

# 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`

2026/01/30, v0.14.1

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

### Abstract

`eqnlines` is a L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  package providing a framework for typesetting single- and multi-line equations which extends the established equation environments of L<sup>A</sup>T<sub>E</sub>X 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

<b>1</b>	<b>Implementation</b>	<b>3</b>
<b>2</b>	<b>General Support</b>	<b>3</b>
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
<b>3</b>	<b>Parameters and Registers</b>	<b>12</b>
3.1	Parameters . . . . .	12
3.2	Registers . . . . .	15
3.3	Hooks . . . . .	18
<b>4</b>	<b>Features</b>	<b>18</b>
4.1	Punctuation . . . . .	18
4.2	Math Classes at Alignment . . . . .	21

4.3	Framed Cells . . . . .	22
4.4	Single-Line Composition . . . . .	22
4.5	Alternative Content Description . . . . .	23
<b>5</b>	<b>Equation Numbering</b>	<b>23</b>
5.1	Supporting Definitions . . . . .	24
5.2	Schemes . . . . .	25
5.3	Interface . . . . .	28
5.4	Integration . . . . .	33
5.5	Positioning . . . . .	36
5.6	Component Display . . . . .	39
5.7	Tag Composition . . . . .	42
<b>6</b>	<b>Subequation Numbering</b>	<b>45</b>
6.1	Definitions . . . . .	45
6.2	Environment . . . . .	46
6.3	Subequation Scheme . . . . .	48
<b>7</b>	<b>Display Equations Support</b>	<b>48</b>
7.1	Display Breaks . . . . .	48
7.2	Explicit Vertical Space . . . . .	50
7.3	Default Vertical Spacing . . . . .	51
7.4	Entry and Exit . . . . .	52
7.5	Stack . . . . .	57
<b>8</b>	<b>Multi-Line Support</b>	<b>59</b>
8.1	Measure Support . . . . .	59
8.2	Line Breaks . . . . .	60
8.3	Intertext . . . . .	62
8.4	Line Marks . . . . .	64
<b>9</b>	<b>Column Placement</b>	<b>65</b>
9.1	Supporting Definitions . . . . .	65
9.2	Shape Schemes . . . . .	66
9.3	Width Data . . . . .	69
9.4	Best Line Selection . . . . .	72
9.5	Tag Margin . . . . .	74
9.6	Single Column . . . . .	74
9.7	Multiple Columns . . . . .	75
<b>10</b>	<b>Single Column Arrangement</b>	<b>81</b>
10.1	Supporting Definitions . . . . .	81
10.2	Arrangement Methods . . . . .	82
10.3	Central Alignment . . . . .	83
10.4	Left Alignment . . . . .	84
10.5	Right Alignment . . . . .	86
<b>11</b>	<b>Equations Box Environment</b>	<b>87</b>
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	Options Processing . . . . .	94
11.7	Environment . . . . .	95

<b>12 Single-Line Equation</b>	<b>96</b>
12.1 Native Mode . . . . .	96
12.2 Print . . . . .	97
<b>13 Multi-Line with Single Column</b>	<b>98</b>
13.1 Measure . . . . .	98
13.2 Column Placement . . . . .	100
13.3 Print . . . . .	100
<b>14 Multi-Line with Multiple Columns</b>	<b>102</b>
14.1 Support . . . . .	102
14.2 Transpose . . . . .	102
14.3 Measure . . . . .	104
14.4 Columns Placement . . . . .	107
14.5 Print . . . . .	107
<b>15 Interface</b>	<b>111</b>
15.1 Scanning the Equation Body . . . . .	111
15.2 Options Processing . . . . .	114
15.3 Single-Line Main . . . . .	116
15.4 Multi-Line Main . . . . .	117
15.5 Equations Environment . . . . .	119
<b>16 Options</b>	<b>122</b>
16.1 Selection Tools . . . . .	122
16.2 Options Declarations . . . . .	122
16.3 Parameter Presets . . . . .	135
16.4 Component Selection . . . . .	137
16.5 Global and Package Options . . . . .	148

# 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<sup>A</sup>T<sub>E</sub>X 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  $\varepsilon$ -T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X defines `\dollar@dollar@begin` and `\dollar@dollar@end` to represent (and adjust) the beginning and end of bare T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X `\@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<sup>A</sup>T<sub>E</sub>X’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<sup>A</sup>T<sub>E</sub>X 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 groups<sup>1</sup> 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}}

```

<sup>1</sup>See <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<sup>A</sup>T<sub>E</sub>X’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\eql@warn@tag@unused{\eql@warn@tags@unused{\string\tag}
144 {tag declaration}}
145 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string\tag's}
146 {tag declaration will be lost}{}
147 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}
148 {tag label declaration}}
149 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}
150 {tag label declaration}{}

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

```

## 2.6 amsmath Integration

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

```

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

```

**TODO:** temporary fix for development stages

```

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

```

## 2.7 PDF Tagging Support

`\eql@tagging@...` Proper PDF tagging<sup>2</sup> support requires a L<sup>A</sup>T<sub>E</sub>X version at least of 2025. For the time being, we define an abstraction layer so that the package will collaborate with L<sup>A</sup>T<sub>E</sub>X versions around 2020: **TODO:** adjust to further developments

```

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

```

---

<sup>2</sup>see <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@tablesaverinner{%
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@tablesaverinner\@empty
220   \let\eql@tagging@tableaddinner\@empty

```

```

221 \let\eql@tagging@tablesavelines\@empty
222 \let\eql@tagging@tablesavealign\@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}
```

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

```
341 \let\eq@spread@reset\eq@false
342 \def\eq@spread@val{\jot}
343 \newdimen\eq@spread@
```

`\eq@tagfuzz@` (*dimen*) The value `\eq@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<sub>E</sub>X, nevertheless: **TODO**: probably do not need this due to fixed point arithmetic.

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

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

