox@print@cell{%
1590   \eql@tagging@tagaddbox
1591   \ifdefined\eql@tagsleft
1592     \ifdefined\eql@tagpos@reserve
1593       \ifdim\eql@tagwidth@>\dimexpr\eql@line@avail@+\eql@tagfuzz@\relax
1594         \let\eql@tagpos@reserve\eql@false
1595       \fi
1596     \fi
1597     \if@eqnsw
1598       \eql@tagbox@print@tagsleft
1599     \fi
1600     \kern\displaywidth
1601   \else
1602     \kern\displaywidth

```

```

1603 \ifdefined\eq\tagpos@reserve
1604 \ifdim\eq\tagwidth@>%
1605 \dimexpr\displaywidth-\eq\line@width@+\eq\tagfuzz@\relax
1606 \let\eq\tagpos@reserve\eq\false
1607 \fi
1608 \fi
1609 \if@eqnsw
1610 \eq\tagbox@print@tagsright
1611 \fi
1612 \fi
1613 }

```

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:

```

1614 \eq\amsmath@undefine\c@parentequation
1615 \eq\amsmath@undefine\theparentequation
1616 \ifdefined\c@parentequation\else
1617 \newcounter{parentequation}
1618 \fi

```

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

```

1619 \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:

```

1620 \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`:

```

1621 \def\eq\subequations@init{%
1622 \edef\eq\subequations@restorecounter{%
1623 \global\c@equation\the\c@equation\relax}%
1624 \eq\tags@container@block
1625 \ifdefined\eq\tags@tag
1626 \eq\tags@glabel@step
1627 \protected@edef\theparentequation{\eq\tags@glabel}%
1628 \protected@edef\theparentequation{\eq\tags@tag}%
1629 \else

```

```

1630 \advance\c@equation\@ne
1631 \protected@edef\theparentequation{\theequation}%
1632 \ifdefined\theHequation
1633 \protected@edef\theHparentequation{\theHequation}%
1634 \fi
1635 \fi
1636 \global\c@parentequation\c@equation
1637 \global\c@equation\z@
1638 \let\theequation\eq@subequations@template
1639 \def\theHequation{\theHparentequation.\arabic{equation}}%
1640 }

```

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:

```

1641 \def\eq@subequations@close{%
1642 \ifnum\c@equation=\z@
1643 \eq@subequations@restorecounter
1644 \else
1645 \global\c@equation\c@parentequation
1646 \fi
1647 }

```

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@printslabel`

```

1648 \def\eq@subequations@start{%
1649 \let\eq@tags@container@block\eq@tags@container@clear
1650 \eq@nextopt@process{subequations}%
1651 \eq@subequations@init
1652 \eq@tags@glabel@step
1653 \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1654 \eq@Hy@anchor\eq@subequations@currentHref
1655 \edef\eq@subequations@thepage{\thepage}%
1656 \def\@currentcounter{equation}%
1657 \let\@currentHref\eq@subequations@currentHref
1658 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1659 \eq@tags@container@block
1660 \ifdefined\eq@tags@name
1661 \let\@currentlabelname\eq@tags@name
1662 \else
1663 \let\@currentlabelname\eq@tags@name@generic
1664 \fi
1665 \let\eq@subequations@active\eq@true
1666 \ifdefined\eq@tags@label
1667 \eq@SK@label\eq@tags@label
1668 \fi
1669 \ignorespaces
1670 }

```

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:

```

1671 \def\eq@subequations@end{%
1672   \ifnum\c@equation>\z@
1673     \eq@tags@container@block
1674     \ifdefined\eq@tags@label
1675       \begingroup
1676         \def\@currentcounter{equation}%
1677         \let\thepage\eq@subequations@thepage
1678         \let\@currentHref\eq@subequations@currentHref
1679 % \TODO how about tag* ?! also within equations!
1680         \protected@edef\@currentlabel{\p@equation\theparentequation}%
1681         \ifdefined\eq@tags@name
1682           \let\@currentlabelname\eq@tags@name
1683         \else
1684           \let\@currentlabelname\eq@tags@name@generic
1685         \fi
1686         \expandafter\eq@label@clean\expandafter{\eq@tags@label}%
1687       \endgroup
1688     \fi
1689   \fi
1690   \eq@subequations@close
1691 }

```

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

```

1692 \newenvironment{eq@subequations}{%
1693   \eq@verbose@info\eq@verbose@msg@enterenv
1694   \eq@subequations@testall\eq@subequations@start%
1695 }{%
1696   \eq@subequations@end
1697   \ignorespacesafterend
1698   \eq@verbose@info\eq@verbose@msg@leaveenv
1699 }

```

TODO: describe

```

1700 \def\eq@subequations@testall{\eq@parseopt@main\eq@subequations@parse}
1701 \def\eq@subequations@parse{%
1702   \ifx\eq@parseopt@token[%]
1703     \let\eq@parseopt@next\eq@parseopt@opt
1704   \fi
1705   \ifx\eq@parseopt@token\eq@atxi
1706     \let\eq@parseopt@next\eq@parseopt@label
1707   \fi
1708   \ifx\eq@parseopt@token\eq@atxii
1709     \let\eq@parseopt@next\eq@parseopt@label
1710   \fi
1711   \ifx\eq@parseopt@token\label
1712     \let\eq@parseopt@next\eq@parseopt@end
1713   \fi
1714 }

```

6.3 Subequation Scheme

TODO: describe

```

1715 \def\eq@numbering@subeq@init{%
1716   \let\eq@save@theequation\theequation

```

```

1717 \let\eql@save@theHequation\theHequation
1718 \eql@subequations@init
1719 \let\eql@tags@container@parent\eql@tags@container@block
1720 \let\eql@tags@container@block\eql@tags@container@clear
1721 }

```

TODO: describe

```

1722 \def\eql@numbering@subeq@test{%
1723   \ifnum\eql@tagrows@<\tw@
1724     \let\eql@tags@container@block\eql@tags@container@parent
1725     \let\eql@numbering@subeq@use\eql@false
1726     \let\theequation\eql@save@theequation
1727     \let\theHequation\eql@save@theHequation
1728     \eql@subequations@restorecounter
1729   \fi
1730 }

```

TODO: describe

```

1731 % \TODO note must not use setcounter here (when calc is loaded)
1732 \def\eql@numbering@subeq@close{%
1733   \eql@subequations@close
1734 }

```

7 Display Equations Support

TODO: describe

```

1735 \let\eql@display@injectbefore\@undefined
1736 \let\eql@display@injectafter\@undefined
1737 \let\eql@interline@container\@undefined
1738 \def\eql@interline@container@clear{%
1739   \eql@displaybreak@pen@\@MM
1740   \eql@vspaceskip@\z@skip
1741 }

```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```

1742 \interdisplaylinepenalty\@M

```

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

```

1743 \def\eql@getdsp@pen#1{%
1744   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1745 }

```

TODO: allow a displaybreak before equations

```

1746 \protected\def\eql@displaybreak@default{%
1747   \eql@warning{Invalid use of \string\displaybreak}{}%
1748   \eql@teststaropt@loose\@gobble\eql@gobbleopt{}}
1749 \eql@amsmath@after{\let\eql@displaybreak@default\displaybreak}
1750 \eql@amsmath@let\displaybreak\eql@displaybreak@default

```



```

1751 \newcount\eqldisplaybreak@pen@
1752 \newcount\eqldisplaybreak@prepen@
1753 \newcount\eqldisplaybreak@postpen@

```

TODO: describe

```

1754 \protected\def\eqldisplaybreak{%
1755   \relax
1756   \eql@ampprotecttwo\eql@teststaroropt@tight
1757   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1758 }

1759 \def\eqldisplaybreak@star#1{%
1760   \global\eql@appendexpand\eql@interline@container{%
1761     \eqldisplaybreak@pen@\the\numexpr#1\relax\relax}%
1762 }

1763 \def\eqldisplaybreak@level[#1]{%
1764   \ifnum#1<\z@
1765     \global\eql@append\eql@interline@container{\eqldisplaybreak@pen@\@MM}%
1766   \else
1767     \global\eql@appendexpand\eql@interline@container{%
1768       \eqldisplaybreak@pen@-\@getpen{#1}\relax}%
1769   \fi
1770 }

```

TODO: describe

```

1771 \def\eqldisplaybreak@pre#1{%
1772   \ifnum#1<\z@
1773     \eqldisplaybreak@prepen@\@MM
1774   \else
1775     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1776   \fi
1777 }

```

TODO: describe

```

1778 \def\eqldisplaybreak@post#1{%
1779   \ifnum#1<\z@
1780     \eqldisplaybreak@postpen@\@MM
1781   \else
1782     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1783   \fi
1784 }

```

TODO: describe

```

1785 \def\eqldisplaybreak@inter#1{%
1786   \ifnum#1<\z@
1787     \interdisplaylinepenalty\@M
1788   \else
1789     \interdisplaylinepenalty\eql@getdsp@pen{#1}\relax
1790   \fi
1791 }

```

7.2 Explicit Vertical Space

TODO: describe

`eq\@vspaceskip@ (skip)`

```
1792 \newskip\eq\@vspaceskip@

1793 \let\eq\@vspace@org\@vspace
1794 \def\eq\@vspace{%
1795   \ifvmode
1796     \expandafter\eq\@vspace@immediate
1797   \else
1798     \expandafter\eq\@vspace@line
1799   \fi
1800 }
```

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

```
1801 \def\eq\@vspace@line{%
1802   \eq\@ifstar@loose\eq\@vspace@addfixedbefore\eq\@vspace@add
1803 }
1804 \def\eq\@vspace@add#1{%
1805   \global\eq\@appendexpand\eq\@interline@container{%
1806     \advance\eq\@vspaceskip@\the\glueexpr#1\relax\relax}%
1807 \def\eq\@vspace@addfixedbefore#1{%
1808   \global\eq\@appendexpand\eq\@interline@container{%
1809     \noexpand\eq\@append\noexpand\eq\@display@injectbefore{%
1810       \skip@\the\glueexpr#1\relax\relax
1811       \penalty\@M
1812       \vskip\skip@
1813       \global\advance\eq\@line@interline@\skip@
1814     }%
1815   }%
1816 }
1817 \def\eq\@vspace@addfixedafter#1{%
1818   \global\eq\@appendexpand\eq\@interline@container{%
1819     \noexpand\eq\@append\noexpand\eq\@display@injectafter{%
1820       \dimen@\prevdepth
1821       \hrule\@height\z@
1822       \skip@\the\glueexpr#1\relax\relax
1823       \penalty\@M
1824       \vskip\skip@
1825       \global\advance\eq\@line@interline@\skip@
1826       \prevdepth\dimen@
1827     }%
1828   }%
1829 }
```

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

```
1830 \def\eq\@vspace@immediate{%
1831   \noalign\bgroup
1832     \eq\@ifstar@loose\eq\@vspace@fixed\eq\@vspace@discardable
1833 }
1834 \def\eq\@vspace@fixed#1{%
1835   \skip@\glueexpr#1\relax
1836   \ifnum\eq\@row@=\@one
1837     \global\eq\@appendexpand\eq\@display@container{%
1838       \advance\eq\@abovespace@\the\skip@\relax}%
1839   \else\ifnum\eq\@row@>\eq\@totalrows@
1840     \global\eq\@appendexpand\eq\@display@container{%
1841       \advance\eq\@belowspace@\the\skip@\relax}%
1842   \fi
1843 }
```

```

1842     \else
1843       \dimen@\prevdepth
1844       \hrule\@height\z@
1845       \penalty\@M
1846       \vskip\skip@
1847       \global\advance\eql@line@interline@\skip@
1848       \prevdepth\dimen@
1849     \fi\fi
1850   \egroup
1851 }
1852 \def\eql@vspace@discardable#1{%
1853   \skip@\glueexpr#1\relax
1854   \ifnum\eql@row@=\@ne
1855     \global\eql@appendexpand\eql@display@container{%
1856       \advance\eql@abovespace@\the\skip@\relax}%
1857   \else\ifnum\eql@row@>\eql@totalrows@
1858     \global\eql@appendexpand\eql@display@container{%
1859       \advance\eql@belowspace@\the\skip@\relax}%
1860   \else
1861     \vskip\skip@
1862     \global\advance\eql@line@interline@\skip@
1863   \fi\fi
1864   \egroup
1865 }

```

7.3 Default Vertical Spacing

TODO: describe

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

```

1866 \newbox\eql@strutbox@
1867 \def\eql@strut@depth{.3}
1868 \def\eql@strut{\copy\eql@strutbox@}
1869 \let\eql@strut@cell\eql@strut
1870 \let\eql@strut@tag\eql@strut
1871 \def\eql@strut@make{%
1872   \setbox\eql@strutbox@\hbox{%
1873     \@tempdimb\dimexpr
1874       \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1875     \@tempdima\dimexpr
1876       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1877     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@
1878   }
1879 }
1880 \AtBeginDocument{\eql@strut@make}

```

TODO: describe **TODO:** uses `amsmath \spread@equation`

```

1881 \def\eql@spread@set{%
1882   \ifdefined\eql@spread@reset
1883     \lineskip\normallineskip
1884     \lineskiplimit\normallineskiplimit
1885     \baselineskip\normalbaselineskip
1886   \fi
1887   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1888     +\normalbaselineskip-\baselineskip\relax

```

```

1889 \ifdim\eql@spread@>\z@
1890 \openup\eql@spread@
1891 \ifdefined\spread@equation
1892 \let\spread@equation\@empty
1893 \fi
1894 \fi
1895 }

```

7.4 Entry and Exit

TODO: describe

```

1896 \let\eql@beamerbasecolor@fix\@empty
1897 \AddToHook{package/beamerbasecolor/after}{%
1898 \def\eql@beamerbasecolor@fix{%
1899 \donotcolorouterdisplaymaths
1900 \donotcoloroutermaths
1901 \beamer@setdisplaymathcolor
1902 }%
1903 }

```

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

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

```

1904 \newskip\eql@abovespace@
1905 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1906 \def\eql@display@enter{%
1907 \if@noskipsec\leavevmode\par\fi
1908 \ifvmode
1909 \eql@prevdepth@\prevdepth
1910 \nointerlineskip
1911 \noindent
1912 \else
1913 \eql@prevdepth@\maxdimen
1914 \fi
1915 \eql@beamerbasecolor@fix
1916 }

```

`\eql@display@adjust`

```

1917 \def\eql@display@adjust{%
1918 \ifdefined\eql@display@linewidth
1919 \displaywidth\glueexpr\eql@display@linewidth\relax
1920 \advance\displaywidth-\displayindent
1921 \fi
1922 \ifdefined\eql@display@marginleft
1923 \advance\displaywidth\displayindent
1924 \displayindent\glueexpr\eql@display@marginleft\relax
1925 \advance\displaywidth-\displayindent
1926 \fi
1927 \ifdefined\eql@display@marginright
1928 \advance\displaywidth-\glueexpr\eql@display@marginright\relax
1929 \fi
1930 \ifdim\displaywidth<\z@
1931 \displaywidth\z@
1932 \fi

```

1933 }

\eqldisplay@init

```
1934 \def\eqldisplay@init{%
1935   \let\displaybreak\eqldisplaybreak
1936   \let\eqlvspace@org\vspace
1937   \let\vspace\eqlvspace
1938   \let\eqncontrol\eql@control
1939   \let\eqldisplay@injectbefore\@empty
1940   \let\eqldisplay@injectafter\@empty
1941   \eql@spread@set
1942   \eql@strut@make
1943   \let\eql@frame@cmd\@undefined
1944 }
```

\eqldisplay@print

```
1945 \def\eqldisplay@print{%
1946   \let\eqldisplay@container\@empty
1947   \eqldisplay@firstavail@z@
1948   \eqldisplay@aboveextend@z@
1949   \eqldisplay@belowextend@z@
1950   \eql@punct@term@set
1951   \global\let\eql@interline@container\eql@interline@container@clear
1952 }
```

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

```
1953 \def\eqldisplay@halign@init#1{%
1954   \eql@row@z@
1955   \eql@prevgraf@\prevgraf
1956   \everycr{\noalign{%
1957     \global\advance\eql@row@\@ne
1958     \prevgraf\numexpr\prevgraf+\@ne\relax
1959     #1%
1960   }}%
1961 }
```

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

```
1962 \def\eqldisplay@halign@start{%
1963   \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1964   \ifdim\eql@prevdepth@=\maxdimen\else
1965     \prevdepth\eql@prevdepth@
1966   \fi
1967   \ifdim\prevdepth=-\@m\p@\else
1968     \ifdefined\eqldisplay@height
1969       \skip@\baselineskip
1970       \advance\skip@-\glueexpr\eqldisplay@height\relax
1971       \advance\skip@-\prevdepth\relax
1972       \ifdim\skip@<\lineskiplimit
1973         \skip@\lineskip
1974       \fi
1975       \advance\skip@-\eql@spread@\relax
1976       \vskip\skip@
1977       \nointerlineskip
1978     \else
1979       \vskip-\eql@spread@\relax
```

```

1980   \fi
1981   \fi
1982 }

```

TODO: describe

```

1983 \def\eqldisplay@vspace{%
1984   \advance\abovedisplayskip\eql@abovespace@
1985   \advance\belowdisplayskip\eql@belowspace@
1986 }

```

TODO: describe

```

1987 \def\eqldisplay@vspace@native{%
1988   \advance\abovedisplayskip\eql@abovespace@
1989   \advance\belowdisplayskip\eql@belowspace@
1990   \advance\abovedisplayshortskip\eql@abovespace@
1991   \advance\belowdisplayshortskip\eql@belowspace@
1992 }

```

TODO: describe

```

1993 \def\eqldisplay@penalty{%
1994   \ifnum\eqldisplaybreak@postpen@=\@MM\else
1995     \postdisplaypenalty\eqldisplaybreak@postpen@
1996   \fi
1997   \ifnum\eqldisplaybreak@pen@=\@MM\else
1998     \postdisplaypenalty\eqldisplaybreak@pen@
1999   \fi
2000   \ifnum\eqldisplaybreak@prepen@=\@MM\else
2001     \predisdisplaypenalty\eqldisplaybreak@prepen@
2002   \fi
2003 }

```

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

```

2004 \def\eqldisplay@halign@end{%
2005   \eqldisplay@interline@container
2006   \eqldisplay@injectbefore
2007   \global\eqldisplay@prevgraf@=\numexpr\prevgraf+\@ne\relax
2008   \ifdefined\eqldisplay@depth
2009     \prevdepth\glueexpr\eqldisplay@depth\relax
2010   \fi
2011 }

```

`\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?

```

2012 \def\eqldisplay@close{%
2013   \eqldisplay@container
2014   \ifdim\eqldisplay@firstavail@<\z@
2015     \eqldisplay@firstavail@=\z@
2016   \fi
2017   \eqldisplay@skip@mode@leave@=\z@
2018   \ifdim\eqldisplay@prevdepth@=\maxdimen
2019     \ifdim\predisplaysize=-\maxdimen
2020       \eqldisplay@skip@mode@above@=\eqldisplay@skip@mode@cont@above\relax
2021       \eqldisplay@skip@mode@below@=\eqldisplay@skip@mode@cont@below\relax
2022     \else
2023       \eqldisplay@skip@mode@above@=\z@

```

```

2024 \eqLskip@mode@below@z@
2025 \advance\eqLdisplay@firstavail@\displayindent
2026 \ifdim\eqLdisplay@firstavail@>\predisplaysize
2027 \ifcase\eqLskip@mode@short\relax
2028 \or
2029 \eqLskip@mode@above@\@ne
2030 \or
2031 \eqLskip@mode@above@\@ne
2032 \ifnum\eqL@totalrows@=\@ne
2033 \eqLskip@mode@below@\@ne
2034 \fi
2035 \or
2036 \eqLskip@mode@above@\@ne
2037 \eqLskip@mode@below@\@ne
2038 \fi
2039 \fi
2040 \fi
2041 \else
2042 \ifdim\eqL@prevdepth@=-\@m\p@
2043 \eqLskip@mode@above@\eqLskip@mode@top@above\relax
2044 \eqLskip@mode@below@\eqLskip@mode@top@below\relax
2045 \else
2046 \eqLskip@mode@above@\eqLskip@mode@par@above\relax
2047 \eqLskip@mode@below@\eqLskip@mode@par@below\relax
2048 \fi
2049 \fi
2050 \ifcase\eqLskip@mode@above@
2051 \or\or\or
2052 \predisplaypenaltyz@
2053 \or
2054 \predisplaypenaltyz@
2055 \fi
2056 \ifcase\eqLskip@mode@below@
2057 \or\or\or
2058 \eqLskip@mode@leave@\@ne
2059 \or
2060 \eqLskip@mode@leave@\tw@
2061 \fi
2062 \ifdefined\eqLskip@force@above
2063 \eqLskip@mode@above@\eqLskip@force@above\relax
2064 \fi
2065 \ifdefined\eqLskip@force@below
2066 \eqLskip@mode@below@\eqLskip@force@below\relax
2067 \fi
2068 \ifdefined\eqLskip@force@leave
2069 \eqLskip@mode@leave@\eqLskip@force@leave\relax
2070 \fi
2071 \ifnum\eqLskip@mode@leave@>z@
2072 \postdisplaypenaltyz@
2073 \fi
2074 \ifcase\eqLskip@mode@above@
2075 \abovedisplayskip\glueexpr\eqLskip@long@above\relax
2076 \or
2077 \abovedisplayskip\glueexpr\eqLskip@short@above\relax
2078 \or
2079 \abovedisplayskip\glueexpr\eqLskip@cont@above\relax
2080 \or
2081 \abovedisplayskip\glueexpr\eqLskip@par@above\relax

```

```

2082 \or
2083   \abovedisplayskip\glueexpr\eq\@skip@top@above\relax
2084 \or
2085   \abovedisplayskip\z@skip
2086 \or
2087   \abovedisplayskip\glueexpr\eq\@skip@med@above\relax
2088 \or
2089   \abovedisplayskip\glueexpr\eq\@skip@custom@above\relax
2090 \fi
2091 \ifcase\eq\@skip@mode@below@
2092   \belowdisplayskip\glueexpr\eq\@skip@long@below\relax
2093 \or
2094   \belowdisplayskip\glueexpr\eq\@skip@short@below\relax
2095 \or
2096   \belowdisplayskip\glueexpr\eq\@skip@cont@below\relax
2097 \or
2098   \belowdisplayskip\glueexpr\eq\@skip@par@below\relax
2099 \or
2100   \belowdisplayskip\glueexpr\eq\@skip@top@below\relax
2101 \or
2102   \belowdisplayskip\z@skip
2103 \or
2104   \belowdisplayskip\glueexpr\eq\@skip@med@below\relax
2105 \or
2106   \belowdisplayskip\glueexpr\eq\@skip@custom@below\relax
2107 \fi
2108 \global\eq\@skip@mode@leave@\eq\@skip@mode@leave@
2109 \eq\@interline@container
2110 \advance\eq\@belowspace@\eq\@vspaceskip@
2111 \eq\@display@penalty
2112 \eq\@display@vspace
2113 \skip@\glueexpr\eq\@skip@tag@above\relax
2114 \ifdim\skip@>\abovedisplayskip
2115   \skip@\abovedisplayskip
2116 \fi
2117 \advance\abovedisplayskip-\eq\@display@aboveextend@\relax
2118 \ifdim\abovedisplayskip<\skip@
2119   \abovedisplayskip\skip@
2120 \fi
2121 \skip@\glueexpr\eq\@skip@tag@below\relax
2122 \ifdim\skip@>\belowdisplayskip
2123   \skip@\belowdisplayskip
2124 \fi
2125 \ifdim\eq\@display@belowextend@>\z@
2126   \advance\belowdisplayskip-\eq\@display@belowextend@\relax
2127   \ifdim\belowdisplayskip<\skip@
2128     \belowdisplayskip\skip@
2129   \fi
2130 \fi
2131 }

```

TODO: describe

```

2132 \def\eq\@display@leave{%
2133   \prevgraf\eq\@prevgraf@
2134   \ifcase\eq\@skip@mode@leave@
2135     \or
2136     \endgraf
2137     \or

```



```

2138 \endgraf
2139 \prevdepth-\@m\p@
2140 \fi
2141 }

```

TODO: describe

```

2142 \def\eqldisplay@nest{%
2143 \let\displaybreak\eqldisplaybreak@default
2144 \let\intertext\eqlintertext@default
2145 \let\vspace\eqlvspace@org
2146 }

```

TODO: describe

```

2147 \def\eqldisplay@restore{%
2148 \let\label\eqllabel@org
2149 \let\tag\eqltag@default
2150 \let\raisetag\eqlraisetag@default
2151 \let\displaybreak\eqldisplaybreak@default
2152 \let\intertext\eqlintertext@default
2153 \let\vspace\eqlvspace@org
2154 }

```

TODO: describe

```

2155 \eqlappend\arrayparboxrestore{%
2156 \eqldisplay@restore
2157 \ifdefined\eql@ampproof@active
2158 \eql@amprevert
2159 \fi
2160 \displayfalse
2161 }

```

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)

```

2162 \def\eql@stack@enable{%
2163 \let\eql@stack@save@equations\eql@stack@save@equations@
2164 \let\eql@stack@save@box\eql@stack@save@box@
2165 }

```

TODO: describe

```

2166 \let\eql@stack@save@equations\eql@stack@enable
2167 \let\eql@stack@save@box\eql@stack@enable
2168 \let\eql@stack@restore\@empty

```

TODO: describe

```

2169 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2170 \def\eql@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

```

TODO: further global variables: global registers: `\eql@nextopt`, `\eql@tags@glabel@` used locally without possibility of change between setting and retrieving:

`\eql@prevgraf@`, `\eql@skip@mode@leave@`, `\eql@shape@lastrow`, `\eql@frame@prevcmd`

TODO: to be reviewed: `\eql@intertext@after`, `\eql@intertext@opt` **TODO:** describe

```

2171 \def\eql@stack@save@equations@{%
2172   \let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2173   \let\eql@stack@cell@container\eql@cell@container
2174   \let\eql@stack@tags@container\eql@tags@container
2175   \let\eql@stack@interline@container\eql@interline@container
2176   \let\eql@stack@block@container\eql@display@container
2177   \let\eql@stack@dimensions@tab\eql@dimensions@tab
2178   \edef\eql@stack@restore{%
2179     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2180     \eql@stack@save\let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2181     \eql@stack@save\let\eql@stack@cell@container\eql@cell@container
2182     \eql@stack@save\let\eql@stack@tags@container\eql@tags@container
2183     \eql@stack@save\let\eql@stack@interline@container\eql@interline@container
2184     \eql@stack@save\let\eql@stack@dimensions@tab\eql@dimensions@tab
2185     \eql@stack@save\let\eql@stack@block@container\eql@display@container
2186     \eql@stack@save@reg\eql@column@
2187     \eql@stack@save@reg\eql@totalcolumns@
2188     \eql@stack@save@reg\eql@line@avail@
2189     \eql@stack@save@reg\eql@line@pos@
2190     \eql@stack@save@reg\eql@line@width@
2191     \eql@stack@save@reg\eql@line@depth@
2192     \eql@stack@save@reg\eql@line@height@
2193     \eql@stack@save@reg\eql@line@prevdepth@
2194     \eql@stack@save@reg\eql@line@interline@
2195     \eql@stack@save@reg\eql@totalheight@
2196     \eql@stack@save@reg\eql@tagwidth@max@
2197     \eql@stack@save@reg\eql@tagpos@row@
2198     \eql@stack@save@reg\eql@row@
2199     \eql@stack@save@reg\eql@tagrows@
2200   }%
2201 }