```
346 \def\eq@display@height{\@undefined}
347 \def\eq@display@depth{\@undefined}
```

`\eq@skip@mode@short` The setting `\eq@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\eq@skip@mode@short{2}

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

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

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

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

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

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

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

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

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

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

`\eq@skip@med@above` The surrounding glue for an equation with reduced spacing is given by `\eq@skip@med@above` and `\eq@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 quad<sup>3</sup> 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@

```

---

<sup>3</sup>another 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)
\eqldprevdepth@ (dimen)
\eqldprevgraf@ (counter)
392 \newbox\eqldcellbox@
393 \newbox\eqldtagbox@
394 \newdimen\eqldcellwidth@
395 \newdimen\eqldprevwidth@
396 \newdimen\eqldtagwidth@
397 \newdimen\eqldprevdepth@
398 \newcount\eqldprevgraf@
```

```

\eqldtotalwidth@ (dimen)
\eqldtagwidth@max@ (dimen)
\eqldtotalheight@ (dimen)
399 \newdimen\eqldtotalwidth@
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)
\eqldline@pos@ (dimen)
\eqldwidthsep@ (counter)
\eqldavailsep@ (counter)
\eqldline@possep@ (counter)
\eqldline@offset@ (dimen)
\eqldprevdepth@ (dimen)
\eqldinterline@ (dimen)
406 \newdimen\eqldline@width@
407 \newdimen\eqldline@avail@
408 \newdimen\eqldline@pos@
409 \newcount\eqldline@availsep@
410 \newcount\eqldline@widthsep@
411 \newcount\eqldline@possep@
412 \newdimen\eqldline@offset@
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 the 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<sup>A</sup>T<sub>E</sub>X 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@line`  
`\eql@punct@block` **TODO:** update

```
\eql@punct@next
\eql@punct@top
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@top\@undefined
```

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

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

```
\eql@punct@top@set TODO: describe
\eql@punct@top@stop
\eql@punct@top@reset
462 \def\eql@punct@top@set{%
463   \let\eql@punct@top\eql@punct@block}
464 \def\eql@punct@top@stop{%
465   \let\eql@punct@top\relax}
466 \def\eql@punct@top@reset{%
467   \let\eql@punct@top\@undefined}
```

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

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

```

472 \ifx#1\eql@punct@relax
473   \let#1\@undefined
474 \fi
475 \ifx#1\eql@punct@tilde
476   \let#1\@empty
477 \fi
478 }
479 \def\eql@punct@clear{%
480   \let\eql@punct@col\@empty
481   \let\eql@punct@line\@empty
482   \let\eql@punct@block\@empty
483 }

```

Set the puncton for 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:** describe

```

484 \def\eql@punct@addopt{%
485   \eql@ifstar@tight\eql@punct@addopt@relax\eql@punct@addopt@set}
486 \def\eql@punct@addopt@set#1{\eqnaddopt{punct={#1}}\ignorespaces}
487 \def\eql@punct@addopt@relax{\eqnaddopt{punct*}\ignorespaces}

```

**TODO:** describe

```

488 \def\eql@punct@setnext{%
489   \eql@ifstar@tight\eql@punct@setnext@relax\eql@punct@setnext@set}
490 \def\eql@punct@setnext@set#1{%
491   \eql@punct@set\eql@punct@next{#1}%
492   \ifdefined\eql@punct@next\else\let\eql@punct@next\relax\fi
493   \ignorespaces}
494 \def\eql@punct@setnext@relax{\let\eql@punct@next\relax\ignorespaces}

```

`\eqnpunct` **TODO:** describe

```

495 \let\eqnpunct\eql@punct@addopt

```

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

```

496 \def\eql@punct@fill@next#1{%
497   \ifdefined\eql@punct@next
498     \ifx\eql@punct@next\relax
499       \let\eql@punct@next\@undefined
500     \fi
501   \else
502     \ifx\eql@punct@top\relax\else
503       \let\eql@punct@next#1%
504     \fi
505   \fi
506 }

```

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

```

507 \def\eql@punct@output@next{%
508   \ifx\eql@punct@next\@empty\else
509     \eql@punct@sep
510     \eql@punct@next
511   \fi
512   \let\eql@punct@next\@undefined
513 }

```

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

```
514 \def\eql@punct@print@next{%  
515   \ifdefined\eql@punct@next  
516     \eql@punct@output@next  
517   \fi  
518 }
```

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

```
519 \def\eql@punct@apply@next{%  
520   \ifdefined\eql@punct@next  
521     \eql@punct@output@next  
522     \eql@punct@top@stop  
523   \fi  
524 }
```

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

```
525 \def\eql@punct@print@col{%  
526   \eql@punct@fill@next\eql@punct@col  
527   \eql@punct@print@next  
528 }
```

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

```
529 \def\eql@punct@apply@col{%  
530   \eql@punct@fill@next\eql@punct@col  
531   \eql@punct@apply@next  
532 }
```

Output the punctuation for the present line unless disabled:

`eql@punct@print@line`

```
533 \def\eql@punct@print@line{%  
534   \eql@punct@fill@next\eql@punct@line  
535   \eql@punct@print@next  
536 }
```

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

`eql@punct@apply@line`

```
537 \def\eql@punct@apply@line{%  
538   \eql@punct@fill@next\eql@punct@line  
539   \eql@punct@apply@next  
540 }
```

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

```
541 \def\eql@punct@apply@block{%  
542   \eql@punct@fill@next\eql@punct@block  
543   \eql@punct@apply@next  
544 }
```

`\eqlopunct@apply@top` Output the top punctuation unless disabled. Stop further punctuation globally:

```
545 \def\eqlopunct@apply@top{%
546   \eqlopunct@fill@next\eqlopunct@top
547   \eqlopunct@print@next
548   \global\eqlopunct@top@stop
549 }
```

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

```
550 \newcommand{\eqnpunctapply}{\ifmmode\else\unskip\fi\eqlopunct@apply@top}
```

## 4.2 Math Classes at Alignment

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

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

```
551 \def\eqlopclass@innerright@sel@{%
552   \ifdim\eqlopwidth@=\z@
553     \eqlopclass@innerlead
554   \else
555     \eqlopclass@innerright
556   \fi
557 }
```

`\class@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:

`\class@innerright@set`

`\class@innerlead@set`

```
558 \def\eqlopclass@innerleft@set#1{%
559   \def\eqlopclass@innerleft{#1}%
560 }
561 \def\eqlopclass@innerright@set#1{%
562   \def\eqlopclass@innerright{#1}%
563   \let\eqlopclass@innerright@sel\eqlopclass@innerright
564 }
565 \def\eqlopclass@innerlead@set#1{%
566   \def\eqlopclass@innerlead{#1}%
567   \let\eqlopclass@innerright@sel\eqlopclass@innerright@sel@
568 }
```

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

`\eqlopclass@eqamp`

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

## 4.3 Framed Cells

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

```
579 \let\eql@frame@cmd\@undefined
580 \newdimen\eql@frame@margin@
581 \def\eql@frame@set[#1]{%
582   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
583 \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbx}
584 \def\eql@frame@measure{%
585   \setbox\z@\hbox{\eql@frame@cmd}}%
586   \eql@frame@margin@.5\wd\z@
587 }
588 \def\eql@frame@print{%
589   \setbox\eql@cellbox@\hbox{%
590     \eql@frame@cmd{\unhbox\eql@cellbox@}%
591   }%
592 }
593 \def\eql@frame@adjust{%
594   \setbox\eql@cellbox@\hbox{%
595     \eql@frame@cmd{%
596       \unhbox\eql@cellbox@
597       \unkern
598       \unskip
599     }%
600     \hfil
601     \kern\z@
602   }%
603 }
```

## 4.4 Single-Line Composition

**TODO:** describe

```
\eql@break@line
\eql@break@col
604 \def\eql@break@line{%
605   \let\eql@break@sep\eql@break@line@sep
606   \let\eql@break@shortsep\eql@break@line@shortsep
607   \let\eql@break@print\eql@punct@print@line
608   \let\eql@punct@termcr\eql@false
609   \eql@ampprotect\eql@break@testall\eql@break@process}
610 \def\eql@break@col{%
611   \let\eql@break@sep\eql@break@col@sep
612   \let\eql@break@shortsep\eql@break@col@shortsep
613   \let\eql@break@print\eql@punct@print@col
614   \let\eql@punct@term@cr\eql@false
615   \eql@ampprotect\eql@break@testall\eql@break@process}
```

```
\eql@break@testall TODO: describe
\eql@break@parse
616 \def\eql@break@testall{\eql@parseopt@aux\eql@break@parse}
617 \def\eql@break@parse{%
618   \ifx\eql@parseopt@token[%
619     \let\eql@parseopt@next\eql@break@parse@val
620     \fi
621   \ifx\eql@parseopt@token*%
622     \let\eql@parseopt@next\eql@break@parse@star
```

```

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@punctpass
632 \fi
633 \ifx\eql@parseopt@token' %
634   \let\eql@parseopt@next\eql@parseopt@punctnext
635 \fi
636 \ifx\eql@parseopt@token!%
637   \let\eql@parseopt@next\eql@parseopt@puncttermcr
638 \fi
639 }
640 \def\eql@break@parse@val[#1]{\def\eql@break@sep{#1}\eql@parseopt@peek}
641 \def\eql@break@parse@star#1{%
642   \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}

```

`\eql@break@process`

```

643 \def\eql@break@process{%
644   \ifdefined\eql@punct@term@cr\eql@punct@apply@top\fi
645   \eql@break@print
646   \hspace{\glueexpr\eql@break@sep\relax}}

```

`\eql@break@join`

```

647 \def\eql@break@join{\eql@srbgroup
648   \eql@ifstar@tight
649   {\eql@break@join@opt[\eql@break@col@shortsep]}%
650   {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
651 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
652   \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

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

```

653 \newcommand{\eqnsep}{\eql@break@col}
654 \newcommand{\eqnbreak}{\eql@break@line}
655 \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

```

656 \DeclareRobustCommand{\eqnalt}[2][{}]{

```

## 5 Equation Numbering

**TODO:** describe

## 5.1 Supporting Definitions

### Parameters.

```
657 \let\eql@tags@autolabel\eql@false
658 \let\eql@tags@autotag\eql@true
659 \let\eql@tags@warn\eql@true

660 \def\eql@tags@name@generic{[equation]}

661 \let\eql@tagpos@doconvert\eql@false

662 \def\eql@tagpos@snap{4pt}
```

### Registers.

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

664 \let\eql@numbering@active\eql@true
665 \let\eql@numbering@multi\eql@true

666 \let\eql@tags@container\@undefined
667 \def\eql@tags@container@clear{%
668   \let\eql@tags@label\@undefined
669   \let\eql@tags@name\@undefined
670   \let\eql@tags@tag\@undefined
671   \let\eql@tags@ref\@undefined
672   \let\eql@tags@anchor\@empty
673   \eql@tagpos@shift@z@
674   \eql@tagpos@smashup@z@
675   \eql@tagpos@smashdown@z@
676   \let\eql@tagpos@reserve\eql@true
677 }

678 \let\eql@tags@label\@undefined
679 \let\eql@tags@name\@undefined
680 \let\eql@tags@tag\@undefined
681 \let\eql@tags@ref\@undefined
682 \let\eql@tags@frame@cmd\@firstofone
```

ags@glabel@ (*counter*)

```
683 \newcount\eql@tags@glabel@
684 \eql@tags@glabel@z@
685 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
686 \def\eql@tags@glabel@step{global\advance\eql@tags@glabel@\@ne}

687 \let\eql@tagpos@continuous\eql@false

688 \newcount\eql@tagpos@row@
689 \newcount\eql@tagpos@prevrow@
690 \newdimen\eql@tagpos@shift@
691 \newdimen\eql@tagpos@smashup@
692 \newdimen\eql@tagpos@smashdown@
693 \newdimen\eql@tagpos@current@
694 \newdimen\eql@tagpos@plain@
695 \newdimen\eql@tagpos@raised@
696 \newdimen\eql@tagpos@target@
697 \newdimen\eql@tagpos@headroom@
698 \newdimen\eql@tagpos@footroom@
```



## 5.2 Schemes

**TODO:** describe

Table.