```

TODO: describe

```

2202 \def\eql@stack@save@box@{%
2203   \let\eql@stack@cell@container\eql@cell@container
2204   \edef\eql@stack@restore{%
2205     \eql@stack@save\let\eql@stack@cell@container\eql@cell@container
2206     \eql@stack@save@reg\eql@row@
2207   }%
2208 }

```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```

2209 \def\eql@measure@init#1#2{%
2210   \eql@dimensions@reset
2211   \let\eql@display@container\@empty
2212   \eql@numbering@measure@init
2213   \eql@row@\z@
2214   \eql@totalheight@\z@
2215   \eql@totalrows@\@M

```

```

2216 \eql@line@prevdepth@-\@m\p@
2217 \eql@line@interline@\z@
2218 \tabskip\z@skip
2219 \everycr{\noalign{%
2220   \global\advance\eql@row@\@ne
2221   #1%
2222 }}%
2223 \eql@punct@term@set
2224 \global\let\eql@interline@container\eql@interline@container@clear
2225 \eql@measure@savestate
2226 \eql@display@halign@letcr{#2}%
2227 }

```

TODO: describe

```

2228 \def\eql@measure@tag{%
2229   \eql@tagwidth@\z@
2230   \ifdefined\eql@numbering@multi
2231     \if@eqnsw
2232       \eql@tags@container
2233       \eql@tagbox@make\eql@composetag@measure
2234       \ifdefined\eql@tagpos@reserve\else
2235         \eql@tagwidth@\z@
2236       \fi
2237     \fi
2238   \fi
2239 }

```

TODO: describe

```

2240 \def\eql@measure@endrow{%
2241   \ifdim\eql@line@prevdepth@=-\@m\p@\else
2242     \dimen@\dimexpr\baselineskip-\eql@line@height@-\eql@line@prevdepth@\relax
2243     \ifdim\dimen@<\lineskiplimit
2244       \dimen@\lineskip
2245     \fi
2246     \advance\eql@line@interline@\dimen@
2247   \fi
2248   \eql@dimensions@endrow
2249   \ifdim\eql@tagwidth@>\eql@tagwidth@max@
2250     \global\eql@tagwidth@max@\eql@tagwidth@
2251   \fi
2252   \ifdim\eql@tagwidth@>\z@
2253     \global\advance\eql@tagrows@\@ne
2254   \fi
2255   \global\advance\eql@totalheight@\dimexpr
2256     \eql@line@interline@+\eql@line@height@+\eql@line@depth@
2257   \global\eql@line@interline@\z@
2258   \global\eql@line@prevdepth@\eql@line@depth@
2259 }

```

TODO: describe

```

2260 \def\eql@measure@close{%
2261   \advance\eql@row@-\tw@
2262   \eql@totalrows@\eql@row@
2263   \ifnum\eql@totalrows@>\z@
2264     \eql@dimensions@get@\@ne
2265     \eql@topheight@\dimexpr\eql@line@height@+\eql@line@interline@\relax
2266     \eql@dimensions@get\eql@totalrows@

```

```

2267 \eql@bottomdepth@eql@line@depth@
2268 \fi
2269 \eql@numbering@measure@blocktag
2270 \begingroup
2271 \eql@tags@container
2272 \if@eqnsw
2273 \eql@tagbox@make\eql@composetag@measure
2274 \ifdefined\eql@tagpos@reserve\else
2275 \eql@tagwidth@z@
2276 \fi
2277 \eql@dimensions@saveblocktag
2278 \else
2279 \eql@dimensions@savenoblocktag
2280 \eql@numbering@warnunused
2281 \fi
2282 \endgroup
2283 \eql@dimensions@get\z@
2284 \eql@measure@restorestate
2285 }

```

measure@restorestate

eql@measure@savestate

```

2286 \let\eql@measure@restorestate\@empty
2287 \def\eql@measure@savestate{%
2288 \begingroup
2289 \def\@elt##1{%
2290 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2291 \global\edef\@gtempa{\cl@ckpt}%
2292 \endgroup
2293 \let\eql@measure@restorestate\@gtempa
2294 }

```

8.2 Line Breaks

TODO: describe

\eql@display@cr

```

2295 \def\eql@display@cr{%
2296 \eql@ampprotect\eql@display@cr@testall\eql@display@cr@process}

```

l@display@cr@testall

TODO: describe

eql@display@cr@parse

```

2297 \def\eql@display@cr@testall{\eql@parseopt@aux\eql@display@cr@parse}
2298 \def\eql@display@cr@parse{%
2299 \ifx\eql@parseopt@token[%
2300 \let\eql@parseopt@next\eql@display@cr@parse@vspace
2301 \fi
2302 \ifx\eql@parseopt@token*%
2303 \let\eql@parseopt@next\eql@display@cr@parse@star
2304 \fi
2305 \ifx\eql@parseopt@token.%
2306 \let\eql@parseopt@next\eql@parseopt@punctpass
2307 \fi
2308 \ifx\eql@parseopt@token,%
2309 \let\eql@parseopt@next\eql@parseopt@punctpass
2310 \fi
2311 \ifx\eql@parseopt@token~%

```

```

2312 \let\eql@parseopt@next\eql@display@cr@parse@cont
2313 \fi
2314 \ifx\eql@parseopt@token'%
2315 \let\eql@parseopt@next\eql@parseopt@punctnext
2316 \fi
2317 \ifx\eql@parseopt@token&%
2318 \let\eql@parseopt@next\eql@parseopt@end
2319 \fi
2320 }
2321 \def\eql@display@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2322 \def\eql@display@cr@parse@star#1{\eql@displaybreak@star\@M\eql@parseopt@peek}
2323 \def\eql@display@cr@parse@cont#1{\numbernext\eqnpunct~\eql@parseopt@peek}

```

l@display@cr@process

```

2324 \def\eql@display@cr@process{%
2325 \eql@display@endline
2326 \cr
2327 \noalign{%
2328 \eql@interline@container
2329 \eql@display@injectbefore
2330 \ifnum\eql@displaybreak@pen@=\@MM
2331 \penalty\interdisplaylinepenalty
2332 \else
2333 \penalty\eql@displaybreak@pen@
2334 \fi
2335 \vskip\eql@vspaceskip@
2336 \global\advance\eql@line@interline@\eql@vspaceskip@
2337 \eql@display@injectafter
2338 \global\let\eql@interline@container\eql@interline@container@clear
2339 }%
2340 }

```

display@halign@letcr

```

2341 \def\eql@display@halign@letcr#1{%
2342 \let\\eql@display@cr
2343 \let\eql@display@endline#1%
2344 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2345 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2346 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2347 \def\eql@intertext@process{%
2348 \eql@display@endline
2349 \cr
2350 \ifmeasuring@
2351 \expandafter\@gobble
2352 \else
2353 \expandafter\eql@intertext@print

```

```

2354 \fi
2355 }

```

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

```

2356 \def\eql@intertext#print#1{%
2357   \noalign{%
2358     \eql@display@halign@end
2359     \let\eql@skip@force@below\z@
2360     \let\eql@skip@force@above\z@
2361     \eql@setkeys{intertext}\eql@intertext@opt
2362     \openup-\eql@spread@
2363     \penalty\postdisplaypenalty
2364     \ifcase\eql@skip@force@below\relax
2365       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2366     \or
2367       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2368     \or
2369       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2370     \or
2371       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2372     \or
2373       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2374     \or
2375       \advance\eql@vspaceskip@\z@skip
2376     \or
2377       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2378     \or
2379       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2380   \fi
2381   \vskip\eql@vspaceskip@
2382   \global\let\eql@interline@container\eql@interline@container@clear
2383   \vbox{%
2384     \@parboxrestore
2385     \ifdim
2386       \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2387     \else
2388       \parshape\@ne
2389       \@totalleftmargin\linewidth
2390     \fi
2391     \noindent
2392     \prevgraf\eql@prevgraf@
2393     \ignorespaces
2394     #1%
2395     \par
2396     \global\eql@prevgraf@\prevgraf
2397   }%
2398   \penalty\predisplaypenalty
2399   \ifcase\eql@skip@force@above\relax
2400     \vskip\glueexpr\eql@skip@long@above\relax
2401   \or
2402     \vskip\glueexpr\eql@skip@short@above\relax
2403   \or
2404     \vskip\glueexpr\eql@skip@cont@above\relax
2405   \or
2406     \vskip\glueexpr\eql@skip@par@above\relax
2407   \or
2408     \vskip\glueexpr\eql@skip@top@above\relax

```

```

2409 \or
2410 \vskip\z@skip
2411 \or
2412 \vskip\glueexpr\eq@skip@med@above\relax
2413 \or
2414 \vskip\glueexpr\eq@skip@custom@above\relax
2415 \fi
2416 % \eq@prevdepth@\maxdimen
2417 \eq@prevdepth@\z@
2418 \eq@display@halign@start
2419 }
2420 }

```

TODO: describe

```

2421 \newenvironment{eq@intertext}{%
2422 \eq@testopt@tight\eq@intertext@{}}%
2423 }{%
2424 \aftergroup\eq@intertext@after
2425 \ignorespacesafterend
2426 }

```

TODO: describe

```

2427 \def\eq@intertext@env{intertext}
2428 \def\eq@intertext@[#1]{%
2429 \global\def\eq@intertext@opt{#1}%
2430 \ifx\@currentvir\eq@intertext@env
2431 \expandafter\eq@scan@env\expandafter\eq@intertext@inject
2432 \else
2433 \expandafter\eq@intertext@process
2434 \fi
2435 }

```

TODO: describe

```

2436 \def\eq@intertext@inject{%
2437 \global\edef\eq@intertext@after{%
2438 \noexpand\eq@intertext@process{%
2439 \ifx\eq@scan@body\eq@scan@body@dump
2440 \eq@scan@body@dump
2441 \else
2442 \noexpand\scantokens{\eq@scan@body@dump}%
2443 \fi
2444 }%
2445 }%
2446 }

```

8.4 Line Marks

TODO: describe

```

2447 \def\eq@markline@pos@below{below}
2448 \def\eq@markline@pos@bottom{bottom}
2449 \def\eq@markline@pos@baseline{baseline}
2450 \let\eq@markline@pos\eq@markline@pos@baseline
2451 \let\eq@markline@shift\z@
2452 \def\eq@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2453 \def\eq@markline@symbol{}

```

TODO: describe

```
2454 \def\eq@markline@select#1{%
2455   \let\eq@markline@shift\z@
2456   \eq@setkeys{markline}{#1}%
2457   \eq@markline@print
2458 }
```

TODO: describe

```
2459 \def\eq@markline@print{%
2460   \dimen@=\dimexpr\eq@markline@shift\relax
2461   \ifx\eq@markline@pos\eq@markline@pos@below
2462     \ifdim\dimen@=\z@\else
2463       \penalty\@M
2464       \vskip-\dimen@
2465     \fi
2466     \nointerlineskip
2467     \penalty\@M
2468     \vbox{\hfill\hbox{\eq@markline@symbol}}%
2469   \else
2470     \ifx\eq@markline@pos\eq@markline@pos@baseline
2471       \advance\dimen@\prevdepth
2472     \fi
2473     \setbox\z@\hbox{\raise\dimen@\hbox{\eq@markline@symbol}}%
2474     \dimen@\prevdepth
2475     \ht\z@\z@
2476     \dp\z@\z@
2477     \nointerlineskip
2478     \penalty\@M
2479     \vbox{\hfill\box\z@}%
2480     \prevdepth\dimen@
2481   \fi
2482 }
```

TODO: describe

```
2483 \def\eq@markline@inject#1{%
2484   \let\eq@markline@push\eq@false
2485   \ifx\eq@markline@pos\eq@markline@pos@below\else
2486     \ifdefined\eq@tagsleft\else
2487       \ifx\eq@equations@main\eq@multi@main
2488         \ifdefined\eq@numbering@multi
2489           \if@eqnsw
2490             \let\eq@markline@push\eq@true
2491           \fi
2492         \else
2493           \ifnum\eq@row@=\eq@tagpos@row@
2494             \let\eq@markline@push\eq@true
2495           \fi
2496         \fi
2497       \else
2498         \if@eqnsw
2499           \let\eq@markline@push\eq@true
2500         \fi
2501       \fi
2502     \fi
2503   \fi
2504   \ifdefined\eq@markline@push
2505     \global\eq@append\eq@interline@container{%
```



```

2506      \eql@append\eql@display@injectbefore{\eql@markline@select{push,#1}}}%
2507  \else
2508      \global\eql@append\eql@interline@container{%
2509      \eql@append\eql@display@injectbefore{\eql@markline@select{#1}}}%
2510  \fi
2511 }

```

TODO: describe

```

2512 \def\eql@markline@amsthm@opt[#1]{\eql@markline@inject{qed,#1}}
2513 \def\eql@markline@amsthm@staropt[#1]{\eql@markline@inject{qed,push,#1}}
2514 \def\eql@markline@amsthm@qed{\eql@teststaropt@tight
2515   \eql@markline@amsthm@staropt\eql@markline@amsthm@opt{}}
2516 \def\eql@markline@amsthm@register#1{\eql@letcs{#1@qed}\eql@markline@amsthm@qed}
2517 \def\eql@markline@amsthm@move#1#2{%
2518   \AddToHook{package/amsthm/after}{%
2519     \eql@letcs{#1@qed}\expandafter\csname#2@qed\endcsname}}

```

9 Column Placement

TODO: describe

9.1 Supporting Definitions

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

```

2520 \newcount\eql@shape@pos@
2521 \newdimen\eql@shape@amount@
2522 \let\eql@shape@lastrow\eql@false

```

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

`\eql@marginleft@min@` (*dimen*)
`\eql@marginright@` (*dimen*)
`\eql@centeroffset@` (*dimen*)

```

2523 \newdimen\eql@marginleft@
2524 \newdimen\eql@marginright@
2525 \newdimen\eql@marginleft@min@
2526 \newdimen\eql@centeroffset@

```

9.2 Shape Schemes

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

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

```

2527 \def\eql@shape@tab@default{default}
2528 \def\eql@shape@tab@left{left}
2529 \def\eql@shape@tab@center{center}
2530 \def\eql@shape@tab@right{right}
2531 \def\eql@shape@tab@first{first}
2532 \def\eql@shape@tab@hanging{hanging}
2533 \def\eql@shape@tab@steps{steps}

```

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

```

2534 \let\eq@shape@tab@def\eq@shape@tab@default
2535 \let\eq@shape@tab@\eq@shape@tab@default
2536 \let\eq@shape@tab@l\eq@shape@tab@left
2537 \let\eq@shape@tab@c\eq@shape@tab@center
2538 \let\eq@shape@tab@r\eq@shape@tab@right
2539 \let\eq@shape@tab@rc\eq@shape@tab@first
2540 \let\eq@shape@tab@indent\eq@shape@tab@first
2541 \let\eq@shape@tab@hang\eq@shape@tab@hanging
2542 \let\eq@shape@tab@lc\eq@shape@tab@hanging
2543 \let\eq@shape@tab@outdent\eq@shape@tab@hanging
2544 \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:

```

2545 \let\eq@shape@mode\eq@shape@tab@default

```

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

```

2546 \def\eq@shape@set#1{%
2547   \ifcsname eq@shape@tab@#1\endcsname
2548     \expandafter\let\expandafter\eq@shape@mode
2549       \csname eq@shape@tab@#1\endcsname
2550   \else
2551     \eq@error{shape '1' unknown: setting to default}%
2552     \let\eq@shape@mode\eq@shape@tab@default
2553   \fi
2554 }

```

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

```

2555 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z\eq@shape@amount@z}
2556 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne\eq@shape@amount@z}
2557 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw\eq@shape@amount@z}
2558 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2559 \def\eq@shape@layoutleft@left{\eq@shape@pos@z\eq@shape@amount@z}
2560 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne\eq@shape@amount@z}
2561 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw\eq@shape@amount@z}
2562 \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):

```

2563 \def\eq@shape@layoutcenter@first{%
2564   \eq@shape@pos@z
2565   \eq@shape@amount@z
2566   \ifnum\eq@totalrows@>\@ne
2567     \ifnum\eq@row@=\@ne
2568       \eq@shape@amount@\eq@indent@
2569     \fi
2570   \fi
2571 }
2572 \def\eq@shape@layoutleft@first{%
2573   \eq@shape@pos@z
2574   \eq@shape@amount@z
2575   \ifnum\eq@totalrows@>\@ne
2576     \ifnum\eq@row@=\@ne
2577       \eq@shape@amount@\eq@indent@

```

```

2578     \fi
2579 \fi
2580 }

```

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:

```

2581 \def\eq@shape@layoutcenter@hanging{%
2582   \eq@shape@pos@z@
2583   \eq@shape@amount@eq@indent@
2584   \ifnum\eq@totalrows@>\@ne
2585     \ifnum\eq@row@=\@ne
2586       \eq@shape@amount@z@
2587     \fi
2588   \fi
2589 }
2590 \def\eq@shape@layoutleft@hanging{%
2591   \eq@shape@pos@z@
2592   \eq@shape@amount@z@
2593   \ifnum\eq@totalrows@>\@ne
2594     \ifnum\eq@row@=\@ne
2595       \eq@shape@amount@-\eq@indent@
2596     \fi
2597   \fi
2598 }

```

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:

```

2599 \def\eq@shape@layoutcenter@steps{%
2600   \eq@shape@amount@z@
2601   \eq@shape@pos@\@ne
2602   \ifnum\eq@totalrows@>\@ne
2603     \ifnum\eq@row@=\@ne
2604       \eq@shape@pos@z@
2605     \fi
2606     \ifnum\eq@row@=\eq@totalrows@
2607       \eq@shape@pos@tw@
2608     \fi
2609   \fi
2610 }
2611 \def\eq@shape@layoutleft@steps{%
2612   \eq@shape@pos@z@
2613   \eq@shape@amount@z@
2614   \ifnum\eq@totalrows@>\@ne
2615     \ifnum\eq@row@=\@ne
2616       \eq@shape@amount@-\eq@indent@
2617     \fi
2618     \ifnum\eq@row@=\eq@totalrows@
2619       \eq@shape@amount@\eq@indent@
2620     \fi
2621   \fi
2622 }

```

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

```

2623 \let\eq@shape@eval\@undefined
2624 \def\eq@shape@select{%
2625   \expandafter\let\expandafter\eq@shape@eval
2626     \csname eq@shape%
2627     @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2628     @\eq@shape@mode\endcsname
2629 }

```

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

`\eq@shape@alignright`

`\eq@shape@aligncenter`

```

2630 \protected\def\eq@shape@alignleft{%
2631   \global\eq@append\eq@cell@container{\eq@shape@pos@z}%
2632   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2633 \protected\def\eq@shape@aligncenter{%
2634   \global\eq@append\eq@cell@container{\eq@shape@pos@ne}%
2635   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2636 \protected\def\eq@shape@alignright{%
2637   \global\eq@append\eq@cell@container{\eq@shape@pos@tw}%
2638   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2639 \def\eq@shape@align@testpar#1{%
2640   \eq@ifstar@tight{#1[\eq@indent@]}%
2641   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2642   {\eq@testopt@tight{#1}\z@}}
2643 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2644 \protected\def\eq@shape@alignamount{%
2645   \eq@ampprotecttwo\eq@ifstar@tight
2646   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2647 \def\eq@shape@alignamount@add#1{%
2648   \global\eq@appendexpand\eq@cell@container{%
2649     \advance\eq@shape@amount@the\glueexpr#1\relax\relax}}
2650 \def\eq@shape@alignamount@set#1{%
2651   \global\eq@appendexpand\eq@cell@container{%
2652     \eq@shape@amount@the\glueexpr#1\relax\relax}}
2653 \def\eq@shape@align@enable{%
2654   \let\shoveleft\eq@shape@alignleft
2655   \let\shovecenter\eq@shape@aligncenter
2656   \let\shoveright\eq@shape@alignright
2657   \let\shoveby\eq@shape@alignamount
2658 }

```

TODO: describe

```

2659 \protected\def\eq@shape@align@default{%
2660   \eq@warn@here{\shove...}%
2661   \eq@ampprotect\eq@shape@align@testpar\eq@gobbleopt}
2662 \protected\def\eq@shape@alignamount@default{%
2663   \eq@warn@here{\shove...}%
2664   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble}
2665 \def\eq@shape@align@disable{%
2666   \let\shoveleft\eq@shape@align@default
2667   \let\shovecenter\eq@shape@align@default
2668   \let\shoveright\eq@shape@align@default
2669   \let\shoveby\eq@shape@alignamount@default
2670 }

```

9.3 Width Data

gwidth@block@ (*dimen*)

```
2671 \newdimen\eql@tagwidth@block@
2672 \newdimen\eql@tagheight@block@
2673 \newdimen\eql@tagdepth@block@
```

\eql@dimensions@tab **TODO:** new

```
2674 \let\eql@dimensions@tab\@empty
```

eql@dimensions@reset

```
2675 \def\eql@dimensions@reset{%
2676   \let\eql@dimensions@tab\@empty
2677   \eql@tagwidth@max@z@
2678   \eql@tagrows@z@
2679 }
```

\eql@dimensions@add

```
2680 \def\eql@dimensions@add#1{%
2681   \global\eql@appendexpand\eql@dimensions@tab{#1}%
2682 }
```

ql@dimensions@addreg

```
2683 \def\eql@dimensions@addreg#1{#1\the#1\relax}
```

@dimensions@startrow

```
2684 \def\eql@dimensions@startrow{%
2685   \eql@dimensions@add{\eql@dimensions@addreg\eql@row@}%
2686 }
```

@dimensions@savecell

```
2687 \def\eql@dimensions@savecell{%
2688   \eql@dimensions@add{%
2689     \eql@dimensions@addreg\eql@shape@pos@
2690     \eql@dimensions@addreg\eql@cellwidth@
2691     \eql@dimensions@addreg\eql@shape@amount@
2692     \noexpand\eql@dimensions@cellcall
2693   }%
2694 }
```

l@dimensions@savesep

```
2695 \def\eql@dimensions@savesep{%
2696   \eql@dimensions@add{\noexpand\eql@dimensions@sepcall}%
2697 }
```

ql@dimensions@endrow

```
2698 \def\eql@dimensions@endrow{%
2699   \eql@dimensions@add{,%
2700     \eql@dimensions@addreg\eql@tagwidth@
2701     \eql@dimensions@addreg\eql@line@height@
```

```

2702 \eqldimensions@addreg\eqlline@depth@
2703 \eqldimensions@addreg\eqlline@interline@
2704 ;}%
2705 }

ensions@saveblocktag

2706 \def\eqldimensions@saveblocktag{%
2707 \eqldimensions@add{\eqrow@0\relax,%
2708 \eqldimensions@tagwidth@block@the\eqldimensions@tagwidth@relax
2709 \eqldimensions@tagheight@block@the\ht\eqldimensions@tagbox@relax
2710 \eqldimensions@tagdepth@block@the\dp\eqldimensions@tagbox@relax
2711 \eqldimensions@addreg\eqldimensions@tagpos@shift@
2712 \let\noexpand\eqldimensions@tagpos@reserve\ifdefined\eqldimensions@tagpos@reserve
2713 \noexpand\eqldimensions@true\else\noexpand\eqldimensions@false\fi
2714 ;}%
2715 \global\eqldimensions@tagwidth@max@\eqldimensions@tagwidth@
2716 \global\eqldimensions@taggrows@\@ne
2717 }

ensions@savenoblocktag

2718 \def\eqldimensions@savenoblocktag{%
2719 \eqldimensions@add{\eqrow@0\relax,;}%
2720 }

\eqldimensions@for

2721 \def\eqldimensions@for#1{%
2722 \def\eqldimensions@forcall{#1}%
2723 \expandafter\eqldimensions@forstep\eqldimensions@tab
2724 }

l@dimensions@forstep

2725 \def\eqldimensions@forstep\eqldimensions@row@#1\relax#2,#3;%
2726 \eqldimensions@row@#1\relax
2727 \ifnum\eqldimensions@row@=\z@else
2728 #3%
2729 \def\eqldimensions@cells{#2}%
2730 \eqldimensions@forcall
2731 \expandafter\eqldimensions@forstep
2732 \fi
2733 }

\eqldimensions@get

2734 \def\eqldimensions@get#1{%
2735 \eqldimensions@row@#1\relax
2736 \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@}%
2737 \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
2738 }

ql@dimensions@getdef

2739 \def\eqldimensions@getdef#1{%
2740 \def\eqldimensions@getparse
2741 ##1\eqldimensions@row@#1\relax##2,##3;##4\@nil{%
2742 ##3%

```

```

2743 \def\eql@dimensions@cells{##2}%
2744 }%
2745 }

```

`\eql@colwidth@tab`

```

2746 \let\eql@colwidth@tab\@empty

```

`\eql@colwidth@get`

```

2747 \def\eql@colwidth@get#1{%
2748 \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
2749 }

```

`\eql@colwidth@save`

```

2750 \def\eql@colwidth@save#1{%
2751 \edef\eql@colwidth@tab{%
2752 \noexpand\or\the#1%
2753 \unexpanded\expandafter{\eql@colwidth@tab}%
2754 }%
2755 }

```

`\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@`:

```

2756 \def\eql@dimensions@calc{%
2757 \eql@column@\z@
2758 \eql@line@pos@\z@
2759 \eql@line@possep@\z@
2760 \eql@line@avail@\eql@totalwidth@
2761 \eql@line@availsep@\eql@intercolumns@
2762 \eql@line@width@\z@
2763 \eql@line@widthsep@\z@
2764 \let\eql@dimensions@cellcall\eql@dimensions@calc@call
2765 \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
2766 \eql@dimensions@cells
2767 }

```

`\eql@dimensions@calc@callsep` Callback for each intercolumn space.

```

2768 \def\eql@dimensions@calc@callsep{%
2769 \advance\eql@line@possep@\@ne
2770 }%

```

`\eql@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: **TODD**: implement an offset for central alignment (global!!)