```
699 \def\eql@numbering@tab@sub{sub}
700 \def\eql@numbering@tab@all{all}
701 \def\eql@numbering@tab@first{first}
702 \def\eql@numbering@tab@last{last}
703 \def\eql@numbering@tab@in{in}
704 \def\eql@numbering@tab@out{out}
705 \def\eql@numbering@tab@middle{middle}
706 \def\eql@numbering@tab@best{best}
707 \def\eql@numbering@tab@here{here}
708 \def\eql@numbering@tab@top{top}
709 \def\eql@numbering@tab@bottom{bottom}
710 \def\eql@numbering@tab@center{center}
711 \def\eql@numbering@tab@centerone{centerone}
712 \def\eql@numbering@tab@median{median}
713 \def\eql@numbering@tab@baseline{baseline}

714 \let\eql@numbering@mode\eql@numbering@tab@all
715 \let\eql@numbering@mode@multi\eql@numbering@tab@all
716 \let\eql@numbering@mode@single\eql@numbering@tab@out
```

**TODO:** describe

```
717 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
718 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
719 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
720 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
721 \let\eql@numbering@tab@outside\eql@numbering@tab@out
722 \let\eql@numbering@tab@inside\eql@numbering@tab@in
723 \let\eql@numbering@tab@within\eql@numbering@tab@in
724 \let\eql@numbering@tab@opt\eql@numbering@tab@best
725 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
726 \let\eql@numbering@tab@pick\eql@numbering@tab@here
727 \let\eql@numbering@tab@med\eql@numbering@tab@median
728 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
729 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone
```

**TODO:** describe

```
730 \let\eql@numbering@tab@a\eql@numbering@tab@all
731 \let\eql@numbering@tab@s\eql@numbering@tab@sub
732 \let\eql@numbering@tab@f\eql@numbering@tab@first
733 \let\eql@numbering@tab@l\eql@numbering@tab@last
734 \let\eql@numbering@tab@o\eql@numbering@tab@out
735 \let\eql@numbering@tab@i\eql@numbering@tab@in
736 \let\eql@numbering@tab@h\eql@numbering@tab@here
737 \let\eql@numbering@tab@t\eql@numbering@tab@top
738 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
739 \let\eql@numbering@tab@c\eql@numbering@tab@center
740 \let\eql@numbering@tab@m\eql@numbering@tab@median
741 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
742 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
743 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
744 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone
```

**Implementations.** **TODO:** describe

```
745 \def\eql@numbering@init@all{\let\eql@numbering@multi\eql@true}
```

**TODO:** describe

```
746 \def\eql@numbering@init@sub{%
747   \let\eql@numbering@multi\eql@true
748   \ifdefined\eql@subequations@active
749     \let\eql@numbering@mode\eql@numbering@tab@all
750   \else
751     \let\eql@numbering@subeq@use\eql@true
752   \fi
753 }

754 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}
755 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}
756 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}
```

**TODO:** describe

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

**TODO:** describe

```
764 \def\eql@numbering@init@out{%
765   \ifdefined\eql@tagsleft
766     \eql@numbering@init@first
767   \else
768     \eql@numbering@init@last
769   \fi
770 }
```

**TODO:** describe

```
771 \def\eql@tagpos@eval@middle{%
772   \ifnum\eql@tagpos@row@=\z@
773     \eql@tagpos@row@\numexpr(\eql@totalrows@
774       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
775   \fi
776 }
```

**TODO:** describe

```
777 \def\eql@tagpos@eval@best{%
778   \ifnum\eql@tagpos@row@=\z@
779     \let\eql@numbering@best@use\eql@true
780     \eql@numbering@init@out
781   \fi
782 }
```

**TODO:** describe

```
783 \def\eql@numbering@init@continuous{\let\eql@tagpos@continuous\eql@true}
```

**TODO:** describe

```

784 \let\eql@numbering@init@top\eql@numbering@init@continuous
785 \def\eql@tagpos@eval@top{%
786   \eql@tagpos@current@\z@
787 }

```

**TODO:** describe

```

788 \let\eql@numbering@init@bottom\eql@numbering@init@continuous
789 \def\eql@tagpos@eval@bottom{%
790   \eql@tagpos@current@\dimexpr\eql@totalheight@
791     -\eql@tagheight@block@-\eql@tagdepth@block@\relax
792 }

```

**TODO:** describe

```

793 \let\eql@numbering@init@center\eql@numbering@init@continuous
794 \def\eql@tagpos@eval@center{%
795   \ifnum\eql@totalrows@=\@ne
796     \eql@tagpos@row@\@ne
797   \fi
798   \eql@tagpos@current@\dimexpr(\eql@totalheight@
799     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
800 }

```

**TODO:** describe

```

801 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
802 \def\eql@tagpos@eval@centerone{%
803   \eql@tagpos@current@\dimexpr(\eql@totalheight@
804     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
805 }

```

**TODO:** describe

```

806 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
807 \def\eql@tagpos@eval@baseline{%
808   \eql@tagpos@current@\dimexpr(\eql@totalheight@
809     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
810 }

```

**TODO:** describe

```

811 \let\eql@numbering@init@median\eql@numbering@init@continuous
812 \def\eql@tagpos@eval@median{%
813   \ifnum\eql@tagpos@row@=\z@
814     \ifodd\eql@totalrows@
815       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
816     \else
817       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
818       \eql@dimensions@get\eql@tagpos@row@
819       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
820         +(\eql@line@interline@-\eql@tagheight@block@
821           +\eql@tagdepth@block@)/\tw@\relax
822     \fi
823   \ifnum\eql@totalrows@=\@ne
824     \eql@tagpos@row@\@ne
825   \else
826     \eql@tagpos@adjust@eval@convert
827     \eql@tagpos@row@\z@
828   \fi
829 \fi
830 }

```

**Selection.**