```

2771 \def\eql@dimensions@calc@call{%
2772 \advance\eql@column@\@ne
2773 \ifnum\eql@totalcolumns@=\@ne
2774 \dimen@\eql@totalwidth@
2775 \else

```

```

2776 \dimen@eql@colwidth@get\eql@column@\relax
2777 \fi
2778 \ifdim\eql@cellwidth@>\z@
2779 \ifdim\eql@line@width@=\z@
2780 \eql@line@avail@\eql@line@pos@
2781 \eql@line@availsep@\eql@line@possep@
2782 \ifcase\eql@shape@pos@
2783 \or
2784 \advance\eql@line@avail@\dimexpr
2785 (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2786 \or
2787 \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
2788 \fi
2789 \advance\eql@line@avail@\eql@shape@amount@
2790 \fi
2791 \eql@line@width@\eql@line@pos@
2792 \eql@line@widthsep@\eql@line@possep@
2793 \ifcase\eql@shape@pos@
2794 \advance\eql@line@width@\eql@cellwidth@
2795 \or
2796 \advance\eql@line@width@\dimexpr
2797 (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
2798 \or
2799 \advance\eql@line@width@\dimen@
2800 \fi
2801 \advance\eql@line@width@\eql@shape@amount@
2802 \fi
2803 \advance\eql@line@pos@\dimen@
2804 }

```

9.4 Best Line Selection

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

```
2805 \let\eql@numbering@best@auto\eql@false
```

g@best@row@ (*counter*)

g@best@space@ (*dimen*)

bering@best@use (*bool*)

```

2806 \newcount\eql@numbering@best@row@
2807 \newdimen\eql@numbering@best@space@
2808 \let\eql@numbering@best@use\eql@false

```

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

```

2809 \def\eql@numbering@best@find{%
2810 \eql@numbering@best@row@\z@
2811 \eql@numbering@best@space@\z@
2812 \eql@dimensions@for{%
2813 \eql@dimensions@calc
2814 \ifdefined\eql@tagsleft
2815 \dimen@\eql@line@avail@
2816 \else
2817 \dimen@\dimexpr\eql@totalwidth@-\eql@line@width@\relax
2818 \fi
2819 \ifdim\dimen@>\eql@numbering@best@space@
2820 \eql@numbering@best@row@\eql@row@
2821 \eql@numbering@best@space@\dimen@
2822 \fi

```



```

2823 }%
2824 \ifnum\eq\@numbering@best@row@>\z@
2825   \eq\@tagpos@row@\eq\@numbering@best@row@
2826   \let\eq\@tagpos@continuous\eq\@false
2827   \eq\@tagpos@prevrow@\z@
2828 \fi
2829 }

```

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

```

2830 \def\eq\@numbering@best@test#1{%
2831   \eq\@dimensions@get#1%
2832   \eq\@dimensions@calc
2833   \ifdefined\eq\@tagsleft
2834     \dimen@\dimexpr\eq\@line@avail@
2835       +\eq\@marginleft@+\eq\@line@availsep@\eq\@colsep@\relax
2836   \else
2837     \dimen@\dimexpr\displaywidth-\eq\@line@width@
2838       -\eq\@marginleft@-\eq\@line@widthsep@\eq\@colsep@\relax
2839   \fi
2840   \ifdim\dimen@<\eq\@tagwidth@block@
2841     \let\eq\@numbering@best@use\eq\@true
2842   \fi
2843 }

```

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

```

2844 \def\eq\@numbering@best@eval{%
2845   \ifdefined\eq\@numbering@best@auto
2846     \ifdefined\eq\@numbering@best@use\else
2847       \ifdefined\eq\@numbering@multi\else
2848         \ifnum\eq\@tagpos@row@>\z@
2849           \eq\@numbering@best@test\eq\@tagpos@row@
2850         \fi
2851         \ifnum\eq\@tagpos@prevrow@>\z@
2852           \eq\@numbering@best@test\eq\@tagpos@prevrow@
2853         \fi
2854       \fi
2855     \fi
2856   \fi
2857   \ifdefined\eq\@numbering@best@use
2858     \eq\@numbering@best@find
2859   \fi
2860 }

```

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.

`\adjust@calc@tagmargin`

```

2861 \def\eq\adjust@calc@tagmargin{%
2862   \ifdefined\eq\@tagmargin@val
2863     \eq\@tagmargin@\glueexpr\eq\@tagmargin@val\relax
2864   \else
2865     \eq\@tagmargin@\eq\@tagwidth@max@
2866   \ifdim\eq\@tagmargin@>\z@

```

```

2867     \advance\eql@tagmargin@-\eql@tagsepmin@
2868     \fi
2869 \fi

2870 \dimen@ \eql@tagrows@ \p@
2871 \ifnum \eql@totalrows@ = \@ne
2872     \ifnum \eql@tagrows@ = \@ne
2873         \advance \dimen@ 1sp \relax
2874     \fi
2875 \fi
2876 \ifdim \dimen@ > \eql@totalrows@ \eql@tagmargin@ratio@ \else
2877     \eql@tagmargin@ \z@
2878 \fi

2879 \@tempdima \dimexpr \displaywidth
2880     - \eql@totalwidth@ - \eql@intercolumns@ \eql@colsepmin@ \relax
2881 \@tempdimb \dimexpr \@tempdima - \tw@ \eql@tagmargin@ \relax
2882 \ifdim \@tempdimb > \z@
2883     \ifdim \eql@tagmargin@threshold \@tempdima < \@tempdimb
2884         \eql@tagmargin@ \z@
2885     \fi
2886 \fi
2887 }

```

9.6 Single Column

`\eql@adjust@calc@lines`

```

2888 \def \eql@adjust@calc@lines {%
2889     \eql@totalcolumns@ \@ne
2890     \eql@intercolumns@ \z@
2891     \eql@colsep@ \z@
2892     \ifdefined \eql@layoutleft
2893         \eql@marginleft@ \glueexpr \eql@layoutleftmargin@ \relax
2894         \eql@marginleft@min@ \glueexpr \eql@layoutleftmarginmin@ \relax
2895         \ifdim \eql@marginleft@ < \eql@marginleft@min@
2896             \eql@marginleft@ \eql@marginleft@min@
2897         \fi
2898         \dimen@ \glueexpr \eql@layoutleftmarginmax@ \relax
2899         \ifdim \eql@marginleft@ > \dimen@
2900             \eql@marginleft@ \dimen@
2901         \fi
2902         \eql@marginright@ \z@
2903         \eql@centeroffset@ \z@
2904     \else
2905         \eql@adjust@calc@tagmargin
2906         \ifdefined \eql@paddingleft@val
2907             \eql@marginleft@ \dimexpr
2908                 (\displaywidth - \eql@totalwidth@ - \eql@tagmargin@) / \tw@
2909                 - \glueexpr \eql@paddingleft@val \relax \relax
2910             \ifdim \eql@marginleft@ < \z@
2911                 \eql@marginleft@ \z@
2912             \fi
2913         \else
2914             \eql@marginleft@ \z@
2915         \fi
2916         \ifdefined \eql@paddingright@val
2917             \eql@marginright@ \dimexpr

```

```

2918         (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
2919         -\glueexpr\eq@paddingright@val\relax\relax
2920     \ifdim\eq@marginright@<\z@
2921         \eq@marginright@\z@
2922     \fi
2923 \else
2924     \eq@marginright@\z@
2925 \fi
2926 \ifdim\eq@tagmargin@>\z@
2927     \ifdefined\eq@tagsleft
2928         \ifdim\eq@marginleft@<\eq@tagsepmin@
2929             \eq@marginleft@\eq@tagsepmin@
2930         \fi
2931         \advance\eq@marginleft@\eq@tagmargin@
2932         \advance\eq@centeroffset@\eq@tagmargin@
2933     \else
2934         \ifdim\eq@marginright@<\eq@tagsepmin@
2935             \eq@marginright@\eq@tagsepmin@
2936         \fi
2937         \advance\eq@marginright@\eq@tagmargin@
2938         \advance\eq@centeroffset@-\eq@tagmargin@
2939     \fi
2940 \fi
2941 \eq@marginleft@min@\z@
2942 \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
2943     \ifdefined\eq@tagsleft+\else-\fi\eq@tagmargin@\relax
2944 \fi

2945 \eq@totalwidth@\dimexpr\displaywidth
2946     -\eq@marginleft@-\eq@marginright@\relax
2947 }

```

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:

```

2948 \def\eq@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:

```

2949     \ifnum\eq@totalcolumns@<\thr@@
2950         \eq@totalcolumns@\tw@
2951         \let\eq@columns@fulllength\eq@false
2952     \fi

```

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

```

2953     \eq@intercolumns@\numexpr(\eq@totalcolumns@-\tw@)/\tw@\relax

```

Evaluate the minimum intercolumn space which we will need often:

```

2954     \eq@colsepmin@\glueexpr\eq@colsepmin@val\relax

```

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

```

2955 \ifdefined\eql@layoutleft
2956   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2957   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2958   \ifdim\eql@marginleft@<\eql@marginleft@min@
2959     \eql@marginleft@\eql@marginleft@min@
2960   \fi
2961 \else

```

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

```

2962   \eql@adjust@calc@tagmargin
2963   \ifdefined\eql@columns@fulllength
2964     \ifdim\eql@tagmargin@>\z@
2965       \advance\eql@tagmargin@\eql@tagsepmin@
2966     \fi
2967   \fi
2968   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2969     -\eql@intercolumns@\eql@colsepmin@\relax
2970   \eql@tagmargin@\z@
2971   \fi
2972   \eql@marginleft@min@\z@
2973 \fi

```

Compute the intercolumn space \eql@colsep@:

```

2974 \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:

```

2975   \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2976   \ifdefined\eql@layoutleft
2977     \advance\eql@colsep@-\eql@marginleft@
2978   \else
2979     \advance\eql@colsep@-\eql@tagmargin@
2980   \fi
2981   \count@\eql@intercolumns@
2982   \ifdefined\eql@columns@fulllength\else
2983     \ifdefined\eql@layoutleft
2984       \advance\count@\@ne
2985     \else
2986       \advance\count@\tw@
2987     \fi
2988   \fi
2989   \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:

```

2990   \ifdim\eql@colsep@<\eql@colsepmin@
2991     \eql@colsep@\eql@colsepmin@
2992   \else
2993     \ifdefined\eql@columns@fulllength\else
2994       \dimen@\glueexpr\eql@colsepmax@val\relax
2995       \ifdim\eql@colsep@>\dimen@
2996         \eql@colsep@\dimen@
2997       \fi

```

```

2998     \fi
2999     \fi
3000 \else

```

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

```

3001     \eq@colsep@eq@colsepmin@
3002 \fi

```

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

```

3003 \ifdefined\eq@layoutleft

```

Set the default value:

```

3004     \ifdim\eq@colsep@=\eq@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:

```

3005     \dimen@ \dimexpr \displaywidth - \eq@totalwidth@
3006             - \eq@intercolumns@ \eq@colsep@ \relax
3007     \ifdim \dimen@ < \eq@marginleft@
3008         \ifdim \dimen@ < \eq@marginleft@min@
3009             \eq@marginleft@ \eq@marginleft@min@
3010         \else
3011             \eq@marginleft@ \dimen@
3012         \fi
3013     \fi
3014 \fi
3015 \else

```

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

```

3016     \ifdefined\eq@columns@fulllength
3017         \eq@marginleft@ \z@

```

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

```

3018 \else
3019     \eq@marginleft@ \dimexpr \displaywidth - \eq@totalwidth@
3020             - \eq@intercolumns@ \eq@colsep@ - \eq@tagmargin@ / \tw@ \relax
3021     \ifdim \eq@marginleft@ < \z@
3022         \eq@marginleft@ \z@
3023     \fi
3024 \fi

```

Add tag margin in case of left tags:

```

3025     \ifdefined\eq@tagsleft
3026         \advance \eq@marginleft@ \eq@tagmargin@
3027     \fi
3028 \fi

```

Find the best row for tag placement:

```

3029     \eq@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:

```

3030 \ifdefined\eq\@tagsleft
3031   \let\eq\@adjust@columns@test\eq\@adjust@columns@test@tagsleft
3032 \else
3033   \let\eq\@adjust@columns@test\eq\@adjust@columns@test@tagsright
3034 \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

```

3035 \ifdefined\eq\@numbering@multi
3036   \eq\@dimensions@for{%
3037     \ifdim\eq\@tagwidth@>\z@
3038       \eq\@dimensions@calc
3039       \eq\@adjust@columns@test
3040     \fi
3041   }%
3042 \else
3043   \ifnum\eq\@tagpos@row@>\z@
3044     \ifnum\eq\@tagpos@row@>\eq\@totalrows@\else
3045       \eq\@dimensions@get\eq\@tagpos@row@
3046       \eq\@tagwidth@\eq\@tagwidth@block@
3047       \eq\@dimensions@calc
3048       \eq\@adjust@columns@test
3049     \fi
3050   \fi
3051   \ifnum\eq\@tagpos@prevrow@>\z@
3052     \eq\@dimensions@get\eq\@tagpos@prevrow@
3053     \eq\@tagwidth@\eq\@tagwidth@block@
3054     \eq\@dimensions@calc
3055     \eq\@adjust@columns@test
3056   \fi
3057 \fi

```

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

```

3058 \advance\eq\@totalwidth@\dimexpr
3059   \eq\@intercolumns@\eq\@colsep@+\eq\@marginleft@\relax
3060 }

```

Placement for Right Tags.

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

```

3061 \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?

```

3062   \@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.

```

3063   \ifdim\@tempdima<\dimexpr
3064     \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).

```

3065 \ifdim\@tempdima<\dimexpr
3066 \eq\marginleft@min@+\eq\line@widthsep@\eq\colsepmin@\relax\else

```

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

```

3067 \eq\adjust@columns@modify@tagsright
3068 \fi
3069 \fi
3070 }

```

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

```

3071 \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.

```

3072 \ifnum\eq\line@widthsep@>\z@
3073 \dimen@\@tempdima
3074 \count@\eq\line@widthsep@
3075 \ifdefined\eq\layoutleft
3076 \advance\dimen@-\eq\marginleft@
3077 \else
3078 \ifdefined\eq\columns@fulllength\else
3079 \advance\count@\@ne
3080 \fi
3081 \fi
3082 \divide\dimen@\count@

```

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

```

3083 \ifdim\dimen@<\eq\colsep@
3084 \ifdim\dimen@<\eq\colsepmin@
3085 \eq\colsep@\eq\colsepmin@
3086 \else
3087 \eq\colsep@\dimen@
3088 \fi
3089 \fi
3090 \fi

```

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

```

3091 \dimen@\dimexpr\@tempdima-\eq\line@widthsep@\eq\colsep@\relax
3092 \ifdim\eq\marginleft@>\dimen@
3093 \eq\marginleft@\dimen@
3094 \fi
3095 }

```

Placement for Left Tags.

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

```

3096 \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.

```

3097 \count@\numexpr\eq\intercolumns@-\eq\line@availsep@\relax
3098 \@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.

```

3099 \ifdim\@tempdima>\dimexpr
3100 \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).

```

3101 \ifdim\eq>tagwidth@<%
3102 \ifdefined\eq\layoutleft
3103 \glueexpr\eq\layoutleftmarginmax\relax
3104 \else
3105 \displaywidth
3106 \fi

```

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

```

3107 \ifdim\@tempdima>\dimexpr
3108 \displaywidth-\eq\totalwidth@-\count@\eq\colsepmin@\relax\else

```

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

```

3109 \eq\adjust@columns@modify@tagsleft
3110 \fi
3111 \fi
3112 \fi
3113 }

```

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

```

3114 \def\eq\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.

```

3115 \ifnum\count@>\z@
3116 \dimen@\dimexpr\displaywidth-\eq\totalwidth@-\@tempdima\relax
3117 \ifdefined\eq\columns@fulllength\else
3118 \advance\count@\@ne
3119 \fi
3120 \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.

```

3121 \ifdim\dimen@<\eq\colsep@
3122 \ifdim\dimen@<\eq\colsepmin@
3123 \dimen@\eq\colsepmin@
3124 \fi
3125 \advance\dimen@-\eq\colsep@
3126 \advance\eq\marginleft@-\eq\intercolumns@\dimen@
3127 \advance\eq\colsep@\dimen@
3128 \fi
3129 \fi

```


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

```

3130 \dimen@dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3131 \ifdim\eql@marginleft@<\dimen@
3132 \eql@marginleft@\dimen@
3133 \fi
3134 }

```

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:

```

3135 \ifdefined\inf@bad\else%
3136 \newcount\inf@bad
3137 \inf@bad1000000\relax
3138 \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`:

```

3139 \let\eql@restore@hfuzz\@empty
3140 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}

```

`\eql@alignbadness@` The registers `\eql@alignbadness@` and `\eql@tagbadness@` 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:

```

3141 \newcount\eql@alignbadness@
3142 \newcount\eql@tagbadness@
3143 \newcount\eql@arrange@badness@
3144 \eql@alignbadness@\inf@bad
3145 \eql@tagbadness@\inf@bad

```

10.2 Arrangement Methods

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

```

3146 \def\eql@arrange@try#1{%
3147 \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
3148 \setbox\eql@cellbox@\hbox to\displaywidth{%
3149 \unhbox\eql@cellbox@\unkern\kern#1}%
3150 \eql@arrange@badness@badness
3151 \else
3152 \eql@arrange@badness@\m@ne
3153 \fi
3154 }

```

`\eql@arrange@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

```
3155 \def\eql@arrange@print#1#2{%
3156   \eql@restore@hfuzz
3157   \if@eqnsw
3158     \ifdefined\eql@tagsleft
3159       \eql@tagbox@print@tagsleft
3160     \fi
3161   \fi
3162   \hbox to\displaywidth{%
3163     #1%
3164     \unhbox\eql@cellbox@unkern
3165     #2%
3166     \eql@tagging@mathaddlast
3167   }%
3168   \if@eqnsw
3169     \ifdefined\eql@tagsleft\else
3170       \eql@tagbox@print@tagsright
3171     \fi
3172   \fi
3173 }
```

`\eql@arrange@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 `\eql@display@firstavail@` to the minimum left available space we can guarantee:

```
3174 \def\eql@arrange@print@alignleft#1#2{%
3175   \eql@display@firstavail@set{\dimexpr#1\relax}%
3176   \eql@arrange@print{\kern#1}{\kern#2}%
3177 }

3178 \def\eql@arrange@print@alignright#1#2{%
3179   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3180   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3181 }

3182 \def\eql@arrange@print@aligncenter#1{%
3183   \eql@display@firstavail@set{\dimexpr
3184     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3185   \ifdim#1>\z@
3186     \eql@arrange@print{\kern#1\hfil}{}%
3187   \else
3188     \eql@arrange@print{\hfil}{\kern-#1}%
3189   \fi
3190 }
```

`\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:

```
3191 \def\eql@arrange@init{%
3192   \eql@save@hfuzz
3193   \hfuzz\maxdimen
3194   \eql@shape@select
3195 }
```

`\eql@arrange@print@line` Select the appropriate adjustment method depending on the current alignment position,

the selected tag placement if any: **TODO:** adjust

```

3196 \def\eql@arrange@print@line{%
3197   \eql@tagging@tagaddbox
3198   \csname eql@arrange%
3199     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3200     @init\endcsname
3201   \csname eql@arrange%
3202     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3203     @\ifdefined\eql@tagpos@reserve
3204       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3205       notag\fi\endcsname
3206 }
```

10.3 Central Alignment

TODO: describe

```

3207 \def\eql@arrange@aligncenter@init{%
3208   \eql@tagging@aligncenter
3209   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3210     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3211 }
```

TODO: describe

```

3212 \def\eql@arrange@aligncenter@notag{%
3213   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3214     \ifdim\eql@line@offset@<\eql@marginleft@min@
3215       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3216     \else
3217       \eql@line@offset@
3218     \fi
3219   \eql@arrange@print@aligncenter\eql@line@offset@
3220 \else
3221   \ifdim\eql@line@offset@<\eql@marginleft@min@
3222     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3223   \else
3224     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3225   \fi
3226 \fi
3227 }
```

TODO: describe

```

3228 \def\eql@arrange@aligncenter@tagsright{%
3229   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3230     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3231       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3232     \else
3233       \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3234     \fi
3235   \eql@arrange@print@aligncenter\eql@line@offset@
3236 \else
3237   \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3238   \ifnum\eql@arrange@badness@<\eql@tagbadness@
3239     \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3240       \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3241     \else
```

```

3242     \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3243     \fi
3244   \else
3245     \let\eql@tagpos@reserve\eql@false
3246     \eql@arrange@aligncenter@notag
3247   \fi
3248 \fi
3249 }

3250 \def\eql@arrange@aligncenter@tagsleft{%
3251   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3252     \ifdim\dimexpr\displaywidth-\eql@cellwidth@>\relax%
3253       \ifdim\eql@line@offset@<\eql@tagwidth@
3254         \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@>\relax
3255       \else
3256         \eql@line@offset@
3257       \fi
3258     \eql@arrange@print@aligncenter\eql@line@offset@
3259   \else
3260     \eql@arrange@try\eql@tagwidth@
3261     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3262       \ifdim\eql@line@offset@<\eql@tagwidth@
3263         \eql@arrange@print@alignleft\eql@tagwidth@\z@
3264       \else
3265         \eql@arrange@print@alignright\eql@tagwidth@\z@
3266       \fi
3267     \else
3268       \let\eql@tagpos@reserve\eql@false
3269       \eql@arrange@aligncenter@notag
3270     \fi
3271   \fi
3272 \else
3273   \eql@arrange@aligncenter@notag
3274 \fi
3275 }

```

10.4 Left Alignment

```

3276 \def\eql@arrange@alignleft@init{%
3277   \eql@tagging@alignleft
3278   \eql@line@offset@\dimexpr\eql@marginleft@+\eql@shape@amount@>\relax
3279   \ifdim\eql@line@offset@<\eql@marginleft@min@
3280     \eql@line@offset@\eql@marginleft@min@
3281   \fi
3282 }

3283 \def\eql@arrange@alignleft@notag{%
3284   \ifdim\eql@line@offset@>\eql@marginleft@min@
3285     \eql@arrange@try\eql@line@offset@
3286     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3287       \eql@arrange@print@alignleft\eql@line@offset@\z@
3288     \else
3289       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3290     \fi
3291   \else
3292     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3293   \fi
3294 }

```

```

3295 \def\eqL@arrange@alignleft@tagsright{%
3296   \eqL@arrange@try{\dimexpr\eqL@line@offset@+\eqL@tagwidth@\relax}%
3297   \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3298     \eqL@arrange@print@alignleft\eqL@line@offset@\eqL@tagwidth@
3299   \else
3300     \ifdim\eqL@line@offset@>\eqL@marginleft@min@
3301       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@\relax}%
3302     \fi
3303     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3304       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3305     \else
3306       \let\eqL@tagpos@reserve\eqL@false
3307       \eqL@arrange@alignleft@notag
3308     \fi
3309   \fi
3310 }

3311 \def\eqL@arrange@alignleft@tagsleft{%
3312   \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
3313     \ifdim\eqL@line@offset@>\eqL@tagwidth@
3314       \eqL@arrange@try\eqL@line@offset@
3315       \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3316         \eqL@arrange@print@alignleft\eqL@line@offset@\z@
3317       \else
3318         \eqL@arrange@try\eqL@tagwidth@
3319         \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3320           \eqL@arrange@print@alignright\eqL@tagwidth@\z@
3321         \else
3322           \let\eqL@tagpos@reserve\eqL@false
3323           \eqL@arrange@print@alignright\eqL@marginleft@min@\z@
3324         \fi
3325       \fi
3326     \else
3327       \eqL@arrange@try\eqL@tagwidth@
3328       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3329         \eqL@arrange@print@alignleft\eqL@tagwidth@\z@
3330       \else
3331         \let\eqL@tagpos@reserve\eqL@false
3332         \eqL@arrange@alignleft@notag
3333       \fi
3334     \fi
3335   \else
3336     \eqL@arrange@alignleft@notag
3337   \fi
3338 }

```

10.5 Right Alignment

```

3339 \def\eqL@arrange@alignright@init{%
3340   \eqL@tagging@alignright
3341   \eqL@line@offset@\dimexpr\eqL@marginright@-\eqL@shape@amount@\relax
3342   \ifdim\eqL@line@offset@<\z@
3343     \eqL@line@offset@\z@
3344   \fi
3345 }

```

TODO: describe

```

3346 \def\eqL@arrange@alignright@notag{%
3347   \ifdim\eqL@line@offset@>\z@

```

```

3348 \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3349 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3350 \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3351 \else
3352 \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3353 \fi
3354 \else
3355 \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3356 \fi
3357 }

```

TODO: describe

```

3358 \def\eqL@arrange@alignright@tagsright{%
3359 \ifdim\eqL@line@offset@>\eqL@tagwidth@
3360 \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3361 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3362 \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3363 \else
3364 \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3365 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3366 \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3367 \else
3368 \let\eqL@tagpos@reserve\eqL@false
3369 \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3370 \fi
3371 \fi
3372 \else
3373 \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3374 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3375 \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3376 \else
3377 \let\eqL@tagpos@reserve\eqL@false
3378 \eqL@arrange@alignright@notag
3379 \fi
3380 \fi
3381 }

```

TODO: describe

```

3382 \def\eqL@arrange@alignright@tagsleft{%
3383 \ifdim\eqL@tagwidth@>\eqL@marginleft@min@
3384 \eqL@arrange@try{\dimexpr\eqL@line@offset@+\eqL@tagwidth@relax}%
3385 \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3386 \eqL@arrange@print@alignright\eqL@tagwidth@\eqL@line@offset@
3387 \else
3388 \ifdim\eqL@line@offset@>z@
3389 \eqL@arrange@try\eqL@tagwidth@
3390 \fi
3391 \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3392 \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3393 \else
3394 \let\eqL@tagpos@reserve\eqL@false
3395 \eqL@arrange@alignright@notag
3396 \fi
3397 \fi
3398 \else
3399 \eqL@arrange@alignright@notag
3400 \fi
3401 }

```

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

TODO: describe

`\eql@box@cr`

```
3402 \def\eql@box@cr{%
3403   \ifmmode\else\unskip\fi
3404   \eql@vspaceskip@z@skip
3405   \eql@ampprotect\eql@box@cr@testall\eql@box@cr@process}
```

`\eql@box@cr@testall` **TODO:** describe

`\eql@box@cr@parse`

```
3406 \def\eql@box@cr@testall{\eql@parseopt@aux\eql@box@cr@parse}
3407 \def\eql@box@cr@parse{%
3408   \ifx\eql@parseopt@token[%
3409     \let\eql@parseopt@next\eql@parseopt@vspace
3410   \fi
3411   \ifx\eql@parseopt@token*%
3412     \let\eql@parseopt@next\eql@parseopt@gobble
3413   \fi
3414   \ifx\eql@parseopt@token.%
3415     \let\eql@parseopt@next\eql@parseopt@punctpass
3416   \fi
3417   \ifx\eql@parseopt@token,%
3418     \let\eql@parseopt@next\eql@parseopt@punctpass
3419   \fi
3420   \ifx\eql@parseopt@token~%
3421     \let\eql@parseopt@next\eql@parseopt@punctpass
3422   \fi
3423   \ifx\eql@parseopt@token'%
3424     \let\eql@parseopt@next\eql@parseopt@punctnext
3425   \fi
3426   \ifx\eql@parseopt@token&%
3427     \let\eql@parseopt@next\eql@parseopt@end
3428   \fi
3429 }
```

`\eql@box@cr@process`

```
3430 \def\eql@box@cr@process{%
3431   \eql@box@endline
3432   \expandafter\eql@box@cr@vskip\expandafter{\the\eql@vspaceskip@}%
3433 }
```

`\eql@box@endline`

```
3434 \def\eql@box@endline{%
3435   \eql@punct@apply@line
```

```

3436 \eq@hook@lineout
3437 }

```

\eq@box@cr@vskip

```

3438 \def\eq@box@cr@vskip#1{%
3439 \eq@box@lastcell
3440 \cr
3441 \noalign{%
3442 \vskip#1\relax
3443 }%
3444 }

```

11.2 Stacked Mode

```

3445 \def\eq@box@lastcell@stacked{&\omit\kern-2\eq@colsep@}

```

TODO: templates

```

3446 \def\eq@box@open@stacked{%
3447 \eq@shape@align@enable
3448 \let\eq@box@lastcell\eq@box@lastcell@stacked
3449 \everycr{\noalign{%
3450 \eq@verbose@info\eq@verbose@msg@startline
3451 \global\advance\eq@row@\@ne
3452 }}%
3453 \tabskip\z@skip
3454 \halign\bgroup
3455 &%
3456 \global\let\eq@cell@container\@empty
3457 \setbox\eq@cellbox@\hbox{%
3458 \eq@strut@cell
3459 \@lign
3460 $\m@th\eq@mathstyle
3461 \eq@hook@colin
3462 ##%
3463 \eq@punct@apply@col
3464 \eq@hook@colout
3465 \eq@tagging@mathsave
3466 $%
3467 \eq@tagging@mathaddlast
3468 }%
3469 \ifdefined\eq@shape@lastrow
3470 \eq@totalrows@\eq@row@
3471 \fi
3472 \eq@shape@eval
3473 \eq@cell@container
3474 \ifdefined\eq@frame@cmd
3475 \ifcase\eq@shape@pos@
3476 \eq@frame@measure
3477 \advance\eq@shape@amount@-\eq@frame@margin@
3478 \or\or
3479 \eq@frame@measure
3480 \advance\eq@shape@amount@+\eq@frame@margin@
3481 \fi
3482 \eq@frame@print
3483 \fi
3484 \ifcase\eq@shape@pos@
3485 \kern\eq@shape@amount@

```



```

3486         \box\eql@cellbox@
3487         \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3488         -\eql@shape@amount@+\@flushglue\relax
3489         \eql@tagging@alignleft
3490     \or
3491         \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3492         \box\eql@cellbox@
3493         \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3494         \eql@tagging@aligncenter
3495     \or
3496         \hskip\glueexpr\eql@paddingleft@\eql@paddingright@
3497         +\eql@shape@amount@+\@flushglue\relax
3498         \box\eql@cellbox@
3499         \kern-\eql@shape@amount@
3500         \eql@tagging@alignright
3501     \fi
3502     \tabskip\eql@colsep@\relax
3503 \crrc
3504 \noalign{%
3505     \global\let\eql@shape@lastrow\eql@false
3506     \eql@hook@blockbefore
3507 }%
3508 \eql@hook@blockin
3509 }
3510 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

11.3 Aligned Mode

```

3511 \def\eql@box@lastcell@odd{%
3512     &\omit
3513     \eql@prevwidth@\wd\eql@cellbox@
3514     \let\eql@frame@cmd\eql@frame@prevcmd
3515     \ifdefined\eql@frame@cmd
3516         \eql@frame@measure
3517         \advance\eql@prevwidth@\eql@frame@margin@
3518         \eql@frame@print
3519     \fi
3520     \kern-\eql@prevwidth@
3521     \unhbox\eql@cellbox@
3522     \hfil
3523     &\omit\kern-\eql@colsep@
3524 }%
3525 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}
3526 \def\eql@verbose@msg@startline@aligned{starting new line}
3527 \def\eql@box@open@aligned{%
3528 % \TODO templates
3529     \eql@shape@align@disable
3530     \let\eql@box@lastcell\@empty
3531     \everycr{\noalign{%
3532         \eql@verbose@info\eql@verbose@msg@startline@aligned
3533     }}%
3534     \tabskip\z@skip
3535     \halign\bgroup
3536     &%
3537         \let\eql@box@lastcell\eql@box@lastcell@odd
3538         \global\let\eql@cell@container\@empty
3539         \global\setbox\eql@cellbox@\hbox{%
3540             \eql@strut@cell

```

```

3541 \@lign
3542 $\m@th\eq\@mathstyle
3543 \eq\@hook@colin
3544 ##%
3545 \eq\@punct@apply@next
3546 \eq\@class@innerleft
3547 \eq\@hook@innerleft
3548 \eq\@tagging@mathsave
3549 $%
3550 \eq\@tagging@mathaddlast
3551 }%
3552 \eq\@cell@container
3553 \hfil
3554 \kern\wd\eq\@cellbox@
3555 \ifdefined\eq\@frame@cmd
3556 \eq\@frame@measure
3557 \kern\eq\@frame@margin@
3558 \fi
3559 \global\let\eq\@frame@prevcmd\eq\@frame@cmd
3560 \tabskip\z@skip
3561 &%
3562 \eq\@prevwidth@\wd\eq\@cellbox@
3563 \let\eq\@box@lastcell\eq\@box@lastcell@even
3564 \let\eq\@frame@cmd\eq\@frame@prevcmd
3565 \global\let\eq\@cell@container\@empty
3566 \setbox\eq\@cellbox@\hbox{%
3567 \unhbox\eq\@cellbox@
3568 \eq\@strut@cell
3569 \@lign
3570 $\m@th\eq\@mathstyle
3571 \eq\@hook@innerright
3572 \eq\@class@innerright@sel
3573 ##%
3574 \eq\@punct@apply@col
3575 \eq\@hook@colout
3576 \eq\@tagging@mathsave
3577 $%
3578 \eq\@tagging@mathaddlast
3579 }%
3580 \eq\@cell@container
3581 \ifdefined\eq\@frame@cmd
3582 \eq\@frame@measure
3583 \advance\eq\@prevwidth@\eq\@frame@margin@
3584 \eq\@frame@print
3585 \fi
3586 \kern-\eq\@prevwidth@
3587 \unhbox\eq\@cellbox@
3588 \hfil
3589 \tabskip\eq\@colsep@\relax
3590 \crrc
3591 \noalign{%
3592 \eq\@hook@blockbefore
3593 }%
3594 \eq\@hook@blockin
3595 }
3596 \def\eq\@mode@aligned{\let\eq\@box@open\eq\@box@open@aligned}

```

11.4 Cases Mode

TODO: describe

TODO: how to get proper height in tagging (and avoid nulldelimiterspace) **TODO:** add alignment?

```

3597 \def\eql@box@lastcell@cases{&}%

3598 \let\eql@box@cases@condtext\eql@false
3599 \let\eql@box@cases@condintro\@empty

3600 \def\eql@verbose@msg@startline@cases{starting new line}
3601 \def\eql@box@open@cases{%
3602   \eql@shape@align@disable
3603   \let\eql@box@lastcell\@empty
3604   \everycr{\noalign{%
3605     \eql@verbose@info\eql@verbose@msg@startline@cases
3606   }}%
3607   \tabskip\z@skip
3608   \halign\bgroup
3609     \let\eql@box@lastcell\eql@box@lastcell@cases
3610     \global\let\eql@cell@container\@empty
3611     \global\setbox\eql@cellbox@\hbox{%
3612       \eql@strut@cell
3613       \@lign
3614       $\m@th\eql@mathstyle
3615       \eql@hook@colin
3616       ##%
3617       \eql@punct@apply@next
3618       \eql@tagging@mathsave
3619       $%
3620       \eql@tagging@mathaddlast
3621     }%
3622     \eql@cell@container
3623     \unhbox\eql@cellbox@
3624     \hfil
3625     \eql@tagging@alignleft
3626     \tabskip\eql@colsep@\relax
3627   &%
3628   \let\eql@box@lastcell\@empty
3629   \global\let\eql@cell@container\@empty
3630   \setbox\eql@cellbox@\hbox{%
3631     \unhbox\eql@cellbox@
3632     \eql@strut@cell
3633     \@lign
3634     $\m@th\eql@mathstyle
3635     \ifdefined\eql@box@cases@condtext
3636       \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
3637     \eql@box@cases@condintro
3638     ##%
3639     \eql@punct@apply@col
3640     \egroup
3641     \eql@hook@colout
3642     \eql@tagging@mathsave
3643     $%
3644     \eql@tagging@mathaddlast
3645   }%
3646   \eql@cell@container
3647   \unhbox\eql@cellbox@

```

```

3648     \hfil
3649     \eql@tagging@alignleft
3650     \tabskip\z@skip
3651     \crrcr
3652     \noalign{%
3653         \eql@hook@blockbefore
3654     }%
3655     \eql@hook@blockin
3656 }

3657 \def\eql@mode@cases{\let\eql@box@open\eql@box@open@cases}

```

11.5 Main

```

3658 \let\eql@box@box\center
3659 \let\eql@box@open@undefined
3660 \let\eql@box@frame@firstofone
3661 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

3662 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathinner{\left#1##1\right#2}}}
3663 \def\eql@box@getdim{\setbox\@ne\hbox{\null\ht\@ne\dp\@ne\z@}
3664 \def\eql@box@deldim#1{\hbox{$\m@th\hbox{\null\delimiterspace\z@\left#1
3665     \ifx\eql@box@box\center\center{\box\@ne}\else\box\@ne\fi\right.$}}
3666 \def\eql@box@ldelim#1{%
3667     \def\eql@box@frame##1{\mathinner{\eql@box@getdim\eql@box@deldim#1##1}}}
3668 \def\eql@box@rdelim#1{%
3669     \def\eql@box@frame##1{\mathinner{\eql@box@getdim##1\eql@box@deldim#1}}}

```

TODO: can we avoid setting `\eql@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!

```

3670 \def\eql@box@close{%
3671     \ifvmode\else
3672         \ifmmode\else\unskip\fi
3673         \global\let\eql@shape@lastrow\eql@true
3674         \eql@punct@apply@block
3675         \eql@box@endline
3676         \eql@box@lastcell
3677         \cr
3678     \fi
3679     \noalign{%
3680         \eql@hook@blockafter
3681         \global\let\eql@shape@lastrow\eql@false
3682     }%
3683     \eql@tagging@tablesaveinner
3684 \egroup
3685 }

```

`\eql@box@vcenter`

```

3686 \def\eql@box@vcenter#1{%
3687     \ifmmode
3688         \vcenter{#1}%
3689     \else
3690         $\m@th\vcenter{#1}$%
3691     \fi
3692 }

```

\eql@box@start

```
3693 \let\eql@box@endmath\eql@false
3694 \def\eql@box@start{%
3695   \relax
3696   \ifmmode
3697     \let\eql@box@endmath\eql@false
3698   \else
3699     \let\eql@box@endmath\eql@true
3700     \expandafter$%$
3701   \fi
3702   \eql@box@processopt
3703   \eql@stack@save@box
3704   \let\eql@frame@cmd\@undefined
3705   \let\eql@layoutleft\eql@false
3706   \eql@row@z@
3707   \eql@totalrows@\@M
3708   \eql@shape@select
3709   \setbox\z@\ifx\eql@box@box\vcenter
3710     \expandafter\vbox
3711   \else
3712     \expandafter\eql@box@box
3713   \fi\bgroup
3714   \eql@display@nest
3715   \let\\eql@box@cr
3716   \eql@spread@set
3717   \eql@strut@make
3718   \eql@box@open
3719 }
```

\eql@box@end

```
3720 \def\eql@box@end{%
3721   \eql@box@close
3722   \egroup
3723   \eql@box@frame{%
3724     \ifdefined\eql@display@marginleft
3725       \hskip\glueexpr\eql@display@marginleft\relax
3726     \fi
3727     \ifx\eql@box@box\vcenter
3728       \eql@box@vcenter{\unvbox\z@}%
3729     \else
3730       \box\z@
3731     \fi
3732     \eql@tagging@tableaddinner
3733     \ifdefined\eql@display@marginright
3734       \hskip\glueexpr\eql@display@marginright\relax
3735     \fi
3736   }%
3737   \eql@stack@restore
3738   \ifdefined\eql@box@endmath
3739     \expandafter$%$
3740   \fi
3741 }
```

11.6 Environment

`equationsbox` (*env.*)

```

3742 \newenvironment{equationsbox}{%
3743   \eq@verbose@info\eq@verbose@msg@enterenv
3744   \eq@ampprotect\eq@box@testall\eq@box@start
3745 }{%
3746   \eq@box@end
3747   \eq@verbose@info\eq@verbose@msg@leaveenv
3748 }

3749 \def\eq@box@testall{\eq@parseopt@main\eq@box@parse}
3750 \def\eq@box@parse{%
3751   \ifx\eq@parseopt@token[%
3752     \let\eq@parseopt@next\eq@parseopt@opt
3753   \fi
3754   \ifx\eq@parseopt@token.%
3755     \let\eq@parseopt@next\eq@parseopt@punctpass
3756   \fi
3757   \ifx\eq@parseopt@token,%
3758     \let\eq@parseopt@next\eq@parseopt@punctpass
3759   \fi
3760   \ifx\eq@parseopt@token~%
3761     \let\eq@parseopt@next\eq@parseopt@punctpass
3762   \fi
3763   \ifx\eq@parseopt@token'%
3764     \let\eq@parseopt@next\eq@parseopt@punctopt
3765   \fi
3766   \ifx\eq@parseopt@token=%
3767     \let\eq@parseopt@next\eq@parseopt@lines
3768   \fi
3769   \ifx\eq@parseopt@token|%
3770     \let\eq@parseopt@next\eq@parseopt@columns
3771   \fi
3772   \ifx\eq@parseopt@token<%
3773     \let\eq@parseopt@next\eq@parseopt@ampeq
3774   \fi
3775   \ifx\eq@parseopt@token>%
3776     \let\eq@parseopt@next\eq@parseopt@eqamp
3777   \fi
3778 }
```

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

```

3779 \def\eq@box@processopt{%
3780   \let\eq@box@frame\@firstofone
3781   \let\eq@display@marginleft\@undefined
3782   \let\eq@display@marginright\@undefined
3783   \eq@nextopt@process{equationsbox}%
3784   \let\eq@punct@block\eq@punct@main
3785   \let\eq@punct@main\@undefined
3786   \eq@colsep@\glueexpr\eq@box@colsep\relax
3787   \ifdefined\eq@paddingleft@val
3788     \eq@paddingleft@\glueexpr\eq@paddingleft@val\relax
3789   \else
3790     \eq@paddingleft@\z@
3791   \fi
3792   \ifdefined\eq@paddingright@val
```

```

3793 \eq\paddingright\glueexpr\eq\paddingright\val\relax
3794 \else
3795 \eq\paddingright\z@
3796 \fi
3797 \eq\indent\glueexpr\eq\indent\val\relax
3798 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

3799 \def\eq\single\start\native{%
3800 \eq\display\init
3801 \eq\display\print
3802 \let\raisetag\eq\raisetag\default
3803 \eq\shape\align\disable
3804 \eq\hook\eqin
3805 % \mathopen{}\%
3806 }%

```

TODO: describe

```

3807 \def\eq\single\end\native{%
3808 % \mathclose{}\%
3809 \eq\tags\container
3810 \eq\numbering\single\eval
3811 \if\eqnsw
3812 \ifdefined\eq\tagsleft
3813 \leqno
3814 \else
3815 \eqno
3816 \fi
3817 \eq\composetag\print
3818 \fi
3819 \eq\interline\container
3820 \advance\eq\belowspace\eq\vspaceskip\
3821 \eq\display\container
3822 \eq\display\penalty
3823 \eq\display\vspace\native
3824 }%

```

12.2 Print

```

3825 \def\eq\single\start\print{%
3826 \eq\display\init
3827 \eq\display\print
3828 \eq\shape\align\enable
3829 \eq\totalrows\@one
3830 \eq\row\@one
3831 \eq\arrange\init
3832 \global\let\eq\cell\container\@empty
3833 \prevgraf\numexpr\prevgraf+\@one\relax
3834 \setbox\eq\cellbox\hbox\bgroup
3835 \eq\restore\hfuzz
3836 \eq\strut\cell

```

```

3837     $\m@th\eq@mathstyle%$
3838     \eq@hook@eqin
3839 }

3840 \def\eq@single@end@print{%
3841     \eq@tagging@mathsave
3842     $%$
3843     \hfil
3844     \kern\z@
3845     \egroup
3846     \prevgraf\numexpr\prevgraf-\@ne\relax
3847     \eq@shape@eval
3848     \eq@cell@container
3849     \ifdefined\eq@frame@cmd
3850         \eq@frame@adjust
3851     \fi

3852     \eq@cellwidth@\wd\eq@cellbox@
3853     \eq@line@height@\ht\eq@cellbox@
3854     \eq@line@depth@\dp\eq@cellbox@
3855     \eq@totalwidth@\eq@cellwidth@
3856     \eq@totalheight@\dimexpr\eq@line@height@+\eq@line@depth@\relax
3857     \eq@topheight@\eq@line@height@
3858     \eq@bottomdepth@\eq@line@depth@

3859     \eq@tags@container
3860     \eq@numbering@single@eval
3861     \if@eqnsw
3862         \eq@tagbox@make\eq@composetag@print
3863         \eq@taggrows@\@ne
3864         \ifdefined\eq@tagpos@reserve\else
3865             \eq@tagwidth@\z@
3866         \fi
3867         \eq@tagheight@block@\ht\eq@tagbox@
3868         \eq@tagdepth@block@\dp\eq@tagbox@
3869     \else
3870         \eq@numbering@warnunused
3871         \eq@tagwidth@\z@
3872         \eq@taggrows@\z@
3873     \fi
3874     \eq@tagwidth@max@\eq@tagwidth@
3875     \eq@tagpos@single@eval
3876     \eq@tagpos@print@line@eval

3877     \eq@intercolumns@\z@
3878     \eq@adjust@calc@lines

3879     \eq@display@halign@init{}}%
3880     \halign{##\crr
3881         \noalign{\eq@display@halign@start}}%
3882         \eq@arrange@print@line
3883         \cr
3884         \noalign{\eq@display@halign@end}}%
3885         \eq@tagging@tablesavelines
3886     }%
3887     \eq@tagpos@print@line@end
3888     \eq@display@close
3889 }

```


13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```
3890 \def\eql@lines@measure@line@begin{%
3891   \eql@verbose@info\eql@verbose@msg@startline
3892   \eql@numbering@measure@line@begin
3893   \eql@hook@linein
3894 }
```

TODO: describe

```
3895 \def\eql@lines@measure@line@end{%
3896   \eql@punct@apply@line
3897   \eql@hook@lineout
3898 }
```

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

```
3899 \def\eql@lines@measure@cell{%
3900   \ifdefined\eql@frame@cmd
3901     \ifcase\eql@shape@pos@
3902       \eql@frame@measure
3903       \advance\eql@shape@amount@-\eql@frame@margin@
3904     \or\or
3905       \eql@frame@measure
3906       \advance\eql@shape@amount@+\eql@frame@margin@
3907     \fi
3908     \eql@frame@print
3909   \fi
3910   \eql@cellwidth@\wd\eql@cellbox@
3911   \eql@line@height@\ht\eql@cellbox@
3912   \eql@line@depth@\dp\eql@cellbox@
3913   \eql@dimensions@startrow
3914   \eql@dimensions@savercell
3915   \kern\eql@cellwidth@
3916 }
```

`\eql@lines@measure`

```
3917 \def\eql@lines@measure{%
3918   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
3919   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3920   \eql@totalrows@\@M
3921   \eql@shape@select

3922   \setbox\z@\vbox{\measuring@true\halign{%
3923     \global\let\eql@cell@container\@empty
3924     \setbox\eql@cellbox@\hbox{%
3925       \eql@strut@cell
3926       \@lign
3927       $\m@th\eql@mathstyle
3928       \eql@hook@colin
3929       ##%
```

```

3930         \eql@punct@apply@col
3931         \eql@hook@colout
3932     $%
3933 }%
3934 \ifdefined\eql@shape@lastrow
3935     \eql@totalrows@\eql@row@
3936 \fi
3937 \eql@shape@eval
3938 \eql@cell@container
3939 \eql@lines@measure@cell
3940 \eql@measure@tag
3941 \eql@measure@endrow
3942 \crrr

3943 \noalign{%
3944     \global\let\eql@shape@lastrow\eql@false
3945     \eql@hook@blockbefore
3946 }%
3947 \eql@hook@blockin
3948 \eql@scan@body
3949 \ifvmode\else
3950     \global\let\eql@shape@lastrow\eql@true
3951     \eql@punct@apply@block
3952     \eql@hook@blockout
3953     \eql@display@endline
3954 \cr
3955 \fi
3956 \omit
3957 \cr
3958 \noalign{%
3959     \eql@hook@blockafter
3960     \global\let\eql@shape@lastrow\eql@false
3961 }%
3962 }%

3963 \eql@measure@close

3964 \setbox\z@\vbox{%
3965     \unvbox\z@
3966     \unpenalty
3967     \global\setbox\@ne\lastbox
3968 }%
3969 \eql@totalwidth@\wd\@ne

3970 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
3971 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

3972 \def\eql@lines@adjust{%
3973     \eql@tagpos@adjust@eval
3974     \eql@adjust@calc@lines
3975     \eql@numbering@best@eval
3976 }

```

13.3 Print

TODO: describe

mes@print@line@begin

```
3977 \def\eq@lines@print@line@begin{%
3978   \eq@verbose@info\eq@verbose@msg@startline
3979   \eq@numbering@print@line@begin
3980   \eq@hook@linein
3981 }
```

TODO: describe

```
3982 \def\eq@lines@print@line@end{%
3983   \eq@punct@apply@line
3984   \eq@hook@lineout
3985 }
```

TODO: describe

```
3986 \def\eq@lines@print@line@adjust{%
3987   \ifdefined\eq@frame@cmd
3988     \ifcase\eq@shape@pos@
3989       \eq@frame@measure
3990       \advance\eq@shape@amount@-\eq@frame@margin@
3991     \or\or
3992       \eq@frame@measure
3993       \advance\eq@shape@amount@+\eq@frame@margin@
3994     \fi
3995     \eq@frame@adjust
3996   \fi
3997   \eq@cellwidth@\wd\eq@cellbox@
3998   \eq@line@height@\ht\eq@cellbox@
3999   \eq@line@depth@\dp\eq@cellbox@
4000   \eq@numbering@print@line@eval
4001   \if@eqnsw
4002     \eq@tagbox@make\eq@composetag@print
4003   \fi
4004   \eq@tagpos@print@line@eval
4005   \eq@arrange@print@line
4006   \eq@tagpos@print@line@end
4007 }
```

TODO: describe

```
4008 \def\eq@lines@print{%
4009   \eq@verbose@infoarg\eq@verbose@msg@center\eq@lines@print
4010   \eq@arrange@init
4011   \eq@display@halign@init\eq@lines@print@line@begin
4012   \eq@display@halign@letcr\eq@lines@print@line@end
4013   \tabskip\z@skip

4014   \halign{%
4015     \global\let\eq@cell@container\@empty
4016     \setbox\eq@cellbox@\hbox{%
4017       \eq@restore@hfuzz
4018       \eq@strut@cell
4019       \@lign
4020       $\m@th\eq@mathstyle
4021       \eq@hook@colin
```

```

4022      ##%
4023      \eql@punct@apply@col
4024      \eql@hook@colout
4025      \eql@tagging@mathsave
4026      $%
4027      \hfil
4028      \kern\z@
4029      }%
4030      \eql@shape@eval
4031      \eql@cell@container
4032      \eql@lines@print@line@adjust
4033      \crrr

4034      \noalign{%
4035      \eql@display@halign@start
4036      \eql@numbering@print@block@begin
4037      \eql@hook@blockbefore
4038      }%
4039      \eql@hook@blockin
4040      \eql@scan@body
4041      \ifvmode\else
4042      \relax
4043      \eql@punct@apply@block
4044      \eql@hook@blockout
4045      \eql@display@endline
4046      \cr
4047      \fi
4048      \noalign{%
4049      \eql@hook@blockafter
4050      \eql@display@halign@end
4051      \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print
4052      }%
4053      \eql@tagging@tablesavelines
4054      }%
4055      }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```

\eql@columns@add@amp
@columns@completerow
4056 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4057 \def\eql@columns@completerow{%
4058   \count@=\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
4059   \edef\eql@tmp{%
4060     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4061   \eql@tmp
4062 }

4063 \def\eql@columns@overfull{%
4064   \dimen@\eql@line@width@

```

```

4065 \advance\dimen@-\hfuzz
4066 \ifdim\dimen@>\displaywidth
4067 \setbox\z@\hbox to\displaywidth{\hbox to\eq@line@width@{\hfil}}%
4068 \wd\z@\z@
4069 \ht\z@\eq@line@height@
4070 \dp\z@\eq@line@depth@
4071 \box\z@
4072 \fi
4073 }

```

14.2 Transpose

TODO: describe

TODO: describe

```

4074 \let\eq@transpose@active\eq@false
4075 \def\eq@transpose@end{\eq@transpose@end}
4076 \def\eq@transpose@skip{&\eqnpunct{}}
4077 \def\eq@transpose@complete{%
4078 \relax\ifodd\eq@column@\expandafter\eq@transpose@skip\fi&}

```

TODO: describe

```

4079 \def\eq@transpose{%
4080 \eq@totalcolumns@\z@
4081 \eq@totalrows@\z@
4082 \expandafter\eq@transpose@scan@col\the\eq@scan@reg@&\eq@transpose@end&
4083 \eq@scan@reg@{}%
4084 \eq@row@\z@
4085 \eq@transpose@output@row
4086 }

```

TODO: describe

```

4087 \def\eq@transpose@save@col#1{%
4088 \@namedef{eq@transpose@data@col@\the\eq@totalcolumns@}{%
4089 \ifcase\eq@row@#1\else\let\eq@tmp\eq@transpose@skip\fi}}

```

TODO: describe

```

4090 \def\eq@transpose@scan@col#1&{%
4091 \def\eq@tmpa{#1}%
4092 \ifx\eq@tmpa\eq@transpose@end\else
4093 \advance\eq@totalcolumns@\@ne
4094 \eq@row@\z@
4095 \let\eq@transpose@data@col\@empty
4096 \eq@transpose@scan@row#1\\eq@transpose@end\\%
4097 \ifnum\eq@row@>\eq@totalrows@
4098 \eq@totalrows@\eq@row@
4099 \fi
4100 \expandafter\eq@transpose@save@col\expandafter{\eq@transpose@data@col}%
4101 \expandafter\eq@transpose@scan@col
4102 \fi
4103 }

```

TODO: describe