```

831 \def\eql@numbering@set#1{%
832   \ifcsname eql@numbering@tab@#1\endcsname
833     \expandafter\let\expandafter\eql@numbering@mode
834     \csname eql@numbering@tab@#1\endcsname
835   \ifx\eql@numbering@mode\eql@numbering@tab@all
836     \let\eql@numbering@mode@multi\eql@numbering@mode
837   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
838     \let\eql@numbering@mode@multi\eql@numbering@mode
839   \else
840     \let\eql@numbering@mode@single\eql@numbering@mode
841   \fi\fi
842 \else
843   \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
844   \let\eql@numbering@mode\eql@numbering@tab@all
845 \fi
846 }

```

**TODO:** describe

```

847 \def\eql@numbering@init{%
848   \let\eql@numbering@multi\eql@false
849   \let\eql@tagpos@continuous\eql@false
850   \let\eql@numbering@subeq@use\eql@false
851   \let\eql@numbering@best@use\eql@false
852   \eql@tagpos@row@z@
853   \csname eql@numbering@init@\eql@numbering@mode\endcsname
854   \ifdefined\eql@numbering@active
855     \let\eql@numbering@eqnswinit\@eqnswtrue
856   \else
857     \let\eql@numbering@eqnswinit\@eqnswfalse
858   \fi
859   \let\eql@numbering@active\eql@false
860 }

```

## 5.3 Interface

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

```

861 \eql@amsmath@after{
862   \let\eql@print@eqnum@default\print@eqnum
863   \let\eql@incr@eqnum@default\incr@eqnum
864 }

```

**TODO:** describe

```

865 \protected\def\donumber{%
866   \if@eqnsw\else
867     \global\@eqnswtrue
868     \ifx\print@eqn\@empty
869       \global\let\print@eqn\eql@print@eqnum@default
870     \fi
871     \ifx\incr@eqn\@empty
872       \global\let\incr@eqn\eql@incr@eqnum@default
873     \fi
874   \fi
875 }

```

**TODO:** reconsider operation

`\numberhere`

```
876 \protected\def\eq1@numberhere{%
877   \ifdefined\eq1@numbering@multi
878     \global\@eqnswtrue
879   \else
880     \global\eq1@tagpos@row@\eq1@row@
881   \fi
882 }
```

**TODO:** describe

`\numbernext`

```
883 \protected\def\eq1@numbernext{%
884   \ifdefined\eq1@numbering@multi
885     \global\@eqnswfalse
886   \else
887     \ifdefined\eq1@tagpos@continuous\else
888       \ifnum\eq1@tagpos@row@=\eq1@row@
889         \global\advance\eq1@tagpos@row@\@ne
890       \fi
891     \fi
892   \fi
893 }
```

**Activation Trigger.**

```
894 \def\eq1@tags@autoenable{%
895   \global\@eqnswtrue
896   \ifnum\eq1@tagpos@row@=\m@ne
897     \numberhere
898   \fi
899 }
```

**Labels.** **TODO:** describe

`\eq1@label@org`

```
900 \let\eq1@label@org\label
```

**TODO:** describe

```
901 \def\eq1@label@gobble{\eq1@ampprotect\eq1@testopt@tight\eq1@gobbleoptone{}}
```

**TODO:** describe

```
902 \protected\def\eq1@label{%
903   \eq1@ampprotect\eq1@testopt@tight\eq1@tags@add@labelname\eq1@testopt@default
904 }
```

**TODO:** describe

```
905 \def\eq1@tags@add@labelname[#1]#2{%
906   \def\eq1@tmp{#1}%
907   \ifx\eq1@tmp\eq1@testopt@default\else
908     \eq1@tags@add@name{#1}%
909   \fi
910   \eq1@tags@add@label{#2}%
911 }
```

**TODO:** describe

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

**TODO:** describe

```
920 \def\eql@tags@set@name#1{%
921   \ifdefined\eql@tags@warn
922     \ifdefined\eql@tags@name
923       \eql@warn@name@multiple
924     \fi
925   \fi
926   \def\eql@tags@name{#1}%
927 }
```

**TODO:** describe

```
928 \def\eql@tags@add@label#1{%
929   \ifdefined\eql@tags@autolabel
930     \eql@tags@autoenable
931   \fi
932   \global\eql@appendexpand\eql@tags@container{%
933     \noexpand\eql@tags@set@label{#1}}%
934 }
```

**TODO:** describe

```
935 \def\eql@tags@add@name#1{%
936   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
937   \global\eql@appendmacro\eql@tags@container\eql@tmp
938 }
```

**TODO:** describe

```
939 \def\eql@tags@addblock@label#1{%
940   \eql@appendexpand\eql@tags@container@block{%
941     \noexpand\eql@tags@set@label{#1}}%
942 }
```

**TODO:** describe

```
943 \def\eql@tags@addblock@name#1{%
944   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
945   \eql@appendmacro\eql@tags@container@block\eql@tmp
946 }
```

**Tags.** **TODO:** describe

`\eql@tag@default`

```
947 \protected\def\eql@tag@default{%
948   \eql@warn@here\tag
949   \eql@tag@gobble
950 }
951 \let\tag\eql@tag@default
```

\eql@tag@gobble

```
952 \def\eql@tag@gobble{%
953   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

**TODO:** describe

```
954 \protected\def\eql@tag{%
955   \eql@ampprotecttwo\eql@teststaropt@tight
956   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
957   \eql@testopt@default
958 }
```

\eql@tags@add@tagref

```
959 \def\eql@tags@add@tagref[#1]#2{%
960   \def\eql@tmp{#1}%
961   \ifx\eql@tmp\eql@testopt@default\else
962     \eql@tags@add@ref{#1}%
963   \fi
964   \eql@tags@add@tag{#2}%
965 }
```

**TODO:** describe

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

**TODO:** describe

```
974 \def\eql@tags@set@ref#1{%
975   \ifdefined\eql@tags@warn
976     \ifdefined\eql@tags@ref
977       \eql@warn@ref@multiple
978     \fi
979   \fi
980   \def\eql@tags@ref{#1}%
981 }
```