```

4104 \def\eq@transpose@append@row#1{%
4105 \advance\eq@row@\@ne
4106 \eq@append\eq@transpose@data@col{\or\def\eq@tmp{#1}}}

```

TODO: describe

```
4107 \def\eql@transpose@scan@row#1\\{%
4108   \def\eql@tmpa{#1}%
4109   \ifx\eql@tmpa\eql@transpose@end\else
4110     \ifx\eql@transpose@active+
4111       \eql@transpose@scan@cell#1&\eql@transpose@end&%
4112     \else
4113       \eql@transpose@append@row{#1}%
4114     \fi
4115     \expandafter\eql@transpose@scan@row
4116   \fi
4117 }
```

TODO: describe

```
4118 \def\eql@transpose@scan@cell#1&#2{%
4119   \def\eql@tmpa{#2}%
4120   \ifx\eql@tmpa\eql@transpose@end
4121     \eql@transpose@append@row{#1}%
4122   \else
4123     \eql@transpose@append@row{#1&#2}%
4124     \expandafter\eql@transpose@scan@cell@next
4125   \fi
4126 }
```

TODO: describe

```
4127 \def\eql@transpose@scan@cell@next#1&{%
4128   \def\eql@tmpa{#1}%
4129   \ifx\eql@tmpa\eql@transpose@end\else
4130     \eql@transpose@append@row{&#1}%
4131     \expandafter\eql@transpose@scan@cell@next
4132   \fi
4133 }
```

TODO: describe

```
4134 \def\eql@transpose@output@row{%
4135   \ifnum\eql@row@<\eql@totalrows@
4136     \advance\eql@row@\@ne
4137     \eql@column@\z@
4138     \eql@transpose@output@col
4139     \ifnum\eql@row@<\eql@totalrows@
4140       \eql@scan@addto\\%
4141     \fi
4142     \expandafter\eql@transpose@output@row
4143   \fi
4144 }
```

TODO: describe

```
4145 \def\eql@transpose@output@col{%
4146   \ifnum\eql@column@<\eql@totalcolumns@
4147     \advance\eql@column@\@ne
4148     \csname eql@transpose@data@col@\the\eql@column@\endcsname
4149     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4150     \ifnum\eql@column@<\eql@totalcolumns@
4151       \eql@scan@addto{\eql@transpose@complete}%
4152     \fi
4153     \expandafter\eql@transpose@output@col
```

```

4154 \fi
4155 }

```

14.3 Measure

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

```
s@measure@line@begin
```

```

4156 \def\eql@columns@measure@line@begin{%
4157   \eql@verbose@info\eql@verbose@msg@startline
4158   \global\eql@column@\z@
4159   \global\eql@line@height@\z@
4160   \global\eql@line@depth@\z@
4161   \eql@numbering@measure@line@begin
4162   \eql@hook@linein
4163 }

```

```

4164 \def\eql@columns@measure@cell{%
4165   \eql@cellwidth@\wd\eql@cellbox@
4166   \ifdefined\eql@frame@cmd
4167     \eql@frame@measure
4168     \advance\eql@cellwidth@\eql@frame@margin@
4169   \fi
4170   \ifdim\ht\eql@cellbox@>\eql@line@height@
4171     \global\eql@line@height@\ht\eql@cellbox@
4172   \fi
4173   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4174     \global\eql@line@depth@\dp\eql@cellbox@
4175   \fi
4176   \ifnum\eql@column@=\@ne
4177     \eql@dimensions@startrow
4178   \fi
4179   \ifodd\eql@column@
4180     \eql@shape@pos@\tw@
4181   \else
4182     \eql@shape@pos@\z@
4183   \fi
4184   \eql@shape@amount@\z@
4185   \eql@dimensions@savecell
4186   \ifodd\eql@column@\else
4187     \eql@dimensions@savesep
4188   \fi
4189   \kern\eql@cellwidth@
4190 }

```

```
mns@measure@line@end
```

```

4191 \def\eql@columns@measure@line@end{%
4192   \eql@punct@apply@line
4193   \eql@hook@lineout
4194   &\omit
4195   \ifnum\eql@column@>\eql@totalcolumns@
4196     \global\eql@totalcolumns@\eql@column@
4197   \fi

```

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

additional column in width table is accounted for in column table

```

4198 \ifdefined\eqL@frame@cmd
4199 \advance\eqL@column@\@ne
4200 \wd\eqL@cellbox@\z@
4201 \eqL@columns@measure@cell
4202 \fi
4203 \eqL@measure@tag
4204 \eqL@measure@endrow
4205 }

```

\eqL@columns@measure

```

4206 \def\eqL@columns@measure{%
4207 \eqL@verbose@infoarg\eqL@verbose@msg@enter\eqL@columns@measure
4208 \eqL@totalcolumns@\z@
4209 \eqL@measure@init\eqL@columns@measure@line@begin\eqL@columns@measure@line@end

4210 \setbox\z@\vbox{\measuring@true\halign{%
4211 &%
4212 \global\advance\eqL@column@\@ne
4213 \global\let\eqL@cell@container\@empty
4214 \global\setbox\eqL@cellbox@\hbox{%
4215 \eqL@strut@cell
4216 \@lign
4217 $\m@th\eqL@mathstyle
4218 \eqL@hook@colin
4219 ##%
4220 \eqL@punct@apply@next
4221 \eqL@class@innerleft
4222 \eqL@hook@innerleft
4223 $%
4224 }%
4225 \eqL@cell@container
4226 \hfil
4227 \eqL@columns@measure@cell
4228 \global\let\eqL@frame@prevcmd\eqL@frame@cmd
4229 &%
4230 \eqL@prevwidth@\wd\eqL@cellbox@
4231 \let\eqL@frame@cmd\eqL@frame@prevcmd
4232 \global\advance\eqL@column@\@ne
4233 \global\let\eqL@cell@container\@empty
4234 \setbox\eqL@cellbox@\hbox{%
4235 \eqL@strut@cell
4236 \@lign
4237 $\m@th\eqL@mathstyle
4238 \eqL@hook@innerright
4239 \eqL@class@innerright@sel
4240 ##%
4241 \eqL@punct@apply@col
4242 \eqL@hook@colout
4243 $%
4244 }%
4245 \eqL@cell@container
4246 \eqL@columns@measure@cell
4247 \hfil
4248 \crr
4249 \noalign{%

```



```

4250     \eql@hook@blockbefore
4251 }%
4252 \eql@hook@blockin
4253 \eql@scan@body

4254 \ifvmode\else
4255     \eql@punct@apply@block
4256     \eql@hook@blockout
4257     \eql@display@endline
4258     \cr
4259 \fi
4260 \noalign{%
4261     \eql@hook@blockafter
4262 }%

```

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

```

4263     \omit
4264     \eql@column@\@ne
4265     \eql@columns@completerow
4266     \cr
4267 }}%

4268 \eql@measure@close

4269 \setbox\z@\vbox{%
4270     \unvbox\z@
4271     \unpenalty
4272     \global\setbox\@ne\lastbox
4273 }%
4274 \eql@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4275 \let\eql@colwidth@tab\@empty
4276 \loop
4277     \setbox\@ne\hbox{%
4278         \unhbox\@ne
4279         \unskip
4280         \global\setbox\thr@@\lastbox
4281     }%
4282 \ifhbox\thr@@
4283     \eql@colwidth@save{\wd\thr@@}%
4284 \repeat

4285 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4286 }

```

14.4 Columns Placement

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

```

4287 \def\eql@columns@adjust{%
4288     \ifodd\eql@totalcolumns@
4289         \advance\eql@totalcolumns@\@ne
4290     \fi
4291     \eql@tagpos@adjust@eval
4292     \eql@adjust@calc@columns
4293 }

```

14.5 Print

TODO: describe

columns@print@line@begin

```
4294 \def\eql@columns@print@line@begin{%
4295   \eql@verbose@info\eql@verbose@msg@startline
4296   \global\eql@column@\z@
4297   \global\eql@line@pos@\eql@marginleft@
4298   \global\eql@line@width@\z@
4299   \global\eql@line@avail@\eql@totalwidth@
4300   \global\eql@line@height@\z@
4301   \global\eql@line@depth@\z@
4302   \eql@numbering@print@line@begin
4303   \eql@hook@linein
4304 }
```

l@columns@print@cell

```
4305 \def\eql@columns@print@cell{%
4306   \eql@cellwidth@\wd\eql@cellbox@
4307   \ifodd\eql@column@
4308     \ifdefined\eql@frame@cmd
4309       \eql@frame@measure
4310       \advance\eql@cellwidth@\eql@frame@margin@
4311     \fi
4312     \dimen@\z@
4313   \else
4314     \advance\eql@cellwidth@-\eql@prevwidth@
```

draw a frame

```
4315     \ifdefined\eql@frame@cmd
4316       \eql@frame@measure
4317       \advance\eql@cellwidth@\eql@frame@margin@
4318       \advance\eql@prevwidth@\eql@frame@margin@
4319       \eql@frame@print
4320     \fi
```

update height and depth

```
4321     \ifdim\ht\eql@cellbox@>\eql@line@height@
4322       \global\eql@line@height@\ht\eql@cellbox@
4323     \fi
4324     \ifdim\dp\eql@cellbox@>\eql@line@depth@
4325       \global\eql@line@depth@\dp\eql@cellbox@
4326     \fi
```

print box

```
4327     \kern-\eql@prevwidth@
4328     \unhbox\eql@cellbox@
4329     \dimen@-\eql@cellwidth@
4330   \fi
```

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

```
4331   \advance\dimen@\eql@colwidth@get\eql@column@\relax
4332   \kern\dimen@
```

update available and used space

```

4333 \dimen@eql@colwidth@get\eql@column@\relax
4334 \ifdim\eql@cellwidth@>\z@
4335   \ifdim\eql@line@width@=\z@
4336     \eql@line@avail@\eql@line@pos@
4337     \ifodd\eql@column@
4338       \advance\eql@line@avail@\dimen@
4339       \advance\eql@line@avail@-\eql@cellwidth@
4340     \fi
4341     \global\eql@line@avail@\eql@line@avail@
4342   \fi
4343   \eql@line@width@\eql@line@pos@
4344   \ifodd\eql@column@
4345     \advance\eql@line@width@\dimen@
4346   \else
4347     \advance\eql@line@width@\eql@cellwidth@
4348   \fi
4349   \global\eql@line@width@\eql@line@width@
4350 \fi
4351 \advance\eql@line@pos@\dimen@
4352 \ifodd\eql@column@\else
4353   \advance\eql@line@pos@\eql@colsep@
4354 \fi
4355 \global\eql@line@pos@\eql@line@pos@
4356 }

4357 \def\eql@columns@print@trailright{%
4358   &\omit
4359   \eql@prevwidth@\wd\eql@cellbox@
4360   \let\eql@frame@cmd\eql@frame@prevcmd
4361   \global\advance\eql@column@\@ne
4362   \eql@columns@print@cell
4363 }

```

lums@print@line@end

```

4364 \def\eql@columns@print@line@end{%
4365   \eql@punct@apply@line
4366   \eql@hook@lineout
4367 % \TODO add an even column with empty stuff if box processing deferred
4368   \ifodd\eql@column@
4369     \expandafter\eql@columns@print@trailright
4370   \fi
4371   \eql@columns@completerow
4372   \eql@columns@print@tag
4373 }

```

ql@columns@print@tag

```

4374 \def\eql@columns@print@tag{%
4375   \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4376 \eql@display@firstavail@set\eql@line@avail@
4377 \eql@columns@overfull
4378 \eql@numbering@print@line@eval
4379 \if@eqnsw
4380   \eql@tagbox@make\eql@composetag@print

```

```

4381 \fi
4382 \eql@tagpos@print@line@eval
4383 \eql@tagbox@print@cell
4384 \eql@tagpos@print@line@end
4385 }

```

\eql@columns@print

```

4386 \def\eql@columns@print{%
4387 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
4388 \eql@shape@align@disable
4389 \eql@display@halign@init\eql@columns@print@line@begin
4390 \eql@display@halign@letcr\eql@columns@print@line@end
4391 \tabskip\eql@marginleft@

4392 \halign{%
4393   &%
4394   \global\advance\eql@column@ \@ne
4395   \global\let\eql@cell@container \@empty
4396   \global\setbox\eql@cellbox@ \hbox{%
4397     \eql@strut@cell
4398     \@lign
4399     $\m@th\eql@mathstyle
4400     \eql@hook@colin
4401     ##%
4402     \eql@punct@apply@next
4403     \eql@class@innerleft
4404     \eql@hook@innerleft
4405     \eql@tagging@mathsave
4406     $%
4407     \eql@tagging@mathaddlast
4408   }%
4409   \eql@cell@container
4410   \hfil
4411   \eql@columns@print@cell
4412   \global\let\eql@frame@prevcmd\eql@frame@cmd
4413   \tabskip\z@skip
4414   &%
4415   \eql@prevwidth@ \wd\eql@cellbox@
4416   \let\eql@frame@cmd\eql@frame@prevcmd
4417   \global\advance\eql@column@ \@ne
4418   \global\let\eql@cell@container \@empty
4419   \setbox\eql@cellbox@ \hbox{%
4420     \unhbox\eql@cellbox@
4421     \eql@strut@cell
4422     \@lign
4423     $\m@th\eql@mathstyle
4424     \eql@hook@innerright
4425     \eql@class@innerright@sel
4426     ##%
4427     \eql@punct@apply@col
4428     \eql@hook@colout
4429     \eql@tagging@mathsave
4430     $%
4431     \eql@tagging@mathaddlast
4432   }%
4433   \eql@cell@container
4434   \eql@columns@print@cell
4435   \hfil

```

```

4436     \tabskip\eql@colsep@\relax
4437     \crrcr

4438     \noalign{%
4439         \eql@display@halign@start
4440         \eql@numbering@print@block@begin
4441         \eql@hook@blockbefore
4442     }%
4443     \eql@hook@blockin
4444     \eql@scan@body
4445     \ifvmode\else
4446         \relax
4447         \eql@punct@apply@block
4448         \eql@hook@blockout
4449         \eql@display@endline
4450         \cr
4451     \fi
4452     \noalign{%
4453         \eql@hook@blockafter
4454         \eql@display@halign@end
4455         \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
4456     }%
4457     \eql@tagging@tablesalign
4458 }%
4459 }

```

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.

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

```
4460 \newtoks\eql@scan@reg@
```

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

```

4461 \def\eql@scan@body@dump{\the\eql@scan@reg@}
4462 \def\eql@scan@body@rescan{%
4463     \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
4464 \let\eql@scan@body\eql@scan@body@dump

```

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

```
4465 \long\def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}
```

Scan Modifiers at End.

`\eq1@scan@testend` Scan for modifiers following the end of the scanned block:

```
4466 \def\eq1@scan@testend{%
4467   \eq1@ampprotect\eq1@end@testall\eq1@scan@end}
```

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

`\eq1@scan@env` The macro `\eq1@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 `\eq1@scan@reg@`:

```
4468 \def\eq1@scan@env#1{%
4469   \eq1@verbose@infoarg\eq1@verbose@msg@enter\eq1@scan@env
4470   \def\eq1@scan@end{#1\expandafter\end\expandafter{\@currentvir}}%
4471   \eq1@scan@reg@{\def\eq1@scan@stack{b}}%
```

We call `\eq1@scan@env@iterate` which will scan until the next occurrence of `\end` and then count the number of occurrences of `\begin` before `\end` in `\eq1@scan@stack`. If we simply called `\eq1@scan@env@iterate` directly, the error message for an unwanted `\par` token (usually from a blank line) would refer to `\eq1@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 `\eq1@scan@env@iterate`:

```
4472 \edef\eq1@scan@iterate{\expandafter\noexpand\csname\@currentvir\endcsname}%
4473 \expandafter\let\expandafter\eq1@scan@env@org\eq1@scan@iterate
4474 \ifdefined\eq1@scan@par
4475   \expandafter\let\eq1@scan@iterate\eq1@scan@env@iterate
4476 \else
4477   \expandafter\let\eq1@scan@iterate\eq1@scan@env@iterate@nopar
4478 \fi
4479 \eq1@scan@iterate
4480 }
```

`\eq1@scan@env@iterate` `\eq1@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 `\eq1@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:

```
4481 \long\def\eq1@scan@env@iterate#1\end#2{%
4482   \edef\eq1@scan@stack{%
4483     \eq1@scan@env@count#1\begin\end\expandafter\@gobble\eq1@scan@stack}%
4484   \ifx\@empty\eq1@scan@stack
4485     \@checkend{#2}%
4486     \eq1@scan@addto{#1}%
4487     \expandafter\let\eq1@scan@iterate\eq1@scan@env@org
4488     \eq1@verbose@infoarg\eq1@verbose@msg@leave\eq1@scan@env
4489     \expandafter\eq1@scan@testend
4490   \else
4491     \eq1@scan@addto{#1\end{#2}}%
4492     \expandafter\eq1@scan@iterate
4493   \fi
4494 }
```

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

```
4495 \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.

```
4496 \long\def\eql@scan@env@count#1\begin#2{%
4497   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4498 }
```

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

```
4499 \def\eql@scan@env@cancel{%
4500   \@namedef{end\@currentvir}{\ignorespacesafterend}%
4501 }
```

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 `\[`:

```
4502 \def\eql@scan@sqr#1{%
4503   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
4504   \def\eql@scan@end{#1}}%
4505   \eql@scan@reg@{\def\eql@scan@stack{b}}%
4506   \let\eql@scan@sqr@org\[%\]
4507   \ifdefined\eql@scan@par
4508     \let\[\eql@scan@sqr@iterate%\]
4509   \else
4510     \let\[\eql@scan@sqr@iterate@nopar%\]
4511   \fi
4512   \[%\]
4513 }
```

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

```
4514 \long\def\eql@scan@sqr@iterate#1\]{%
4515   \edef\eql@scan@stack{%
4516     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4517   \ifx\@empty\eql@scan@stack
4518     \let\[\eql@scan@sqr@org%\]
4519     \eql@scan@addto{#1}%
4520     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
4521     \expandafter\eql@scan@testend
4522   \else
4523     \eql@scan@addto{#1}}%
4524   \expandafter\[%\]
4525   \fi
4526 }
```

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

```
4527 \def\eql@scan@sqr@iterate@nopar#1\]{\eql@scan@sqr@iterate#1\}}
```

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

```

4528 \long\def\eql@scan@sqr@count#1\[#2{%\]
4529 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4530 }

```

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

```

4531 \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 `\>`:

```

4532 \def\eql@scan@ang#1{%
4533 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
4534 \def\eql@scan@end{#1\>}%
4535 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
4536 \let\eql@scan@ang@org\<%\>
4537 \ifdefined\eql@scan@par
4538 \let\<\eql@scan@ang@iterate%\>
4539 \else
4540 \let\<\eql@scan@ang@iterate@nopar%\>
4541 \fi
4542 \<%\>
4543 }

```

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

```

4544 \long\def\eql@scan@ang@iterate#1\>{%
4545 \edef\eql@scan@stack{%
4546 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4547 \ifx\@empty\eql@scan@stack
4548 \let\<\eql@scan@ang@org%\>
4549 \eql@scan@addto{#1}%
4550 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
4551 \expandafter\eql@scan@testend
4552 \else
4553 \eql@scan@addto{#1\>}%
4554 \expandafter\<%\>
4555 \fi
4556 }

```

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

```

4557 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}

```

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

```

4558 \long\def\eql@scan@ang@count#1\<#2{%\>
4559 \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4560 }

```

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

```

4561 \def\eqlequations@testall{\eql@parseopt@main\eqlequations@parse}
4562 \def\eqlequations@parse{%
4563   \ifx\eql@parseopt@token*%
4564     \let\eql@parseopt@next\eql@parseopt@nonumber
4565   \fi
4566   \ifx\eql@parseopt@token!%
4567     \let\eql@parseopt@next\eql@parseopt@donumber
4568   \fi
4569   \ifx\eql@parseopt@token/%
4570     \let\eql@parseopt@next\eql@parseopt@transpose
4571   \fi
4572   \ifx\eql@parseopt@token[%]
4573     \let\eql@parseopt@next\eql@parseopt@opt
4574   \fi
4575   \ifx\eql@parseopt@token\eql@atxi
4576     \let\eql@parseopt@next\eql@parseopt@label
4577   \fi
4578   \ifx\eql@parseopt@token\eql@atxii
4579     \let\eql@parseopt@next\eql@parseopt@label
4580   \fi
4581   \ifx\eql@parseopt@token.%
4582     \let\eql@parseopt@next\eql@parseopt@punctpass
4583   \fi
4584   \ifx\eql@parseopt@token,%
4585     \let\eql@parseopt@next\eql@parseopt@punctpass
4586   \fi
4587   \ifx\eql@parseopt@token~%
4588     \let\eql@parseopt@next\eql@parseopt@punctpass
4589   \fi
4590   \ifx\eql@parseopt@token'%
4591     \let\eql@parseopt@next\eql@parseopt@punctopt
4592   \fi
4593   \ifx\eql@parseopt@token-%
4594     \let\eql@parseopt@next\eql@parseopt@single
4595   \fi
4596   \ifx\eql@parseopt@token=%
4597     \let\eql@parseopt@next\eql@parseopt@lines
4598   \fi
4599   \ifx\eql@parseopt@token|%
4600     \let\eql@parseopt@next\eql@parseopt@columns
4601   \fi
4602   \ifx\eql@parseopt@token<%
4603     \let\eql@parseopt@next\eql@parseopt@ampeq
4604   \fi
4605   \ifx\eql@parseopt@token>%
4606     \let\eql@parseopt@next\eql@parseopt@eqamp
4607   \fi
4608   \ifx\eql@parseopt@token\label
4609     \let\eql@parseopt@next\eql@parseopt@end
4610   \fi
4611   \ifx\eql@parseopt@token\begin
4612     \let\eql@parseopt@next\eql@parseopt@end
4613   \fi
4614 }
```

`\eql@end@testall` **TODO:** describe

```

4615 \def\eql@end@testall{\eql@parseopt@main\eql@end@parse}
4616 \def\eql@end@parse{%
4617   \ifx\eql@parseopt@token.%
4618     \let\eql@parseopt@next\eql@parseopt@punctpass
4619   \fi
4620   \ifx\eql@parseopt@token,%
4621     \let\eql@parseopt@next\eql@parseopt@punctpass
4622   \fi
4623   \ifx\eql@parseopt@token~%
4624     \let\eql@parseopt@next\eql@parseopt@punctpass
4625   \fi
4626   \ifx\eql@parseopt@token'%
4627     \let\eql@parseopt@next\eql@parseopt@punctblock
4628   \fi
4629 }

```

`\eql@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:

```

4630 \def\eql@equations@processopt{%
4631   \let\eql@tags@container@block\eql@tags@container@clear
4632   \let\eql@tags@frame@cmd\@firstofone
4633   \let\eql@skip@force@above\@undefined
4634   \let\eql@skip@force@below\@undefined
4635   \let\eql@skip@force@leave\@undefined
4636   \let\eql@display@linewidth\@undefined
4637   \let\eql@display@marginleft\@undefined
4638   \let\eql@display@marginright\@undefined
4639   \eql@abovespace@\z@skip
4640   \eql@belowspace@\z@skip
4641   \eql@displaybreak@prepen@\@MM
4642   \eql@displaybreak@postpen@\@MM
4643   \eql@nextopt@process{equations}%
4644   \let\eql@punct@block\eql@punct@main
4645   \let\eql@punct@main\@undefined
4646   \eql@indent@\glueexpr\eql@indent@val\relax
4647   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4648 }

```

15.3 Single-Line Main

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

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

```

4649 \def\eql@single@cr@error{%
4650   \eql@error{Cannot use '\string\\' within display equation.
4651     Please switch to equations environment}%
4652 }

```

`\eql@single@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:

```

4653 \def\eql@single@start{%
4654   \eql@display@enter
4655   \eql@tagging@start
4656   \eql@dollar@dollar@begin
4657   \eql@display@adjust
4658   \eql@numbering@init
4659   \eql@stack@save@equations
4660   \eql@numbering@single@init
4661   \eql@punct@term@set
4662   \ifdefined\eql@single@cr@mode
4663     \let\\\eql@single@cr@mode
4664   \fi
4665   \ifdefined\eql@single@native
4666     \let\eql@single@start@sel\eql@single@start@native
4667     \let\eql@single@end@sel\eql@single@end@native
4668   \else
4669     \let\eql@single@start@sel\eql@single@start@print
4670     \let\eql@single@end@sel\eql@single@end@print
4671   \fi
4672   \eql@single@start@sel
4673 }
```

`\eql@single@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:

```

4674 \def\eql@single@end{%
4675   \eql@punct@apply@block
4676   \eql@hook@eqout
4677   \eql@single@end@sel
4678   \eql@stack@restore
4679   \eql@dollar@dollar@end
4680   \eql@tagging@end
4681   \eql@display@leave
4682 }
```

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

```

4683 \def\eql@single@main{%
4684   \expandafter\eql@single@start
4685   \eql@scan@body
4686   \eql@single@end
4687 }
```

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

```

4688 \def\eql@mode@single{%
4689   \ifdefined\eql@single@doscan
4690     \let\eql@equations@main\eql@single@main
4691     \let\eql@equations@end\@empty
4692   \else
4693     \let\eql@equations@main\@undefined
4694     \let\eql@equations@end\eql@single@end
4695   \fi
4696 }
```

15.4 Multi-Line Main

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

```
4697 \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

```
4698 \def\eql@multi@main{%
4699   \eql@display@enter
4700   \eql@tagging@start
4701   \eql@dollar@dollar@begin
4702   \eql@display@adjust
4703   \eql@numbering@init
4704   \eql@stack@save@equations
4705   \ifdefined\eql@transpose@active
4706     \ifdefined\eql@multi@mode@lines\else
4707       \eql@transpose
4708     \fi
4709   \fi
4710   \ifdefined\eql@numbering@subeq@use
4711     \eql@numbering@subeq@init
4712   \fi
4713   \eql@display@init
4714   \let\intertext\eql@intertext
4715   \let\endintertext\endeql@intertext
4716   \eql@shape@align@enable
```

Now measure the given multi-line equations body:

```
4717   \ifdefined\eql@multi@mode@lines
4718     \eql@lines@measure
4719   \else
4720     \ifdefined\eql@ampproof@active
4721       \eql@ampproof
4722     \fi
4723     \eql@columns@measure
4724   \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`:

```
4725   \ifdefined\eql@numbering@subeq@use
4726     \eql@numbering@subeq@test
4727   \fi
4728   \ifdefined\eql@multi@mode@lines\else
4729     \ifdefined\eql@multi@lines@fallback
4730       \ifnum\eql@totalcolumns@=\@ne
4731         \let\eql@multi@mode@lines\eql@true
4732         \ifx\eql@multi@lines@fallback\z@\else
4733           \eql@lines@measure
4734         \fi
4735       \fi
4736     \fi
4737   \fi
```

Adjust the multi-line equations body:

```
4738 \ifdefined\eq@multi@mode@lines
4739   \eq@lines@adjust
4740 \else
4741   \eq@columns@adjust
4742 \fi
```

Now print the multi-line equations body:

```
4743 \eq@display@print
4744 \eq@numbering@print@init
4745 \ifdefined\eq@multi@mode@lines
4746   \eq@lines@print
4747 \else
4748   \eq@columns@print
4749 \fi
4750 \eq@display@close
```

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

```
4751 \ifdefined\eq@numbering@subeq@use
4752   \eq@numbering@subeq@close
4753 \fi
4754 \eq@stack@restore
4755 \eq@dollar@dollar@end
4756 \eq@tagging@end
4757 \eq@display@leave
4758 }
```

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

```
\eq@mode@lines
4759 \def\eq@mode@columns{%
4760   \let\eq@equations@main\eq@multi@main
4761   \let\eq@equations@end\@empty
4762   \let\eq@multi@mode@lines\eq@false
4763 }
4764 \def\eq@mode@lines{%
4765   \let\eq@equations@main\eq@multi@main
4766   \let\eq@equations@end\@empty
4767   \let\eq@multi@mode@lines\eq@true
4768 }
```

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 `\eq@equations@start`:

```
4769 \newenvironment{equations}{%
4770   \eq@verbose@info\eq@verbose@msg@enterenv
4771   \ifmmode
4772     \eq@error@mathmode{\string\begin{\@currenvir}}%
```

```

4773 \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4774 \else
4775 \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4776 \expandafter\eql@equations@start
4777 \fi
4778 }{
4779 \eql@equations@end
4780 \ignorespacesafterend
4781 \eql@verbose@info\eql@verbose@msg@leaveenv
4782 }
4783 \eql@markline@amsthm@register{equations}
4784 \eql@tagging@register@luamml{equations}

```

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

```

4785 \def\eql@equations@start{%
4786 \eql@equations@processopt
4787 \ifdefined\eql@equations@main
4788 \expandafter\eql@scan@env\expandafter\eql@equations@main
4789 \else
4790 \expandafter\eql@single@start
4791 \fi
4792 }

```

Square Brackets.

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

```

4793 \newenvironment{equations@sqr}{}{}
4794 \eql@markline@amsthm@register{equations@sqr}
4795 \eql@tagging@register@luamml{equations@sqr}

```

`\l@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`:

```

4796 \protected\def\eql@equations@sqr@open{%
4797 \ifmmode
4798 \eql@error@mathmode{\string\[\dots\string\]}%
4799 \expandafter\eql@scan@sqr\expandafter\eql@scan@sqrang@cancel
4800 \else
4801 \expandafter\eqnaddopt\expandafter{\eql@equations@sqr@opt}%
4802 \begin{equations@sqr}%
4803 \eql@verbose@info\eql@verbose@msg@enterenv
4804 \let\]\eql@equations@sqr@close
4805 \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4806 \expandafter\eql@equations@sqr@start
4807 \fi
4808 }

```

`\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`:

```

4809 \def\eql@equations@sqr@start{%
4810 \eql@equations@processopt

```

```

4811 \ifdefined\eql@equations@main
4812   \expandafter\eql@scan@sqr\expandafter\eql@equations@main
4813 \else
4814   \expandafter\eql@single@start
4815 \fi
4816 }

```

`\eql@equations@sqr@close` Definition for ‘\]’. Parse modifiers following ‘\]’ and hand on to `\eql@equations@sqr@end`:

```

4817 \protected\def\eql@equations@sqr@close{%
4818   \eql@ampprotect\eql@end@testall\eql@equations@sqr@end}

```

`\eql@equations@sqr@end` **TODO:** complete End `\[...]` block:

```

4819 \def\eql@equations@sqr@end{%
4820   \eql@equations@end
4821   \eql@verbose@info\eql@verbose@msg@leaveenv
4822   \end{equations@sqr}%
4823   \ignorespaces
4824 }

```

TODO: describe

```

\eql@sqr@open
\eql@sqr@close
4825 \let\eql@sqr@open\eql@equations@sqr@open
4826 \protected\def\eql@sqr@close{%
4827   \eql@error{'\string\]' may only close '\string\[']\}%
4828 }

```

Angle Brackets.

`\eql@equations@ang` (*env.*) Define a pseudo-environment `equations@ang`:

```

4829 \newenvironment{equations@ang}{}{}
4830 \newenvironment{equationsbox@ang}{}{}
4831 \eql@markline@amsthm@register{equations@ang}
4832 \eql@tagging@register@luamml{equations@ang}

```

`\eql@ang@open` Definition for ‘\<’. Forward to `equationsbox` if in math mode, otherwise to `equations`:

```

4833 \protected\def\eql@ang@open{%
4834   \ifmmode
4835     \expandafter\eqnadopt\expandafter{\eql@box@ang@opt}%
4836     \begin{equationsbox@ang}%
4837     \eql@verbose@info\eql@verbose@msg@enterenv
4838     \let\>\eql@box@ang@close
4839     \expandafter\eql@ampprotect\expandafter\eql@box@testall
4840     \expandafter\eql@box@start
4841   \else
4842     \expandafter\eqnadopt\expandafter{\eql@equations@ang@opt}%
4843     \begin{equations@ang}%
4844     \eql@verbose@info\eql@verbose@msg@enterenv
4845     \let\>\eql@equations@ang@close
4846     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4847     \expandafter\eql@equations@ang@start
4848   \fi
4849 }

```

`\eql@ang@close` Definition for ‘>’: **TODO:** NOTE: `\protected` acts as `\relax` and starts a row in `\halign`, so we overwrite `\>` when starting.

```
4850 \protected\def\eql@ang@close{%
4851   \eql@error{'\string\>' may only close '\string\<'}%\>
4852 }
```

`@equations@ang@start` Process arguments and start handling the equation:

```
4853 \def\eql@equations@ang@start{%
4854   \eql@equations@processopt
4855   \ifdefined\eql@equations@main
4856     \expandafter\eql@scan@ang\expandafter\eql@equations@main
4857   \else
4858     \expandafter\eql@single@start
4859   \fi
4860 }
```

`@equations@ang@close` **TODO:** describe

```
4861 \def\eql@equations@ang@close{%
4862   \eql@ampprotect\eql@end@testall\eql@equations@ang@end}
```

`ql@equations@ang@end` **TODO:** describe

```
4863 \def\eql@equations@ang@end{%
4864   \eql@equations@end
4865   \eql@verbose@info\eql@verbose@msg@leaveenv
4866   \end{equations@ang}%
4867   \ignorespaces
4868 }
```

`\eql@box@ang@close` **TODO:** describe

```
4869 \def\eql@box@ang@close{%
4870   \eql@ampprotect\eql@end@testall\eql@box@ang@end}
```

`\eql@box@ang@end` **TODO:** describe

```
4871 \def\eql@box@ang@end{%
4872   \eql@box@end
4873   \eql@verbose@info\eql@verbose@msg@leaveenv
4874   \end{equationsbox@ang}%
4875   \ignorespaces
4876 }
```

16 Options

16.1 Selection Tools

`ql@decide@abovebelow` Select between values ‘above’ or ‘below’ or both: execute the corresponding code provided in the latter two arguments:

```
4877 \def\eql@decide@abovebelow#1#2#3#4#5{%
4878   \eql@decide@select{#1}{#2}{#3}{%
4879     {abovebelow,both,tb}{#4#5},%
4880     {above,top,t}{#4},%
4881     {below,bottom,b}{#5}}}
```


`\eql@decide@situation` Select a particular vertical spacing situation and store it in the macro #4:

```
4882 \def\eql@decide@situation#1#2#3#4{%
4883   \eql@decide@select{#1}{#2}{#3}{%
4884     {{long}}{\def#4{0}}},%
4885     {{short}}{\def#4{1}}},%
4886     {{cont}}{\def#4{2}}},%
4887     {{par}}{\def#4{3}}},%
4888     {{top}}{\def#4{4}}},%
4889     {{noskip}}{\def#4{5}}},%
4890     {{medskip}}{\def#4{6}}}}}
```

`\eql@decide@delim` **TODO:** describe

```
4891 \def\eql@decide@delim#1#2#3{%
4892   \eql@decide@select{#1}{#2}{#3}{%
4893     {{,.,\eql@decide@false}}{\eql@box@wrap{}{}}},%
4894     {{\eql@decide@true,r,round}}{\eql@box@delim()}},%
4895     {{s,sqr,square}}{\eql@box@delim[]{}},%
4896     {{c,curly,braces}}{\eql@box@delim\lbrace\rbrace}},%
4897     {{a,ang,angle}}{\eql@box@delim\langle\rangle}},%
4898     {{v,vert}}{\eql@box@delim\vert\vert}},%
4899     {{d,dvert}}{\eql@box@delim\Vert\Vert}},%
4900     {\relax{\eql@box@delim#3}}}}}
```

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:

```
4901 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{%
4902   \eql@mode@stacked}
4903 \eql@define@key{equationsbox}{aligned,align,al,columns,col}[]{%
4904   \eql@mode@aligned}
4905 \eql@define@key{equationsbox}{cases}[]{%
4906   \eql@mode@cases\eql@box@ldelim\lbrace%
4907   \def\eql@box@colsep{\eql@box@condsep}}
4908 \eql@define@key{equationsbox}{matrix}[r]{%
4909   \eql@mode@stacked\eql@shape@set{center}%
4910   \let\eql@spread@reset\eql@true\def\eql@spread@val{\z@}%
4911   \def\eql@box@colsep{\eql@box@shortsep}%
4912   \let\eql@mathstyle\@empty\eql@punct@clear
4913   \eql@decide@delim{#3}{#2}{#1}}
4914 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
4915 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
4916 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
4917 \eql@define@key{equationsbox}{intro}{%
4918   \def\eql@box@cases@condintro{#1}}
4919 \eql@define@key{equationsbox}{introtext}{%
4920   \def\eql@box@cases@condintro{%
4921     \ifmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}}
4922 \eql@define@key{equationsbox}{textcond}[true]{%
4923   \eql@decide@select{#3}{#2}{#1}{%
4924     {{\eql@decide@true,text}}{\let\eql@box@cases@condtext\eql@true}},%
4925     {{\eql@decide@false,math}}{\let\eql@box@cases@condtext\eql@false}}}
```

```

4926 \eqld@define@key{setup}{boxangopt}[]{%
4927   \def\eqld@box@ang@opt{columns,#1}}

```

Modes for Equations Environment. Declare modes and switches for the equations environment:

```

4928 \eqld@define@key{equations}{equation,eq,single,1}[]{\eqld@mode@single}
4929 \eqld@define@key{equations}{gathered,gather,ga,lines,ln}[]{%
4930   \eqld@mode@lines}
4931 \eqld@define@key{equations}{aligned,align,al,columns,col}[]{%
4932   \eqld@mode@columns}
4933 \eqld@define@key{equations,setup}{transpose}[true]{%
4934   \eqld@decide@select{#3}{#2}{#1}{%
4935     {\eqld@decide@false{\let\eqld@transpose@active\eqld@false}},%
4936     {\noamp,plain,restricted}{\let\eqld@transpose@active\eqld@true}},%
4937     {\eqld@decide@true,amp,cont}{\let\eqld@transpose@active=+}}}%
4938 \eqld@define@key{equations}{native}[true]{%
4939   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native%
4940   \ifdefined\eqld@single@native\let\eqld@layoutleft\eqld@false\fi}
4941 \eqld@define@key{setup}{native}[true]{%
4942   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native}
4943 \eqld@define@key{setup}{scanequation}[true]{%
4944   \eqld@decide@bool{#3}{#2}{#1}\eqld@single@doscan}
4945 \eqld@define@key{setup}{sqropt}[]{%
4946   \def\eqld@equations@sqr@opt{equation,#1}}
4947 \eqld@define@key{setup}{angopt}[]{%
4948   \def\eqld@equations@ang@opt{columns,#1}}

```

Vertical Spacing. Settings concerning the spacing of lines: **TODO:** set at end of env only!

```

4949 \def\eqld@keycat{equations,equationsbox,setup}
4950 \eqld@define@key\eqld@keycat{spread}{%
4951   \let\eqld@spread@reset\eqld@false\def\eqld@spread@val{#1}}
4952 \eqld@define@key\eqld@keycat{spread*}[0pt]{%
4953   \let\eqld@spread@reset\eqld@true\def\eqld@spread@val{#1}}
4954 \eqld@define@key\eqld@keycat{strut}[true]{\eqld@decide@select{#3}{#2}{#1}{%
4955   {\eqld@decide@false{\let\eqld@strut@cell\relax\let\eqld@strut@tag\relax}},%
4956   {{cell}{\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\relax}},%
4957   {{tag}{\let\eqld@strut@cell\relax\let\eqld@strut@tag\eqld@strut}},%
4958   {\eqld@decide@true
4959     {\let\eqld@strut@cell\eqld@strut\let\eqld@strut@tag\eqld@strut}}}}
4960 \eqld@define@key{setup}{strutdepth}{\def\eqld@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

4961 \eqld@define@key\eqld@keycat{displayheight}[strut]{%
4962   \eqld@decide@select{#3}{#2}{#1}{%
4963     {\eqld@decide@false{\let\eqld@display@height\undefined}},%
4964     {{strut}{\def\eqld@display@height{\ht\eqld@strutbox@}}},%
4965     {\relax{\def\eqld@display@height{#1}}}}}%
4966 \eqld@define@key\eqld@keycat{displaydepth}[strut]{%
4967   \eqld@decide@select{#3}{#2}{#1}{%
4968     {\eqld@decide@false{\let\eqld@display@depth\undefined}},%
4969     {{strut}{\def\eqld@display@depth{\dp\eqld@strutbox@}}},%
4970     {\relax{\def\eqld@display@depth{#1}}}}}%

```

Settings concerning page breaks:

```

4971 \eqld@define@key{equations}{prebreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
4972   {{force,4,\eqld@decide@true}{\eqld@displaybreak@pre4}},%
4973   {{high,3}{\eqld@displaybreak@pre3}},%
4974   {{med,medium,2}{\eqld@displaybreak@pre2}},%
4975   {{low,1}{\eqld@displaybreak@pre1}},%
4976   {{0,\eqld@decide@false}{\eqld@displaybreak@pre0}},%
4977   {{default,inherit,-1}{\eqld@displaybreak@pre@m@ne}}}}
4978 \eqld@define@key{equations}{postbreak}[4]{\eqld@decide@select{#3}{#2}{#1}{%
4979   {{force,4,\eqld@decide@true}{\eqld@displaybreak@post4}},%
4980   {{high,3}{\eqld@displaybreak@post3}},%
4981   {{med,medium,2}{\eqld@displaybreak@post2}},%
4982   {{low,1}{\eqld@displaybreak@post1}},%
4983   {{0,\eqld@decide@false}{\eqld@displaybreak@post0}},%
4984   {{default,inherit,-1}{\eqld@displaybreak@post@m@ne}}}}
4985 \eqld@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
4986   \eqld@decide@select{#3}{#2}{#1}{%
4987     {{full,4}{\eqld@displaybreak@inter4}},%
4988     {{high,3}{\eqld@displaybreak@inter3}},%
4989     {{med,medium,2}{\eqld@displaybreak@inter2}},%
4990     {{low,1}{\eqld@displaybreak@inter1}},%
4991     {{0,\eqld@decide@false}{\eqld@displaybreak@inter\z@}}}}
4992 \eqld@define@key{equations}{prepenalty}{%
4993   \eqld@displaybreak@prepen@numexpr#1\relax}
4994 \eqld@define@key{equations}{postpenalty}{%
4995   \eqld@displaybreak@postpen@numexpr#1\relax}
4996 \eqld@define@key{equations,setup}{interpenalty}{%
4997   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

4998 \eqld@define@key{control}{vspace}[]{\eqld@vspace@add{#1}}
4999 \eqld@define@key{control}{vspace*}[]{\eqld@vspace@addfixedbefore{#1}}
5000 \eqld@define@key{control}{vspace!}[]{\eqld@vspace@addfixedafter{#1}}
5001 \eqld@define@key{control}{break}[4]{\eqld@displaybreak@level{#1}}
5002 \eqld@define@key{control}{penalty}[]{\eqld@displaybreak@star{#1}}

```

Override vertical spacing situation: **TODO:** short should just apply to above?! or as far as short would apply...

```

5003 \eqld@define@key{equations}{noskip}[both]{%
5004   \eqld@decide@abovebelow{#3}{#2}{#1}%
5005   {\def\eqld@skip@force@above{5}}%
5006   {\def\eqld@skip@force@below{5}}}
5007 \eqld@define@key{equations}{short}[above]{%
5008   \eqld@decide@abovebelow{#3}{#2}{#1}%
5009   {\def\eqld@skip@force@above{1}}%
5010   {\def\eqld@skip@force@below{1}}}
5011 \eqld@define@key{equations}{long}[both]{%
5012   \eqld@decide@abovebelow{#3}{#2}{#1}%
5013   {\def\eqld@skip@force@above{0}}%
5014   {\def\eqld@skip@force@below{0}}}
5015 \eqld@define@key{equations}{medskip}[both]{%
5016   \eqld@decide@abovebelow{#3}{#2}{#1}%
5017   {\def\eqld@skip@force@above{6}}%
5018   {\def\eqld@skip@force@below{6}}}
5019 \eqld@define@key{equations}{par}[par]{%
5020   \eqld@decide@select{#3}{#2}{#1}{%
5021     {{default,\eqld@decide@false}{\let\eqld@skip@force@leave\undefined}},%
5022     {{cont,hmode}{\let\eqld@skip@force@leave\z@}},%
5023     {{par,vmode}{\let\eqld@skip@force@leave@ne

```

```

5024 \ifdefined\eql@skip@force@below\else
5025 \def\eql@skip@force@below{3}%
5026 \fi}},%
5027 {{top}}{\let\eql@skip@force@leave\tw@
5028 \ifdefined\eql@skip@force@below\else
5029 \def\eql@skip@force@below{4}
5030 \fi}}}}

```

Specify vertical spacing explicitly:

```

5031 \eql@define@key{equations}{skip}{%
5032 \def\eql@skip@force@above{7}%
5033 \def\eql@skip@custom@above{#1}%
5034 \let\eql@skip@force@below\eql@skip@force@above
5035 \let\eql@skip@custom@below\eql@skip@custom@above}
5036 \eql@define@key{equations}{aboveskip}{%
5037 \def\eql@skip@force@above{7}%
5038 \def\eql@skip@custom@above{#1}}
5039 \eql@define@key{equations}{belowskip}{%
5040 \def\eql@skip@force@below{7}%
5041 \def\eql@skip@custom@below{#1}}
5042 \eql@define@key{equations}{abovespace}{%
5043 \advance\eql@abovespace@\glueexpr#1\relax}
5044 \eql@define@key{equations}{belowspace}{%
5045 \advance\eql@belowspace@\glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5046 \eql@define@key{intertext}{skip}{%
5047 \def\eql@skip@force@above{7}%
5048 \def\eql@skip@custom@above{#1}%
5049 \let\eql@skip@force@below\eql@skip@force@above
5050 \let\eql@skip@custom@below\eql@skip@custom@above}
5051 \eql@define@key{intertext}{aboveskip}{%
5052 \def\eql@skip@force@below{7}%
5053 \def\eql@skip@custom@below{#1}}
5054 \eql@define@key{intertext}{belowskip}{%
5055 \def\eql@skip@force@above{7}%
5056 \def\eql@skip@custom@above{#1}}
5057 \eql@define@key{intertext}{noskip}[both]{%
5058 \eql@decide@abovebelow{#3}{#2}{#1}%
5059 {\def\eql@skip@force@below{5}}%
5060 {\def\eql@skip@force@above{5}}}
5061 \eql@define@key{intertext}{short}[both]{%
5062 \eql@decide@abovebelow{#3}{#2}{#1}%
5063 {\def\eql@skip@force@below{1}}%
5064 {\def\eql@skip@force@above{1}}}
5065 \eql@define@key{intertext}{long}[both]{%
5066 \eql@decide@abovebelow{#3}{#2}{#1}%
5067 {\def\eql@skip@force@below{0}}%
5068 {\def\eql@skip@force@above{0}}}
5069 \eql@define@key{intertext}{medskip}[both]{%
5070 \eql@decide@abovebelow{#3}{#2}{#1}%
5071 {\def\eql@skip@force@below{6}}%
5072 {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

5073 \eql@define@key{setup}{skip,longskip}{%
5074 \abovedisplayskip\glueexpr#1\relax
5075 \belowdisplayskip\abovedisplayskip

```

```

5076 \def\eql@skip@long@above{#1}%
5077 \let\eql@skip@long@below\eql@skip@long@above}
5078 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5079 \abovedisplayskip\glueexpr#1\relax
5080 \def\eql@skip@long@above{#1}}
5081 \eql@define@key{setup}{belowskip,belowlongskip}{%
5082 \belowdisplayskip\glueexpr#1\relax
5083 \def\eql@skip@long@below{#1}}
5084 \eql@define@key{setup}{aboveshortskip}{%
5085 \abovedisplayshortskip\glueexpr#1\relax
5086 \def\eql@skip@short@above{#1}}
5087 \eql@define@key{setup}{belowshortskip}{%
5088 \belowdisplayshortskip\glueexpr#1\relax
5089 \def\eql@skip@short@below{#1}}
5090 \eql@define@key{setup}{tagskip}{%
5091 \def\eql@skip@tag@above{#1}%
5092 \let\eql@skip@tag@below\eql@skip@tag@above}
5093 \eql@define@key{setup}{abovetagskip}{%
5094 \def\eql@skip@tag@above{#1}}
5095 \eql@define@key{setup}{belowtagskip}{%
5096 \def\eql@skip@tag@below{#1}}
5097 \eql@define@key{setup}{medskip}{%
5098 \def\eql@skip@med@above{#1}%
5099 \let\eql@skip@med@below\eql@skip@med@above}
5100 \eql@define@key{setup}{abovemedskip}{%
5101 \def\eql@skip@med@above{#1}}
5102 \eql@define@key{setup}{belowmedskip}{%
5103 \def\eql@skip@med@below{#1}}
5104 \eql@define@key{setup}{abovetopskip}{%
5105 \def\eql@skip@top@above{#1}}
5106 \eql@define@key{setup}{belowtopskip}{%
5107 \def\eql@skip@top@below{#1}}
5108 \eql@define@key{setup}{aboveparskip}{%
5109 \def\eql@skip@par@above{#1}}
5110 \eql@define@key{setup}{belowparskip}{%
5111 \def\eql@skip@par@below{#1}}
5112 \eql@define@key{setup}{abovecontskip}{%
5113 \eql@decide@select{#3}{#2}{#1}{%
5114 {hide}{\def\eql@skip@cont@above{\eql@spread@val-\eql@skip@long@below}}},%
5115 {\relax{\def\eql@skip@cont@above{#1}}}}}%
5116 \eql@define@key{setup}{belowcontskip}{%
5117 \def\eql@skip@cont@below{#1}}
5118 \eql@define@key{setup}{shortmode}{%
5119 \eql@decide@select{#3}{#2}{#1}{%
5120 {{\eql@decide@false,never}{\def\eql@skip@mode@short{0}}},%
5121 {{above,neverbelow,belowoff}{\def\eql@skip@mode@short{1}}},%
5122 {{belowone,belowsingle}{\def\eql@skip@mode@short{2}}},%
5123 {{belowall,always,on}{\def\eql@skip@mode@short{3}}}}}%
5124 \eql@define@key{setup}{abovecontmode}{%
5125 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@above}
5126 \eql@define@key{setup}{belowcontmode}{%
5127 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@cont@below}
5128 \eql@define@key{setup}{aboveparmode}{%
5129 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@above}
5130 \eql@define@key{setup}{belowparmode}{%
5131 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@par@below}
5132 \eql@define@key{setup}{abovetopmode}{%
5133 \eql@decide@situation{#3}{#2}{#1}\eql@skip@mode@top@above}

```

```

5134 \eqld@define@key{setup}{belowtopmode}{%
5135   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@below}

```

Labels and Tag Declaration. Specify label and tag for equations and subequations:

```

5136 \def\eqld@keycat{equations,subequations}
5137 \eqld@define@key\eqld@keycat{label}{\eqld@tags@addblock@label{#1}}
5138 \eqld@define@key\eqld@keycat{labelname}{\eqld@tags@addblock@name{#1}}
5139 \eqld@define@key\eqld@keycat{tag}{\eqld@tags@addblock@tag{#1}}
5140 \eqld@define@key\eqld@keycat{tag*}{%
5141   \eqld@tags@addblock@tagform@off\eqld@tags@addblock@tag{#1}}
5142 \eqld@define@key\eqld@keycat{taglabel}{\eqld@tags@addblock@ref{#1}}

```

TODO: describe

```

5143 \eqld@define@key{control}{label}{\eqld@tags@add@label{#1}}
5144 \eqld@define@key{control}{labelname}{\eqld@tags@add@name{#1}}
5145 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
5146 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
5147 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}
5148 \eqld@define@key{control}{shifttag}{\eqld@tags@add@raiseshift{#1}}
5149 \eqld@define@key{control}{smashtag}{\eqld@tags@add@raisesmash{#1}}
5150 \eqld@define@key{control}{pushtag}{\eqld@tags@add@forceraise}

```

TODO: describe

```

5151 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
5152 \eqld@define@key{setup}{autolabel}[true]{%
5153   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
5154 \eqld@define@key{setup}{autotag}[true]{%
5155   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

5156 \def\eqld@keycat{equations,setup}
5157 \eqld@define@key\eqld@keycat{tagmargin}[auto]{%
5158   \eqld@decide@select{#3}{#2}{#1}{%
5159     {\auto,\eqld@decide@false}{\let\eqld@tagmargin@val\@undefined}},%
5160     {\relax{\def\eqld@tagmargin@val{#1}}}}}%
5161 \eqld@define@key\eqld@keycat{tagmargin*}{%
5162   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
5163 \eqld@define@key\eqld@keycat{tagmarginratio}{%
5164   \eqld@tagmargin@ratio@{dimexpr#1pt\relax}}
5165 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
5166   \def\eqld@tagmargin@threshold{#1}}
5167 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5168 \eqld@define@key\eqld@keycat{mintagwidth}{%
5169   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5170 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin{#1}}
5171 \eqld@define@key\eqld@keycat{tagsnap}{%
5172   \eqld@decide@select{#3}{#2}{#1}{%
5173     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5174     {\relax{\def\eqld@tagpos@snap{#1}}}}}%

```

Tag Layout. Configure methods to declare equation tag layout:

```

5175 \def\eqld@keycat{equations,setup}
5176 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
5177   \eqld@tags@taglayout@set{#1}}

```

```

5178 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
5179   \eqld@tags@taglayout@set@direct{#1}}
5180 \eqld@define@key\eqld@keycat{tagform}{%
5181   \eqld@tags@tagform@set@#1}
5182 \eqld@define@key\eqld@keycat{tagform*}{%
5183   \eqld@tags@tagform@set@direct{#1}}
5184 \eqld@define@key\eqld@keycat{subeqtemplate}{%
5185   \def\eqld@subequations@template####1####2{#1}%
5186   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}

5187 \eqld@define@key{control}{tagbox,taglayout}{%
5188   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
5189 \eqld@define@key{control}{tagbox*,taglayout*}{%
5190   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
5191 \eqld@define@key{control}{tagform}{%
5192   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@#1}}
5193 \eqld@define@key{control}{tagform*}{####1}{%
5194   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5195 \def\eqld@keycat{equations,setup}
5196 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%
5197   \eqld@decide@select{#3}{#2}{#1}{%
5198     {\eqld@decide@false,0,*}{\let\eqld@numbering@active\eqld@false}},%
5199     {\eqld@decide@true,!}{\let\eqld@numbering@active\eqld@true}},%
5200     {\none,n,-}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5201       \let\eqld@numbering@active\eqld@false}},%
5202     {\single,1}{\let\eqld@numbering@mode\eqld@numbering@mode@single
5203       \let\eqld@numbering@active\eqld@true}},%
5204     {\multi,@}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5205       \let\eqld@numbering@active\eqld@true}},%
5206     {\relax{\eqld@numbering@set{#1}}}}}}
5207 \eqld@define@key\eqld@keycat{nonumber,nn,*}[]{}%
5208   \let\eqld@numbering@active\eqld@false}
5209 \eqld@define@key\eqld@keycat{donumber,dn,!}[]{}%
5210   \let\eqld@numbering@active\eqld@true}
5211 \eqld@define@key\eqld@keycat{tagsleft,leqno}[]{\let\eqld@tagsleft\eqld@true}
5212 \eqld@define@key\eqld@keycat{tagsright,reqno}[]{\let\eqld@tagsleft\eqld@false}
5213 \eqld@define@key\eqld@keycat{tags,eqno}{%
5214   \eqld@decide@select{#3}{#2}{#1}{%
5215     {\right,r}{\let\eqld@tagsleft\eqld@false}},%
5216     {\left,l}{\let\eqld@tagsleft\eqld@true}}}}
5217 \eqld@define@key\eqld@keycat{evadetag,avoidtag}[true]{%
5218   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
5219 \eqld@define@key\eqld@keycat{tagbetween}[true]{%
5220   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

TODO: describe

```

5221 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
5222 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
5223 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
5224 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

Horizontal Layout. Configure horizontal alignment mode and margin for left alignment:


```

5225 \def\eq@keycat{equations,setup}
5226 \eq@define@key\eq@keycat{layout}{\eq@decide@select{#3}{#2}{#1}{%
5227   {{center,c}}{\let\eq@layoutleft\eq@false}},%
5228   {{left,l}}{\let\eq@layoutleft\eq@true}}}}
5229 \eq@define@key\eq@keycat{center}[]{\let\eq@layoutleft\eq@false}
5230 \eq@define@key\eq@keycat{flushleft,left}[]{\let\eq@layoutleft\eq@true}
5231 \eq@define@key\eq@keycat{leftmargin}{\def\eq@layoutleftmargin{#1}}
5232 \eq@define@key\eq@keycat{leftmargin*}{%
5233   \settowidth\dimen@{#1}\edef\eq@layoutleftmargin{\the\dimen@}}
5234 \eq@define@key\eq@keycat{minleftmargin}{%
5235   \def\eq@layoutleftmarginmin{#1}}
5236 \eq@define@key\eq@keycat{maxleftmargin}{%
5237   \eq@decide@select{#3}{#2}{#1}{%
5238     {\eq@decide@false{\def\eq@layoutleftmarginmax{.5\maxdimen}}},%
5239     {\relax{\def\eq@layoutleftmarginmax{#1}}}}}

5240 \def\eq@keycat{equations,equationsbox}
5241 \eq@define@key\eq@keycat{margin}{%
5242   \def\eq@display@marginleft{#1}\def\eq@display@marginright{#1}}
5243 \eq@define@key\eq@keycat{marginleft}{\def\eq@display@marginleft{#1}}
5244 \eq@define@key\eq@keycat{marginright}{\def\eq@display@marginright{#1}}
5245 \eq@define@key\eq@keycat{linewidth,width}{\def\eq@display@linewidth{#1}}

```

Horizontal Spacing and Columns. Configure column spacing and compression threshold:

```

5246 \def\eq@keycat{equations,setup}
5247 \eq@define@key\eq@keycat{alignshrink}{\eq@decide@select{#3}{#2}{#1}{%
5248   {{max,full,4}}{\eq@alignbadness@inf@bad}},%
5249   {{high,3}}{\eq@alignbadness@54\relax}},%
5250   {{med,medium,2}}{\eq@alignbadness@18\relax}},%
5251   {{low,1}}{\eq@alignbadness@6\relax}},%
5252   {{0,\eq@decide@false}}{\eq@alignbadness@z@}}}}
5253 \eq@define@key\eq@keycat{tagshrink}{\eq@decide@select{#3}{#2}{#1}{%
5254   {{max,full,4}}{\eq@tagbadness@inf@bad}},%
5255   {{high,3}}{\eq@tagbadness@54\relax}},%
5256   {{med,medium,2}}{\eq@tagbadness@18\relax}},%
5257   {{low,1}}{\eq@tagbadness@6\relax}},%
5258   {{0,\eq@decide@false}}{\eq@tagbadness@z@}}}}
5259 \eq@define@key\eq@keycat{alignbadness}{\eq@alignbadness@numexpr#1\relax}
5260 \eq@define@key\eq@keycat{tagbadness}{\eq@tagbadness@numexpr#1\relax}
5261 \eq@define@key\eq@keycat{mincolsep}{\eq@decide@select{#3}{#2}{#1}{%
5262   {{0,\eq@decide@false}}{\def\eq@colsepmin@val{0pt}}},%
5263   {\relax{\def\eq@colsepmin@val{#1}}}}}
5264 \eq@define@key\eq@keycat{maxcolsep}{\eq@decide@select{#3}{#2}{#1}{%
5265   {\eq@decide@false{\def\eq@colsepmax@val{.5\maxdimen}}},%
5266   {\relax{\def\eq@colsepmax@val{#1}}}}}
5267 \eq@define@key\eq@keycat{fulllength}[true]{%
5268   \eq@decide@bool{#3}{#2}{#1}\eq@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5269 \eq@define@key\eq@keycat{linesep}{\eq@decide@select{#3}{#2}{#1}{%
5270   {{0,\eq@decide@false}}{\def\eq@break@line@sep{0pt}}},%
5271   {\relax{\def\eq@break@line@sep{#1}}}}}
5272 \eq@define@key\eq@keycat{linesep*}{\eq@decide@select{#3}{#2}{#1}{%
5273   {{0,\eq@decide@false}}{\def\eq@break@line@shortsep{0pt}}},%
5274   {\relax{\def\eq@break@line@shortsep{#1}}}}}
5275 \eq@define@key\eq@keycat{equationsbox,setup}{colsep}{\eq@decide@select{#3}{#2}{#1}{%

```



```

5276     {{0,\eql@decide@false}}{\def\eql@box@colsep{0pt}}},%
5277     {{short}}{\def\eql@box@colsep{\eql@box@shortsep}}},%
5278     {\relax{\def\eql@box@colsep{#1}}}}}%
5279     \let\eql@break@col@sep\eql@box@colsep}
5280 \eql@define@key{equations}{colsep}{\eql@decide@select{#3}{#2}{#1}}{%
5281     {{0,\eql@decide@false}}{\def\eql@break@col@sep{0pt}}},%
5282     {\relax{\def\eql@break@col@sep{#1}}}}}%
5283     \let\eql@colsepmin@val\eql@box@colsep
5284     \let\eql@colsepmax@val\eql@box@colsep
5285     \let\eql@box@colsep\eql@break@col@sep}
5286 \eql@define@key\eql@keycat{colsep*}{\eql@decide@select{#3}{#2}{#1}}{%
5287     {{0,\eql@decide@false}}{\def\eql@break@col@shortsep{0pt}}},%
5288     {\relax{\def\eql@break@col@shortsep{#1}}}}}%
5289 \eql@define@key{equationsbox,setup}{colsep*}{\eql@decide@select{#3}{#2}{#1}}{%
5290     {{0,\eql@decide@false}}{\def\eql@box@shortsep{0pt}}},%
5291     {\relax{\def\eql@box@shortsep{#1}}}}}%
5292 \eql@define@key{equationsbox,setup}{condsep}{\eql@decide@select{#3}{#2}{#1}}{%
5293     {{0,\eql@decide@false}}{\def\eql@box@condsep{0pt}}},%
5294     {\relax{\def\eql@box@condsep{#1}}}}}%

```

Horizontal Shape. Configure horizontal alignment schemes:

```

5295 \def\eql@keycat{equations,equationsbox,setup}
5296 \eql@define@key\eql@keycat{shape}[default]{\eql@shape@set{#1}}
5297 \eql@define@key\eql@keycat{padding,pad}[indent]{%
5298     \eql@decide@select{#3}{#2}{#1}}{%
5299     {{max}}{\let\eql@paddingleft@val\@undefined}},%
5300     {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5301     {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5302     {\relax{\def\eql@paddingleft@val{#1}}}}}%
5303     \let\eql@paddingright@val\eql@paddingleft@val}
5304 \eql@define@key\eql@keycat{padleft}[indent]{%
5305     \eql@decide@select{#3}{#2}{#1}}{%
5306     {{max}}{\let\eql@paddingleft@val\@undefined}},%
5307     {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5308     {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5309     {\relax{\def\eql@paddingleft@val{#1}}}}}%
5310 \eql@define@key\eql@keycat{padright}[indent]{%
5311     \eql@decide@select{#3}{#2}{#1}}{%
5312     {{max}}{\let\eql@paddingright@val\@undefined}},%
5313     {{indent}}{\def\eql@paddingright@val{\eql@indent@val}}},%
5314     {{0,\eql@decide@false}}{\def\eql@paddingright@val{0pt}}},%
5315     {\relax{\def\eql@paddingright@val{#1}}}}}%
5316 \eql@define@key\eql@keycat{indent}[2em]{%
5317     \def\eql@indent@val{#1}}

```

TODO: describe

```

5318 \eql@define@key{control}{align}[]{%
5319     \eql@decide@select{#3}{#2}{#1}}{%
5320     {{l,left}}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}}},%
5321     {{c,center}}{\global\eql@append\eql@cell@container{\eql@shape@pos@one}}},%
5322     {{r,right}}{\global\eql@append\eql@cell@container{\eql@shape@pos@tw}}}}}%
5323 \eql@define@key{control}{shift,shiftto}[]{%
5324     \eql@decide@select{#3}{#2}{#1}}{%
5325     {{*,indent}}{\eql@shape@alignamount@set{\eql@indent@val}}},%
5326     {{!,outdent}}{\eql@shape@alignamount@set{-\eql@indent@val}}},%
5327     {\relax{\eql@shape@alignamount@set{#1}}}}}%
5328 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

Math Classes at Alignment. Configure math classes at alignment marker:

```

5329 \def\eq@keycat{equations,equationsbox,setup}
5330 \eq@define@key\eq@keycat{classout}{\eq@class@innerleft@set{#1}}
5331 \eq@define@key\eq@keycat{classin}{\eq@class@innerright@set{#1}}
5332 \eq@define@key\eq@keycat{classlead,classin*}{\eq@class@innerlead@set{#1}}
5333 \eq@define@key\eq@keycat{ampeq}[]{\eq@class@ampeq}
5334 \eq@define@key\eq@keycat{eqamp}[]{\eq@class@eqamp}
5335 \eq@define@key\eq@keycat{class}{\eq@decide@select{#3}{#2}{#1}{%
5336   {\ampeq,amprel,eqafter,beforerel}\eq@class@ampeq},%
5337   {\eqamp,relamp,eqbefore,afterrel}\eq@class@eqamp}}}
```

Math Styles. Configure math classes at alignment marker:

```

5338 \eq@define@key\eq@keycat{style}[display]{%
5339   \eq@decide@select{#3}{#2}{#1}{%
5340     {\text,\eq@decide@false}{\let\eq@mathstyle\@empty}},%
5341     {\display,\eq@decide@true}{\let\eq@mathstyle\displaystyle}}}
```

Punctuation. Configure punctuation defaults: **TODO:** describe

```

5342 \def\eq@punct@all#1#2#3#4#5\eq@punct@end{%
5343   \def\eq@tmp{#4}\def\eq@tmpa{1}%
5344   \ifx\eq@tmp\eq@tmpa
5345     \ifnum#5=1111\relax
5346       \eq@punct@set\eq@punct@col{#1}%
5347       \eq@punct@set\eq@punct@line{#2}%
5348       \eq@punct@set\eq@punct@main{#3}%
5349     \else\ifnum#5=111\relax
5350       \eq@punct@set\eq@punct@line{#1}%
5351       \eq@punct@set\eq@punct@main{#2}%
5352     \else\ifnum#5=11\relax
5353       \eq@punct@set\eq@punct@main{#1}%
5354     \else
5355       \eq@punct@clear
5356     \fi\fi\fi
5357   \else
5358     \eq@error{Too many arguments to punctall}%
5359   \fi
5360 }
```

TODO: describe

```

5361 \def\eq@keycat{equations,equationsbox,setup}
5362 \eq@define@key\eq@keycat{punctsep}[\,]{\def\eq@punct@sep{#1}}
5363 \eq@define@key\eq@keycat{punct}[\,]{\eq@punct@set\eq@punct@main{#1}}
5364 \eq@define@key\eq@keycat{punct*}[]{\eq@punct@set\eq@punct@main\relax}
5365 \eq@define@key\eq@keycat{punctline}[\,]{\eq@punct@set\eq@punct@line{#1}}
5366 \eq@define@key\eq@keycat{punctline*}[]{\eq@punct@set\eq@punct@line\relax}
5367 \eq@define@key\eq@keycat{punctcol}[\,]{\eq@punct@set\eq@punct@col{#1}}
5368 \eq@define@key\eq@keycat{punctcol*}[]{\eq@punct@set\eq@punct@col\relax}
5369 \eq@define@key\eq@keycat{punctall}[,,.]{\eq@punct@all#111111\eq@punct@end}

5370 \eq@define@key{control}{punctsep}[\,]{\def\eq@punct@sep{#1}}
5371 \eq@define@key{control}{punct}[\,]{\eq@punct@set\eq@punct@next{#1}}
5372 \eq@define@key{control}{punct*}[]{\eq@punct@set\eq@punct@block\relax}
5373 \eq@define@key{control}{punctapply}[]{\eqnpunctapply}
```

Frames. **TODO:** describe

```

5374 \eqld@define@key{equationsbox}{frame}[\fbox]{%
5375   \def\eql@box@frame{#1}%
5376   \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
5377 \eqld@define@key{equationsbox}{wrap}{\eql@box@wrap#1}
5378 \eqld@define@key{equationsbox}{delim}[r]{\eqld@decide@delim{#3}{#2}{#1}}
5379 \eqld@define@key{equationsbox}{ldelim}{\eql@box@ldelim#1}
5380 \eqld@define@key{equationsbox}{rdelim}{\eql@box@rdelim#1}
5381 \eqld@define@key{equationsbox}{lbrace}[]{\eql@box@ldelim\lbrace}
5382 \eqld@define@key{equationsbox}{rbrace}[]{\eql@box@rdelim\rbrace}
5383 \eqld@define@key{equationsbox}{lrbrace,lrbraces}[]{\eql@box@delim\lbrace\rbrace}
5384 \eqld@define@key{eqld@keycat{braces}}[lr]{%
5385   \eqld@decide@select{#3}{#2}{#1}{%
5386     {\eqld@decide@false}{\eql@box@wrap}{}}},%
5387     {\l,left}{\eql@box@ldelim\lbrace}},%
5388     {\r,right}{\eql@box@rdelim\rbrace}},%
5389     {\eqld@decide@true,lr,both}{\eql@box@delim\lbrace\rbrace}}}
```

TODO: describe

```

5390 \eqld@define@key{control}{framecell}[\fbox]{%
5391   \global\eqld@append\eqld@cell@container{\def\eqld@frame@cmd{#1}}
5392 \eqld@define@key{control}{frametag}[\fbox]{%
5393   \global\eqld@append\eqld@tags@container{\def\eqld@tags@frame@cmd{#1}}}
```

Alternative Content Description. Alternative content description for accessibility or documentation purposes: **TODO:** implement in PDF tagging

```

5394 \eqld@define@key{equations,equationsbox}{alt}{}
```

Injects.

```

5395 \eqld@define@key{control}{inject}{%
5396   \global\eqld@append\eqld@interline@container{%
5397     \eqld@append\eqld@display@injectbefore{#1}}
5398 \eqld@define@key{control}{inject*}{%
5399   \global\eqld@append\eqld@interline@container{%
5400     \eqld@append\eqld@display@injectafter{#1}}
5401 \eqld@define@key{control}{markline}[]{\eqld@markline@inject{#1}}
5402 \eqld@define@key{control}{markline*}[]{\eqld@markline@inject{push,#1}}
5403 \eqld@define@key{control}{qed}[]{\eqld@markline@inject{qed,#1}}
5404 \eqld@define@key{control}{qed*}[]{\eqld@markline@inject{qed,push,#1}}}
```

TODO: describe

```

5405 \eqld@define@key{markline}{pos}{%
5406   \eqld@decide@select{#3}{#2}{#1}{%
5407     {{below,push}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5408     {{baseline}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5409     {{bottom}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}
5410 \eqld@define@key{markline}{below,push}[]{%
5411   \let\eqld@markline@pos\eqld@markline@pos@below}
5412 \eqld@define@key{markline}{baseline}[]{%
5413   \let\eqld@markline@pos\eqld@markline@pos@baseline}
5414 \eqld@define@key{markline}{bottom}[]{%
5415   \let\eqld@markline@pos\eqld@markline@pos@bottom}
5416 \eqld@define@key{markline}{shift}{\def\eqld@markline@shift{#1}}
5417 \eqld@define@key{markline}{symbol}{\def\eqld@markline@symbol{#1}}}
```

```

5418 \eqld@define@key{markline}{qed}[]{\let\eqld@markline@symbol\eqld@markline@qed}
5419 \eqld@define@key{setup}{marksymbol}{\def\eqld@markline@symbol{#1}}
5420 \eqld@define@key{setup}{qedsymbol}{\def\eqld@markline@qed{#1}}
5421 \eqld@define@key{setup}{markpos}{%
5422   \eqld@decide@select{#3}{#2}{#1}{%
5423     {{below}}{\let\eqld@markline@pos\eqld@markline@pos@below}},%
5424     {{baseline}}{\let\eqld@markline@pos\eqld@markline@pos@baseline}},%
5425     {{bottom}}{\let\eqld@markline@pos\eqld@markline@pos@bottom}}}}

```

Global Switches. Set global switches:

```

5426 \let\eqld@multi@linesfallback\eqld@false
5427 \let\eqld@scan@par\eqld@false
5428 \let\eqld@single@cr@mode\eqld@false
5429 \let\eqld@ampproof@active\eqld@false
5430 \let\eqld@parseopt@warn@main\eqld@warn@parseopt
5431 \let\eqld@parseopt@warn@aux\@empty

5432 \eqld@define@key{equations,setup}{linesfallback}[true]{%
5433   \eqld@decide@select{#3}{#2}{#1}{%
5434     {\eqld@decide@false{\let\eqld@multi@linesfallback\eqld@false}},%
5435     {{reuse,lean}}{\let\eqld@multi@linesfallback\z@}},%
5436     {{measure,full,\eqld@decide@true}}{\let\eqld@multi@linesfallback\eqld@true}}}}
5437 \eqld@define@key{setup}{ampproof}[true]{%
5438   \eqld@decide@bool{#3}{#2}{#1}\eqld@ampproof@active}
5439 \eqld@define@key{equations,setup}{equationcr}{%
5440   \eqld@decide@select{#3}{#2}{#1}{%
5441     {\eqld@decide@false{\let\eqld@single@cr@mode\eqld@false}},%
5442     {{\eqld@decide@true,break}}{\let\eqld@single@cr@mode\eqld@break@line}},%
5443     {{error,verbose}}{\let\eqld@single@cr@mode\eqld@single@cr@error}}}}
5444 \eqld@define@key{setup}{modifierwarning}[true]{%
5445   \eqld@decide@select{#3}{#2}{#1}{%
5446     {\eqld@decide@false{\let\eqld@parseopt@warn@main\@empty
5447       \let\eqld@parseopt@warn@aux\@empty}},%
5448     {{-}}{\let\eqld@parseopt@warn@main\eqld@warn@parseopt
5449       \let\eqld@parseopt@warn@aux\@empty}},%
5450     {\eqld@decide@true{\let\eqld@parseopt@warn@main\eqld@warn@parseopt
5451       \let\eqld@parseopt@warn@aux\eqld@warn@parseopt}},%
5452     {{verbose,+}}{\let\eqld@parseopt@warn@main\eqld@warn@parseopt@verbose
5453       \let\eqld@parseopt@warn@aux\eqld@warn@parseopt@verbose}}}}
5454 \eqld@define@key{equations,setup}{rescan}[true]{%
5455   \eqld@decide@if{#3}{#2}{#1}%
5456   {\let\eqld@scan@body\eqld@scan@body@rescan}%
5457   {\let\eqld@scan@body\eqld@scan@body@dump}}
5458 \eqld@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5459   \eqld@decide@bool{#3}{#2}{#1}\eqld@scan@par}
5460 \eqld@define@key{setup}{defaults}{%
5461   \eqld@decide@select{#3}{#2}{#1}{%
5462     {{classic}}{\eqld@defaults@classic}},%
5463     {{eqnlines}}{\eqld@defaults@eqnlines}}}}
5464 \eqld@define@key{equations,equationsbox,setup}{verbose}[true]{%
5465   \eqld@decide@if{#3}{#2}{#1}\eqld@verbose@on\eqld@verbose@off}

```

Package Options. Declare choices available at loading of package only: **TODO:** adjust

```

5466 \let\eqld@provide@opt@env\tw@
5467 \let\eqld@provide@opt@amsmathpatch\eqld@false
5468 \let\eqld@provide@opt@backup\eqld@false

```

```

5469 \let\eql@provide@opt@ang\eql@true
5470 \let\eql@provide@opt@eqref\eql@true
5471 \let\eql@provide@opt@matrix\eql@true

5472 \eql@define@key{setup}{amsmathends,amsmathpatch}[true]{%
5473   \eql@error@packageoption{#2}%
5474   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathpatch}
5475 \eql@define@key{setup}{backup}[true]{%
5476   \eql@error@packageoption{#2}%
5477   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
5478 \eql@define@key{setup}{env}[equation]{%
5479   \eql@error@packageoption{#2}%
5480   \eql@decide@select{#3}{#2}{#1}{%
5481     {none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
5482     {equation,latex}{\let\eql@provide@opt@env\@ne}},%
5483     {amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}%
5484 \eql@define@key{setup}{ang}[true]{%
5485   \eql@error@packageoption{#2}%
5486   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
5487 \eql@define@key{setup}{eqref}[true]{%
5488   \eql@error@packageoption{#2}%
5489   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}
5490 \eql@define@key{setup}{matrix}[true]{%
5491   \eql@error@packageoption{#2}%
5492   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@matrix}

```

Shortcut Options. **TODO:** describe

```

5493 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5494 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5495 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5496 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5497 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
5498 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
5499 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5500 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5501 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5502 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5503 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
5504 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
5505 \def\eql@parseopt@punctnext#1#2{%
5506   \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
5507 \def\eql@parseopt@punctblock#1#2{%
5508   \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
5509 \def\eql@parseopt@vspace[#1]{%
5510   \advance\eql@vspaceskip@glueexpr#1\relax\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:

```

5511 \def\eql@defaults@classic{%

```

```

5512 \eqnlineset{numberline=all}%
5513 \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5514 \eqnlineset{maxcolsep=off}%
5515 \eqnlineset{spread={\jot}}%
5516 \eqnlineset{tagmargin}%
5517 \eqnlineset{tagmarginratio=1}%
5518 \eqnlineset{tagmarginthreshold=0.5}%
5519 \eqnlineset{leftmargin={\leftmargini}}%
5520 \eqnlineset{padding=max}%
5521 \eqnlineset{evadetag=off}%
5522 \eqnlineset{displayheight=off}%
5523 \eqnlineset{displaydepth=off}%
5524 \eqnlineset{shortmode=belowsingle}%
5525 \eqnlineset{abovecontmode=short}%
5526 \eqnlineset{belowcontmode=short}%
5527 \eqnlineset{aboveparmode=long}%
5528 \eqnlineset{belowparmode=long}%
5529 \eqnlineset{abovetopmode=long}%
5530 \eqnlineset{belowtopmode=long}%
5531 \eqnlineset{abovelongskip={\abovedisplayskip}}%
5532 \eqnlineset{belowlongskip={\belowdisplayskip}}%
5533 \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
5534 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5535 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5536 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5537 \eqnlineset{abovecontskip=0pt}%
5538 \eqnlineset{belowcontskip=0pt}%
5539 \eqnlineset{aboveparskip=0pt}%
5540 \eqnlineset{belowparskip=0pt}%
5541 \eqnlineset{abovetopskip=0pt}%
5542 \eqnlineset{belowtopskip=0pt}%
5543 \eqnlineset{abovetagskip=0pt}%
5544 \eqnlineset{belowtagskip=0pt}%
5545 \eqnlineset{equationcr=off}%
5546 \eqnlineset{linesfallback=false}%
5547 }

```

values based on 10pt vs 12pt

`\ql@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):

```

5548 \def\ql@defaults@eqnlines{%
5549   \eqnlineset{numberline=all}%
5550   \eqnlineset{mintagsep=.5em}%
5551   \eqnlineset{maxcolsep=2em}%
5552   \eqnlineset{spread={0.2\normalbaselineskip}}%
5553   \eqnlineset{tagmargin}%
5554   \eqnlineset{tagmarginratio=.334}%
5555   \eqnlineset{tagmarginthreshold=0.5}%
5556   \eqnlineset{leftmargin={\leftmargini}}%
5557   \eqnlineset{padding=0pt}%
5558   \eqnlineset{evadetag}%
5559   \eqnlineset{displayheight=strut}%
5560   \eqnlineset{displaydepth=strut}%

```