**TODO:** describe

```
982 \def\eql@tags@add@tag#1{%
983   \ifdefined\eql@tags@autotag
984     \eql@tags@autoenable
985   \fi
986   \protected@edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
987   \global\eql@appendmacro\eql@tags@container\eql@tmp
988 }
```

**TODO:** describe

```
989 \def\eql@tags@add@ref#1{%
990   \protected@edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
991   \global\eql@appendmacro\eql@tags@container\eql@tmp
992 }
```

tags@add@tagform@off

```
993 \def\eq\@tags@add@tagform@off{%
994   \global\eq\@append\eq\@tags@container{\let\eq\@tags@tagform\@firstofone}%
995 }
```

**TODO:** describe

```
996 \def\eq\@tags@addblock@tag#1{%
997   \protected@edef\eq\@tmp{\noexpand\eq\@tags@set@tag{#1}}%
998   \eq\@appendmacro\eq\@tags@container@block\eq\@tmp
999 }
```

**TODO:** describe

```
1000 \def\eq\@tags@addblock@ref#1{%
1001   \protected@edef\eq\@tmp{\noexpand\eq\@tags@set@ref{#1}}%
1002   \eq\@appendmacro\eq\@tags@container@block\eq\@tmp
1003 }
```

**TODO:** describe

```
1004 \def\eq\@tags@addblock@tagform@off{%
1005   \eq\@append\eq\@tags@container@block{\let\eq\@tags@tagform\@firstofone}%
1006 }
```

## Raise Tags.

\raisetag

```
1007 \def\eq\@raisetag@default{%
1008   \eq\@warn@here\raisetag
1009   \eq\@raisetag@gobble
1010 }

1011 \def\eq\@raisetag@gobble{%
1012   \eq\@ampprotecttwo\eq\@ifstar@tight\@gobble\@gobble
1013 }
```

**TODO:** describe

```
1014 \eq\@amsmath@let\raisetag\eq\@raisetag@default

1015 \def\eq\@raisetag{%
1016   \eq\@ampprotecttwo\eq\@ifstar@tight\eq\@tags@add@raiseshift\eq\@raisetag@test
1017 }

1018 \def\eq\@raisetag@test#1{%
1019   \def\eq\@tmpa{#1}\def\eq\@tmpb{!}%
1020   \ifx\eq\@tmpa\eq\@tmpb
1021     \eq\@tags@add@forceraise
1022   \else
1023     \eq\@tags@add@raisesmash{#1}%
1024   \fi
1025 }

1026 \def\eq\@tags@add@raiseshift#1{%
1027   \global\eq\@appendexpand\eq\@tags@container{%
1028     \advance\eq\@tagpos@shift@the\glueexpr#1\relax\relax}%
1029 }
```



```

1030 \def\eql@tags@add@raisesmash#1{%
1031   \dimen@{\glueexpr#1\relax
1032   \ifdim\dimen@<\z@
1033     \global\eql@appendexpand\eql@tags@container{%
1034       \advance\eql@tagpos@smashdown@-\the\dimen@\relax}%
1035   \else
1036     \global\eql@appendexpand\eql@tags@container{%
1037       \advance\eql@tagpos@smashup@\the\dimen@\relax}%
1038   \fi
1039 }

1040 \def\eql@tags@add@forceraise{%
1041   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
1042 }

```

## 5.4 Integration

**TODO:** describe

**Support.** **TODO:** describe

```

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

```

**TODO:** not necessary anymore

```

1050 \def\eql@numbering@settools@gobble{%
1051   \let\label\eql@label@gobble
1052   \let\tag\eql@tag@gobble
1053   \let\raisetag\eql@raisetag@gobble
1054   \let\numberhere\relax
1055   \let\numbernext\relax
1056 }

```

```

1057 \def\eql@numbering@autoblock{%
1058   \begingroup
1059     \let\eql@tags@warn\eql@false
1060     \eql@tags@container@block
1061     \ifdefined\eql@tags@autolabel
1062       \ifdefined\eql@tags@label
1063         \global\@eqnswtrue
1064       \fi
1065     \fi
1066     \ifdefined\eql@tags@autotag
1067       \ifdefined\eql@tags@tag
1068         \global\@eqnswtrue
1069       \fi
1070     \fi
1071   \endgroup
1072 }

```

```

1073 \def\eql@numbering@warnunused{%
1074   \ifdefined\eql@tags@label

```

```

1075     \eql@warn@label@unused
1076 \fi
1077 \ifdefined\eql@tags@name
1078     \eql@warn@name@unused
1079 \fi
1080 \ifdefined\eql@tags@tag
1081     \eql@warn@tag@unused
1082 \fi
1083 \ifdefined\eql@tags@erf
1084     \eql@warn@ref@unused
1085 \fi
1086 }

```

**Single Line.** **TODO:** describe

```

1087 \def\eql@numbering@single@init{%
1088     \let\eql@numbering@multi\eql@false
1089     \eql@numbering@settools
1090     \eql@numbering@eqnswinit
1091     \eql@numbering@autoblock
1092     \global\let\eql@tags@container\eql@tags@container@block
1093     \let\eql@tags@warn\eql@true
1094 }

1095 \def\eql@numbering@single@eval{%
1096     \ifnum\eql@tagpos@row@=\m@ne
1097         \@eqnswfalse
1098     \fi
1099 }

```

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

```

1100 \def\eql@numbering@measure@init{%
1101     \eql@numbering@settools
1102     \ifdefined\eql@numbering@multi\else
1103         \eql@numbering@eqnswinit
1104         \eql@numbering@autoblock
1105     \fi
1106     \global\let\eql@tags@container\eql@tags@container@block
1107     \let\eql@tags@warn\eql@true
1108 }

```

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

```

1109 \def\eql@numbering@measure@line@begin{%
1110     \ifdefined\eql@numbering@multi
1111         \global\eql@numbering@eqnswinit
1112     \fi
1113 }

```

**TODO:** describe

```

1114 \def\eql@numbering@measure@blocktag{%
1115     \ifdefined\eql@numbering@multi
1116         \@eqnswfalse
1117     \else
1118         \ifnum\eql@tagpos@row@=\m@ne
1119             \@eqnswfalse
1120         \fi