```

5561 \eqnlineset{shortmode=above}%
5562 \eqnlineset{abovecontmode=noskip}%
5563 \eqnlineset{belowcontmode=long}%
5564 \eqnlineset{aboveparmode=long}%
5565 \eqnlineset{belowparmode=long}%
5566 \eqnlineset{abovetopmode=noskip}%
5567 \eqnlineset{belowtopmode=long}%
5568 \eqnlineset{longskip={0.75\normalbaselineskip
5569   plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5570 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5571   plus 0.25\normalbaselineskip}}%
5572 \eqnlineset{belowshortskip={0.0\normalbaselineskip
5573   plus 0.25\normalbaselineskip}}%
5574 \eqnlineset{medskip={0.4\normalbaselineskip
5575   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5576 \eqnlineset{abovecontskip=0pt}%
5577 \eqnlineset{belowcontskip=0pt}%
5578 \eqnlineset{aboveparskip=0pt}%
5579 \eqnlineset{belowparskip=0pt}%
5580 \eqnlineset{abovetopskip=0pt}%
5581 \eqnlineset{belowtopskip=0pt}%
5582 \eqnlineset{abovetagskip={0.25\normalbaselineskip
5583   minus 0.25\normalbaselineskip}}%
5584 \eqnlineset{belowtagskip={0.25\normalbaselineskip
5585   minus 0.25\normalbaselineskip}}%
5586 \eqnlineset{equationcr=break}%
5587 \eqnlineset{linesfallback=true}%
5588 }

```

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:

```

\eql@provide@moveenv
@provide@undefinecmd
@provide@undefineenv
5589 \def\eql@provide@movecmd#1#2{%
5590   \eql@letcs{#1\expandafter}\csname#2\endcsname
5591 }
5592 \def\eql@provide@moveenv#1#2{%
5593   \eql@provide@movecmd{#1}{#2}%
5594   \ifcsname end#2\endcsname
5595     \eql@provide@movecmd{end#1}{end#2}%
5596   \fi
5597 }
5598 \def\eql@provide@undefinecmd#1{%
5599   \eql@letcs{#1}\undefined
5600 }
5601 \def\eql@provide@undefineenv#1{%
5602   \eql@provide@undefinecmd{#1}%
5603   \eql@provide@undefinecmd{end#1}%
5604 }

```

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:

```
5605 \def\eql@amsmath@endfix#1#2{%
5606   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5607   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5608     \eql@provide@movecmd{end#1}{end#2}%
5609   \fi
5610 }
```

`\eql@amsmath@fixmatrix` **TODO:** describe

`amsmath@fixmatrixend`

```
5611 \def\eql@amsmath@fixmatrix#1{%
5612   \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
5613   \begingroup
5614     \let\matrix@check\@gobble
5615     \def\env@matrix{\noexpand\env@matrix}%
5616     \def\env@cases{\noexpand\env@cases}%
5617     \global\edef\eql@tmp{\eql@tmp}%
5618   \endgroup
5619   \eql@letcs{#1}\eql@tmp
5620 }
5621 \def\eql@amsmath@fixmatrixend#1{%
5622   \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
5623   \begingroup
5624     \expandafter\def\expandafter\endmatrix\expandafter{%
5625       \expandafter\unexpanded\expandafter{\endmatrix}}%
5626     \global\long\edef\eql@tmp{\eql@tmp}%
5627   \endgroup
5628   \eql@letcs{end#1}\eql@tmp
5629 }
```

`\eql@amsmath@fixends` Perform the replacement for all amsmath environments whenever amsmath is loaded:

```
5630 \def\eql@amsmath@fixends{%
5631   \eql@amsmath@after{%
5632     \eql@amsmath@endfix{flalign}{align}%
5633     \eql@amsmath@endfix{alignat}{align}%
5634     \eql@amsmath@endfix{xalignat}{align}%
5635     \eql@amsmath@endfix{xxalignat}{align}%
5636     \eql@amsmath@endfix{gather*}{gather}%
5637     \eql@amsmath@endfix{multline*}{multline}%
5638     \eql@amsmath@endfix{align*}{align}%
5639     \eql@amsmath@endfix{flalign*}{align}%
5640     \eql@amsmath@endfix{alignat*}{align}%
5641     \eql@amsmath@endfix{xalignat*}{align}%
5642     \eql@amsmath@endfix{gathered}{aligned}%
5643     \eql@amsmath@endfix{alignedat}{aligned}%
5644   }
5645 }
```


`@amsmath@fixmatrices` Perform the replacement for all amsmath environments whenever amsmath is loaded:

```
5646 \def\eq@amsmath@fixmatrices{%
5647   \eq@amsmath@after{%
5648     \eq@amsmath@fixmatrix{cases}%
5649     \eq@amsmath@fixmatrix{matrix}%
5650     \eq@amsmath@fixmatrix{pmatrix}%
5651     \eq@amsmath@fixmatrixend{pmatrix}%
5652     \eq@amsmath@fixmatrixend{bmatrix}%
5653     \eq@amsmath@fixmatrixend{Bmatrix}%
5654     \eq@amsmath@fixmatrixend{vmatrix}%
5655     \eq@amsmath@fixmatrixend{Vmatrix}%
5656   }
5657 }
```

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

```
5658 \def\eq@provide@backup@amsenv#1{%
5659   \eq@amsmath@after{%
5660     \eq@provide@moveenv{ams#1}{#1}%
5661     \eq@tagging@register@luamml{ams#1}%
5662     \eq@markline@amsthm@move{ams#1}{#1}%
5663   }%
5664 }
```

`provide@backup@amsbox` **TODO:** describe

```
5665 \def\eq@provide@backup@amsbox#1{%
5666   \eq@amsmath@after{%
5667     \eq@provide@moveenv{ams#1}{#1}%
5668   }%
5669 }
```

`provide@backup@eqref` Copy an eqref to amseqref whenever amsmath is loaded:

```
5670 \def\eq@provide@backup@eqref{%
5671   \eq@amsmath@after{%
5672     \eq@provide@movecmd{amseqref}{eqref}%
5673   }%
5674 }
```

`ide@backup@multlined` The environment `multlined` is supplied by `mathtools`. We copy it to `amsmultlined` anyway, but whenever `mathtools` is loaded:

```
5675 \def\eq@provide@backup@multlined{%
5676   \AddToHook{package/mathtools/after}{%
5677     \eq@provide@moveenv{amsmultlined}{multlined}}%
5678 }
```

`vide@backup@equation` The \LaTeX 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 \LaTeX 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:

```

5679 \def\eql@provide@backup@equation{%
5680   \eql@amsmath@after{%
5681     \eql@provide@moveenv{amsequeation}{equation}%
5682     \eql@provide@moveenv{amsequeation*}{equation*}%
5683     \eql@tagging@register@env{amsequeation}%
5684     \eql@tagging@register@env{amsequeation*}%
5685     \eql@tagging@register@luamml{amsequeation}%
5686     \eql@tagging@register@luamml{amsequeation*}%
5687     \eql@markline@amsthm@move{amsequeation}{equation}%
5688     \eql@markline@amsthm@move{amsequeation*}{equation*}%
5689   }%
5690   \AddToHook{package/hyperref/after}{%
5691     \ifpackageloaded{amsmath}{}%
5692       \eql@provide@moveenv{hyperrefequation}{equation}%
5693       \eql@tagging@register@env{hyperrefequation}%
5694       \eql@tagging@register@luamml{hyperrefequation}%
5695       \eql@markline@amsthm@move{hyperequation}{equation}%
5696     }%
5697   }%
5698   \@ifpackageloaded{amsmath}{}%
5699   \@ifpackageloaded{hyperref}{%
5700     \let\latexequation\H@equation
5701     \let\endlatexequation\H@endequation
5702   }{\eql@provide@moveenv{latexequation}{equation}}%
5703   \eql@tagging@register@env{latexequation}%
5704   \eql@tagging@register@luamml{latexequation}%
5705   \eql@markline@amsthm@move{latexequation}{equation}%
5706 }%
5707 }

```

e@backup@displaymath **TODO:** describe

```

5708 \def\eql@provide@backup@displaymath{%
5709   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5710   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
5711 }

```

@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:

```

5712 \def\eql@provide@backup@subequations{%
5713   \eql@amsmath@after{%
5714     \eql@provide@moveenv{amssubequations}{subequations}%
5715   }%
5716   \AddToHook{package/hyperref/after}{%
5717     \AddToHook{cmd/amssubequations/before}{%
5718       {%
5719         \stepcounter{equation}%
5720         \protected@edef\theHparentequation{\theHequation}%
5721         \addtocounter{equation}{-1}%
5722       }%
5723     }%
5724     \AddToHook{cmd/amssubequations/after}{%
5725       {%
5726         \def\theHequation{\theHparentequation\alph{equation}}%

```

```

5726     \ignorespaces
5727   }%
5728 }%
5729 }

```

`\eql@provide@backup` Backup all amsmath environments:

```

5730 \def\eql@provide@backup{%
5731   \eql@provide@backup@eqref
5732   \eql@provide@backup@equation
5733   \eql@provide@backup@displaymath
5734   \eql@provide@backup@amsenv{gather}%
5735   \eql@provide@backup@amsenv{multline}%
5736   \eql@provide@backup@amsenv{align}%
5737   \eql@provide@backup@amsenv{flalign}%
5738   \eql@provide@backup@amsenv{alignat}%
5739   \eql@provide@backup@amsenv{xalignat}%
5740   \eql@provide@backup@amsenv{xxalignat}%
5741   \eql@provide@backup@amsenv{gather*}%
5742   \eql@provide@backup@amsenv{multline*}%
5743   \eql@provide@backup@amsenv{align*}%
5744   \eql@provide@backup@amsenv{flalign*}%
5745   \eql@provide@backup@amsenv{alignat*}%
5746   \eql@provide@backup@amsenv{xalignat*}%
5747   \eql@provide@backup@amsbox{gathered}%
5748   \eql@provide@backup@multlined
5749   \eql@provide@backup@amsbox{aligned}%
5750   \eql@provide@backup@amsbox{alignedat}%
5751   \eql@provide@backup@amsbox{cases}%
5752   \eql@provide@backup@amsbox{matrix}%
5753   \eql@provide@backup@amsbox{pmatrix}%
5754   \eql@provide@backup@amsbox{bmatrix}%
5755   \eql@provide@backup@amsbox{Bmatrix}%
5756   \eql@provide@backup@amsbox{vmatrix}%
5757   \eql@provide@backup@amsbox{Vmatrix}%
5758   \eql@provide@backup@subequations
5759 }

```

Replacement amsmath Environments. **TODO:** describe

```

5760 \def\eql@alignat@gobblecol#1{%
5761   \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.*)

```

5762 \newenvironment{eql@gathered}
5763   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5764 \newenvironment{eql@multlined}
5765   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5766 \newenvironment{eql@aligned}
5767   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5768 \newenvironment{eql@alignedat}
5769   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5770   {\endequationsbox}
5771 \newenvironment{eql@cases}
5772   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
5773 \newenvironment{eql@matrix}

```

```

5774 {\eqnaddopt{matrix=}\equationsbox}{\endequationsbox}
5775 \newenvironment{eql@pmatrix}
5776 {\eqnaddopt{matrix=r}\equationsbox}{\endequationsbox}
5777 \newenvironment{eql@bmatrix}
5778 {\eqnaddopt{matrix=s}\equationsbox}{\endequationsbox}
5779 \newenvironment{eql@Bmatrix}
5780 {\eqnaddopt{matrix=c}\equationsbox}{\endequationsbox}
5781 \newenvironment{eql@vmatrix}
5782 {\eqnaddopt{matrix=v}\equationsbox}{\endequationsbox}
5783 \newenvironment{eql@Vmatrix}
5784 {\eqnaddopt{matrix=d}\equationsbox}{\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.*)

```

5785 \newenvironment{eql@equation}
5786 {\eqnaddopt{equation}\equations}{\endequations}
5787 \newenvironment{eql@displaymath}
5788 {\eqnaddopt{equation,nonumber}\equations}{\endequations}
5789 \newenvironment{eql@gather}
5790 {\eqnaddopt{lines}\equations}{\endequations}
5791 \newenvironment{eql@multline}
5792 {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5793 {\endequations}
5794 \newenvironment{eql@align}
5795 {\eqnaddopt{columns}\equations}{\endequations}
5796 \newenvironment{eql@flalign}
5797 {\eqnaddopt{fulllength}\eql@align}{\endequations}
5798 \newenvironment{eql@alignat}
5799 {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
5800 \newenvironment{eql@xalignat}
5801 {\eql@alignat@gobblecol\eql@align}{\endequations}
5802 \newenvironment{eql@xxalignat}
5803 {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
5804 \newenvironment{eql@equation*}
5805 {\eqnaddopt{nonumber}\eql@equation}{\endequations}
5806 \newenvironment{eql@gather*}
5807 {\eqnaddopt{nonumber}\eql@gather}{\endequations}
5808 \newenvironment{eql@multline*}
5809 {\eqnaddopt{nonumber}\eql@multline}{\endequations}
5810 \newenvironment{eql@align*}
5811 {\eqnaddopt{nonumber}\eql@align}{\endequations}
5812 \newenvironment{eql@flalign*}
5813 {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
5814 \newenvironment{eql@alignat*}
5815 {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
5816 \newenvironment{eql@xalignat*}
5817 {\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):

```

5818 \def\eql@provide@onlyonce#1#2{%
5819   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5820   \ifx\eql@tmp\eql@tmpa
5821     \let\eql@tmp\@empty
5822   \fi
5823   \ifx\eql@tmp\@empty
5824     \let\eql@tmp\@undefined
5825     \ifx\@nodocument\relax
5826       \def\eql@tmp{#1}%
5827     \fi
5828     \ifcsname eql@provided@#1\endcsname
5829       \def\eql@tmp{#1}%
5830     \fi
5831     \eql@letcs{eql@provided@#1}\eql@true
5832   \else
5833     \def\eql@tmpa{*}%
5834     \ifx\eql@tmp\eql@tmpa
5835       \def\eql@tmp{#1}%
5836     \fi
5837   \fi
5838 }
```

\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:

```

5839 \def\eql@provide@eqref#1{%
5840   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
5841   \ifx\eql@tmp\eql@tmpa
5842     \let\eql@tmp\@empty
5843   \fi
5844   \ifx\eql@tmp\@empty
5845     \eql@provide@onlyonce{eqref}{}%
5846   \else
5847     \def\eql@tmpa{*}%
5848     \ifx\eql@tmp\eql@tmpa
5849       \def\eql@tmp{eqref}%
5850     \else
5851       \edef\eql@tmp{\expandafter\@gobble\string#1}%
5852     \fi
5853   \fi
5854   \ifdefined\eql@tmp
5855     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
5856   \else
5857     \eql@amsmath@after{%
5858       \eql@provide@movecmd{eqref}{eql@eqref}%
5859     }%
5860     \AddToHook{package/mathtools/after}{%
5861       \eql@provide@movecmd{eqref}{eql@eqref}%
5862     }%
5863     \eql@provide@movecmd{eqref}{eql@eqref}%
5864     \eql@amsmath@undefine\eqref
5865   \fi
5866 }
```

\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:

```

5867 \def\eq\provide@amsenv#1#2{%
5868   \eq\provide@onlyonce{#1}{#2}%
5869   \ifdefined\eq\tmp
5870     \eq\provide@moveenv{\eq\tmp}{eq\@#1}%
5871     \eq\tagging@register@luamml{\eq\tmp}%
5872     \eq\markline@amsthm@register{\eq\tmp}%
5873   \else
5874     \eq\amsmath@after{%
5875       \eq\provide@moveenv{#1}{eq\@#1}%
5876       \eq\markline@amsthm@register{#1}%
5877     }%
5878     \AddToHook{package/mathtools/after}{%
5879       \eq\provide@moveenv{#1}{eq\@#1}%
5880       \eq\markline@amsthm@register{#1}%
5881     }%
5882     \eq\provide@moveenv{#1}{eq\@#1}%
5883     \eq\markline@amsthm@register{#1}%
5884     \eq\amsmath@before{\eq\provide@undefineenv{#1}}%
5885   \fi
5886 }

```

\eq\provide@amsbox 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:

```

5887 \def\eq\provide@amsbox#1#2{%
5888   \eq\provide@onlyonce{#1}{#2}%
5889   \ifdefined\eq\tmp
5890     \eq\provide@moveenv{\eq\tmp}{eq\@#1}%
5891   \else
5892     \eq\amsmath@after{%
5893       \eq\provide@moveenv{#1}{eq\@#1}}%
5894     \AddToHook{package/mathtools/after}{%
5895       \eq\provide@moveenv{#1}{eq\@#1}}%
5896     \eq\provide@moveenv{#1}{eq\@#1}%
5897     \eq\amsmath@before{\eq\provide@undefineenv{#1}}%
5898   \fi
5899 }

```

\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:

```

5900 \def\eq\provide@multlined#1{%
5901   \eq\provide@onlyonce{multlined}{#1}%
5902   \ifdefined\eq\tmp
5903     \eq\provide@moveenv{\eq\tmp}{eq\@multlined}%
5904   \else
5905     \AddToHook{package/mathtools/after}{%
5906       \eq\provide@moveenv{multlined}{eq\@multlined}}%
5907     \eq\provide@moveenv{multlined}{eq\@multlined}%
5908     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5909       \eq\provide@undefineenv{multlined}}}%
5910   \fi
5911 }

```

\eq\provide@matrix Provide the cases and matrix environments. Copy into place, and copy again when

amsmath is loaded:

```

5912 \def\eql@provide@matrix#1#2#3{%
5913   \eql@provide@onlyonce{#1}{#3}%
5914   \ifdefined\eql@tmp
5915     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5916     \eql@tagging@register@luamml{\eql@tmp}%
5917   \else
5918     \eql@amsmath@after{%
5919       \eql@provide@moveenv{#1}{eql@#1}%
5920     }%
5921     \eql@provide@moveenv{#1}{eql@#1}%
5922     \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
5923   \fi
5924 }

```

eql@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:

```

5925 \def\eql@provide@equation#1{%
5926   \eql@provide@onlyonce{equation}{#1}%
5927   \ifdefined\eql@tmp
5928     \eql@provide@moveenv{\eql@tmp}{eql@equation}%
5929     \eql@tagging@register@luamml{\eql@tmp}%
5930     \eql@markline@amsthm@register{\eql@tmp}%
5931   \else
5932     \eql@amsmath@after{%
5933       \eql@provide@moveenv{equation}{eql@equation}%
5934       \eql@markline@amsthm@register{equation}%
5935     }%
5936     \AddToHook{package/hyperref/after}{%
5937       \ifpackageloaded{amsmath}{}%
5938       \eql@provide@moveenv{equation}{eql@equation}%
5939       \eql@markline@amsthm@register{equation}%
5940     }%
5941   }%
5942   \eql@provide@moveenv{equation}{eql@equation}%
5943   \eql@markline@amsthm@register{equation}%
5944   \ifdefined\eql@tagging@on
5945     \AddToHook{begindocument/end}{%
5946       \eql@provide@moveenv{equation}{eql@equation}%
5947       \eql@markline@amsthm@register{equation}%
5948     }%
5949   \fi
5950 \fi
5951 }

```

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:

```

5952 \def\eql@provide@equationstar#1{%
5953   \eql@provide@onlyonce{equation*}{#1}%
5954   \ifdefined\eql@tmp
5955     \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
5956     \eql@tagging@register@luamml{\eql@tmp}%
5957     \eql@markline@amsthm@register{\eql@tmp}%

```

```

5958 \else
5959   \eql@amsmath@after{%
5960     \eql@provide@moveenv{equation*}{eql@equation*}%
5961     \eql@markline@amsthm@register{equation*}%
5962   }%
5963   \eql@provide@moveenv{equation*}{eql@equation*}%
5964   \eql@markline@amsthm@register{equation*}%
5965   \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
5966   \ifdefined\eql@tagging@on
5967     \AddToHook{begindocument/end}{%
5968       \eql@provide@moveenv{equation*}{eql@equation*}%
5969       \eql@markline@amsthm@register{equation*}%
5970     }%
5971   \fi
5972 \fi
5973 }

```

`@provide@displaymath` **TODO:** describe

```

5974 \def\eql@provide@displaymath#1{%
5975   \eql@provide@onlyonce{displaymath}{#1}%
5976   \ifdefined\eql@tmp
5977     \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
5978     \eql@markline@amsthm@register{\eql@tmp}%
5979     \eql@tagging@register@luamml{\eql@tmp}%
5980   \else
5981     \eql@provide@moveenv{displaymath}{eql@displaymath}%
5982     \eql@markline@amsthm@register{displaymath}%
5983     \ifdefined\eql@tagging@on
5984       \AddToHook{begindocument/end}{%
5985         \eql@provide@moveenv{displaymath}{eql@displaymath}}%
5986     \fi
5987   \fi
5988 }

```

`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:

```

5989 \def\eql@provide@subequations#1{%
5990   \eql@provide@onlyonce{subequations}{#1}%
5991   \ifdefined\eql@tmp
5992     \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
5993   \else
5994     \eql@amsmath@after{%
5995       \eql@provide@moveenv{subequations}{eql@subequations}%
5996     }%
5997     \AddToHook{package/hyperref/after}{%
5998       \AddToHook{cmd/subequations/before}[hyperref]{}%
5999       \AddToHook{cmd/subequations/after}[hyperref]{}%
6000       \RemoveFromHook{cmd/subequations/before}[hyperref]%
6001       \RemoveFromHook{cmd/subequations/after}[hyperref]%
6002       \AddToHook{begindocument/end}{%
6003         \eql@provide@moveenv{subequations}{eql@subequations}}%
6004     }%

```



```

6005 \eql@provide@moveenv{subequations}{eql@subequations}%
6006 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6007 \fi
6008 }

```

`\eql@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:

```

6009 \def\eql@provide@sqr{%
6010 \let\[ \eql@sqr@open
6011 \let\] \eql@sqr@close
6012 \eql@amsmath@after{%
6013 \let\[ \eql@sqr@open
6014 \let\] \eql@sqr@close
6015 }%
6016 \ifdefined\eql@tagging@on
6017 \AddToHook{begindocument/end}{%
6018 \let\[ \eql@sqr@open
6019 \let\] \eql@sqr@close
6020 }%
6021 \fi
6022 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```

6023 \def\eql@provide@ang{%
6024 \let\< \eql@ang@open
6025 \let\> \eql@ang@close
6026 }

```

Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```

6027 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
6028 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
6029 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
6030 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
6031 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
6032 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
6033 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
6034 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
6035 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
6036 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
6037 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
6038 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
6039 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
6040 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}
6041 \eql@define@key{provide}{alignat*}[]{\eql@provide@amsenv{alignat*}{#1}}
6042 \eql@define@key{provide}{xalignat*}[]{\eql@provide@amsenv{xalignat*}{#1}}
6043 \eql@define@key{provide}{gathered}[]{\eql@provide@amsbox{gathered}{#1}}
6044 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
6045 \eql@define@key{provide}{aligned}[]{\eql@provide@amsbox{aligned}{#1}}
6046 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsbox{alignedat}{#1}}
6047 \eql@define@key{provide}{cases}[]{\eql@provide@matrix{cases}\eql@false{#1}}

```

```

6048 \eql@define@key{provide}{matrix}[]{\eql@provide@matrix{matrix}\eql@false{#1}}
6049 \eql@define@key{provide}{pmatrix}[]{\eql@provide@matrix{pmatrix}\eql@false{#1}}
6050 \eql@define@key{provide}{bmatrix}[]{\eql@provide@matrix{bmatrix}\eql@true{#1}}
6051 \eql@define@key{provide}{Bmatrix}[]{\eql@provide@matrix{Bmatrix}\eql@true{#1}}
6052 \eql@define@key{provide}{vmatrix}[]{\eql@provide@matrix{vmatrix}\eql@true{#1}}
6053 \eql@define@key{provide}{Vmatrix}[]{\eql@provide@matrix{Vmatrix}\eql@true{#1}}
6054 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
6055 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
6056 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
6057 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
6058 \eql@define@key{provide}{tagform}[]{%
6059   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}%
6060 \eql@define@key{provide}{maketag}[]{%
6061   \def\maketag@@@##1{\eql@tags@taglayout{##1}}%

```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```

6062 \newcommand{\eqnlinesprovide}[1]{%
6063   \eql@setkeys{provide}{#1}%
6064   \ignorespaces
6065 }

```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6066 \let\eql@error@packageoption\@gobble
```

Declare math layout options `leqno` and `fleqn` for common L^AT_EX classes:

```

6067 \DeclareOption{leqno}{\eqnlinesset{tagsleft}}
6068 \DeclareOption{fleqn}{\eqnlinesset{left}}

```

Pass undeclared options on to keyval processing:

```
6069 \DeclareOption*{\expandafter\eqnlinesset\expandafter{\CurrentOption}}
```

Set defaults for package:

```

6070 \eql@defaults@eqnlines
6071 \eql@mode@columns
6072 \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:

```

6073 \ifdefined\tagsleft@true\else
6074   \expandafter\newif\csname iftagsleft@\endcsname
6075 \fi
6076 \ifdefined\@fleqntrue\else
6077   \expandafter\newif\csname if@fleqn\endcsname
6078 \fi

```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```

6079 \eql@amsmath@after{%
6080   \ifnum\eql@provide@opt@env=\tw@
6081     \iftagsleft@
6082       \eqnlinesset{tags=left}%

```

```

6083     \else
6084     \eqnlineset{tags=right}%
6085     \fi
6086     \if@fleqn
6087     \eqnlineset{layout=left}%
6088     \else
6089     \eqnlineset{layout=center}%
6090     \fi
6091 \fi
6092 }

```

Process package options:

```
6093 \ProcessOptions
```

`\error@packageoption` Enable error message for exclusive package options:

```

6094 \def\eq@error@packageoption#1{%
6095   \eq@error{may only use '#1' as a package option}%
6096 }

```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```

6097 \ifnum\eq@provide@opt@env=\tw@
6098 \ifdefined\eq@provide@opt@matrix
6099   \let\eq@provide@opt@amsmathpatch\eq@false
6100 \fi\fi
6101 \ifdefined\eq@provide@opt@backup
6102   \let\eq@provide@opt@amsmathpatch\eq@true
6103 \fi
6104 \ifdefined\eq@provide@opt@amsmathpatch
6105   \eq@amsmath@fixends
6106   \eq@amsmath@fixmatrices
6107 \fi

```

Backup all `amsmath` environments that may be overwritten to `ams...`. This will happen before any replacements:

```
6108 \ifdefined\eq@provide@opt@backup\eq@provide@backup\fi
```

Provide native \LaTeX environment `equation` and symbolic shortcut `\[...\]` if desired:

```

6109 \ifnum\eq@provide@opt@env>\z@
6110   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6111 \fi

```

Provide `amsmath` equation environments if desired:

```

6112 \ifnum\eq@provide@opt@env=\tw@
6113   \eqnlinesprovide{%
6114     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6115     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6116     multlined,gathered,aligned,alignedat,%
6117     subequations}
6118 \fi

```

Provide symbolic shortcut `\<...\>` if desired:

```
6119 \ifdefined\eq@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
6120 \ifdefined\eq@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```

Provide `cases` and `matrix` environments if desired:

```
6121 \ifdefined\eq@provide@opt@matrix
6122   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6123 \fi
```