```

```

1121 \ifnum\eq\@totalrows@=\z@
1122 \eq\@eqnswfalse
1123 \fi
1124 \fi
1125 }

```

**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!

```

1126 \def\eq\@numbering@print@init{%
1127 \let\eq\@tags@warn\eq\@false
1128 \ifdefined\eq\@numbering@multi
1129 \eq\@numbering@settools
1130 \global\let\eq\@tags@container\eq\@tags@container@block
1131 \else
1132 \let\eq\@tags@container@block\eq\@tags@container
1133 \eq\@numbering@settools@gobble
1134 \fi
1135 }

```

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

```

1136 \def\eq\@numbering@print@block@begin{%
1137 \ifdefined\eq\@numbering@multi\else
1138 \ifnum\eq\@tagpos@row@>\z@
1139 \eq\@tags@makeblockanchor
1140 \global\eq\@appendexpand\eq\@tags@container@block{%
1141 \def\noexpand\eq\@tags@anchor{%
1142 \unexpanded\expandafter{\eq\@tags@anchor}}}%
1143 \fi
1144 \fi
1145 \ifdefined\eq\@numbering@subeq@use
1146 \eq\@tags@printsubeqlabel
1147 \fi
1148 }

```

**TODO:** describe

```

1149 \def\eq\@numbering@print@line@begin{%
1150 \ifdefined\eq\@numbering@multi
1151 \global\eq\@numbering@eqnswinit
1152 \fi
1153 }

```

**TODO:** describe

```

1154 \def\eq\@numbering@print@line@eval{%
1155 \ifdefined\eq\@numbering@multi
1156 \if\eq\@eqnsw
1157 \eq\@tags@container
1158 \fi
1159 \else
1160 \ifnum\eq\@tagpos@row@=\eq\@row@
1161 \eq\@eqnswtrue
1162 \eq\@tags@container@block
1163 \else
1164 \eq\@eqnswfalse
1165 \fi

```

```

1166 \fi
1167 }

```

## 5.5 Positioning

**TODO:** describe

```

1168 \def\eql@tagpos@single@eval{%
1169   \if@eqnsw
1170     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1171     \ifnum\eql@tagpos@row@>\@ne
1172       \eql@tagpos@row@\@ne
1173     \fi
1174     \ifdefined\eql@tagpos@doconvert
1175       \let\eql@tagpos@continuous\eql@true
1176     \fi
1177     \ifdefined\eql@tagpos@continuous
1178       \eql@tagpos@single@eval@continuous
1179     \fi
1180   \else
1181     \eql@tagpos@row@\z@
1182   \fi
1183   \eql@tagpos@prevrow@\z@
1184   \eql@tagpos@headroom@\z@
1185   \eql@tagpos@footroom@\z@
1186 }

```

**TODO:** describe

```

1187 \def\eql@tagpos@single@eval@continuous{%
1188   \ifnum\eql@tagpos@row@>\z@
1189     \eql@tagpos@target@\eql@tagpos@shift@
1190   \else
1191     \eql@tagpos@target@\dimexpr\eql@line@height@
1192       -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1193   \fi
1194   \eql@tagpos@row@\@ne
1195   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1196     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1197   \eql@tagpos@target@\z@
1198   \fi
1199 }

```

**TODO:** describe

```

1200 \def\eql@tagpos@adjust@eval{%
1201   \if@eqnsw
1202     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1203     \ifnum\eql@tagpos@row@>\eql@totalrows@
1204       \eql@tagpos@row@\eql@totalrows@
1205     \fi
1206     \ifdefined\eql@tagpos@doconvert
1207       \let\eql@tagpos@continuous\eql@true
1208     \fi
1209     \ifdefined\eql@tagpos@continuous
1210       \ifnum\eql@tagpos@row@>\z@
1211         \eql@tagpos@adjust@eval@convert
1212       \fi
1213       \eql@tagpos@adjust@eval@continuous

```

```

1214 \fi
1215 \else
1216 \eql@tagpos@row@z@
1217 \eql@tagpos@prevrow@z@
1218 \fi
1219 }

```

**TODO:** describe

```

1220 \def\eql@tagpos@adjust@eval@convert{%
1221 \eql@tagpos@current@z@
1222 \eql@dimensions@for{%
1223 \ifnum\eql@row@<\eql@tagpos@row@
1224 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1225 +\eql@line@height@+\eql@line@depth@relax
1226 \fi
1227 \ifnum\eql@row@=\eql@tagpos@row@
1228 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1229 +\eql@line@height@-\eql@tagheight@block@relax
1230 \fi
1231 }%
1232 }

```

**TODO:** describe

```

1233 \def\eql@tagpos@adjust@eval@continuous{%
1234 \dimen@dimexpr\eql@tagpos@current@-\eql@tagpos@shift@relax
1235 \eql@tagpos@row@\eql@totalrows@
1236 \eql@tagpos@prevrow@z@
1237 \eql@tagpos@headroom@z@
1238 \eql@tagpos@footroom@z@
1239 \eql@dimensions@for{%
1240 \ifnum\eql@tagpos@row@=\eql@totalrows@
1241 \eql@tagpos@headroom@\eql@line@interline@
1242 \eql@tagpos@target@dimexpr\eql@line@interline@
1243 +\eql@line@height@-\dimen@-\eql@tagheight@block@relax
1244 \ifdim\ifdim\eql@tagpos@target@<z@-\fi
1245 \eql@tagpos@target@<\glueexpr\eql@tagpos@snaprelax
1246 \advance\dimen@\eql@tagpos@target@
1247 \eql@tagpos@target@z@
1248 \fi
1249 \ifdim\eql@tagpos@target@>%
1250 \ifdefined\eql@tagsleft-1sprelax\elsez@\fi
1251 \eql@tagpos@row@\eql@row@
1252 \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1253 \fi
1254 \advance\dimen@-\dimexpr\eql@line@interline@
1255 +\eql@line@depth@+\eql@line@height@relax
1256 \fi
1257 \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1258 \eql@tagpos@footroom@\eql@line@interline@
1259 \fi
1260 }%
1261 }

```

**TODO:** describe

```

1262 \def\eql@tagpos@print@line@eval{%
1263 \ifdefined\eql@tagpos@continuous
1264 \eql@tagpos@print@line@eval@continuous
1265 \else

```

```

1266 \eq\tagpos@print@line@eval@discrete
1267 \fi
1268 }

```

**TODO:** describe

```

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

```

```

1322     \fi
1323   \fi
1324 \else
1325   \ifnum\eql@tagpos@prevrow@=\eql@row@
1326     \eql@tagwidth@\eql@tagwidth@block@
1327   \else
1328     \let\eql@tagpos@reserve\eql@false
1329   \fi
1330 \fi
1331 }

```

**TODO:** describe

```

1332 \def\eql@tagpos@print@line@eval@discrete{%
1333   \if@eqnsw
1334     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1335     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1336     \eql@tagpos@plain@\eql@tagpos@shift@
1337     \eql@tagpos@headroom@z@
1338     \eql@tagpos@footroom@z@
1339     \ifdim\eql@tagpos@shift@>%
1340       \ifdefined\eql@tagsleft-1sp\relax\else z@\fi
1341     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1342   \else
1343     \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1344   \fi
1345 \else
1346   \let\eql@tagpos@reserve\eql@false
1347 \fi
1348 }

```

**TODO:** describe

```

1349 \def\eql@tagpos@print@line@end{%
1350   \ifdefined\eql@tagpos@continuous
1351     \ifnum\eql@tagpos@prevrow@=\eql@row@
1352       \ifdefined\eql@tagpos@reserve
1353         \global\eql@appendexpand\eql@tags@container@block{%
1354           \advance\eql@tagpos@headroom@the\dimexpr\eql@line@height@
1355             +\eql@line@depth@\relax\relax}%
1356         \eql@displaybreak@star\@M
1357       \fi
1358     \fi
1359 \fi
1360 }

```

## 5.6 Component Display

**Showkeys Integration.** **TODO:** describe

```

1361 \let\eql@SK@loaded\eql@false
1362 \let\eql@SK@label\@gobble
1363 \let\eql@SK@clearlabel\@empty
1364 \let\eql@SK@setlabel\@gobble
1365 \let\eql@SK@printlabel@right\@empty
1366 \let\eql@SK@printlabel@left\@empty
1367 \let\eql@SK@printlabel@line\@empty
1368 \def\eql@label@clean{\eql@label@org}
1369 \AddToHook{package/showkeys/after}{

```

```

1370 \let\eql@SK@loaded\eql@true
1371 \def\eql@SK@label#1{\SK@\SK@@label#1}
1372 \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1373 \eql@SK@clearlabel
1374 \def\eql@SK@@label#1>#2\SK@{%
1375   \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1376 }
1377 \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1378 \def\eql@SK@printlabel@right{%
1379   \ifx\eql@SK@lab\relax\else
1380     \rlap{\kern\marginparsep\eql@SK@lab}%
1381     \eql@SK@clearlabel
1382   \fi
1383 }
1384 \def\eql@SK@printlabel@left{%
1385   \ifx\eql@SK@lab\relax\else
1386     \llap{\eql@SK@lab\kern\marginparsep}%
1387     \eql@SK@clearlabel
1388   \fi
1389 }
1390 \def\eql@SK@printlabel@line{%
1391   \ifx\eql@SK@lab\relax\else
1392     \dimen@ \prevdepth
1393     \nointerlineskip
1394     \ifdefined\eql@tagsleft
1395       \llap{%
1396         \eql@SK@lab
1397         \kern\marginparsep
1398       }%
1399       \eql@SK@clearlabel
1400     \else
1401       \rlap{%
1402         \dimen@ \displaywidth
1403         \advance\dimen@ \marginparsep
1404         \kern\dimen@
1405         \eql@SK@lab
1406       }%
1407     \fi
1408     \eql@SK@clearlabel
1409     \prevdepth \dimen@
1410   \fi
1411 }
1412 \let\eql@label@org\label
1413 \def\eql@label@clean{\let\SK@\gobbletwo\eql@label@org}
1414 }

```

## Labels.

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

```

1415 \def\eql@composetag@label{%
1416   \eql@SK@clearlabel
1417   \ifdefined\eql@tags@label
1418     \eql@SK@setlabel\eql@tags@label
1419     \ifdefined\eql@tags@name
1420       \let\@currentlabelname\eql@tags@name
1421     \else
1422       \let\@currentlabelname\eql@tags@name@generic

```



```

1423 \fi
1424 \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1425 \fi
1426 }

```

**TODO:** describe

```

1427 \def\eql@tags@printsubeqlabel{%
1428 \eql@tags@container@parent
1429 \ifdefined\eql@tags@label
1430 \eql@tags@makeblockanchor
1431 \eql@SK@setlabel\eql@tags@label
1432 \begingroup
1433 \def\@currentcounter{equation}%
1434 \eql@tags@anchor
1435 \let\@currentlabelname\eql@tags@name@generic
1436 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1437 \expandafter\eql@label@clean\expandafter{\eql@tags@label}%
1438 \endgroup
1439 \eql@SK@printlabel@line
1440 \fi
1441 }

```

**Hyperref Anchors.** **TODO:** describe

```

1442 \let\eql@Hy@anchor\@gobble
1443 \AddToHook{package/hyperref/after}{
1444 \def\eql@Hy@anchor#1{%
1445 \Hy@raisedlink{\hyper@anchor{#1}}%
1446 }%
1447 }

```

**TODO:** describe

```

1448 \def\eql@tags@makeblockanchor{%
1449 \eql@tags@glabel@step
1450 \eql@Hy@anchor\eql@tags@glabel
1451 \edef\eql@tags@anchor{%
1452 \def\noexpand\thepage{\thepage}%
1453 \def\noexpand\@currentHref{\eql@tags@glabel}%
1454 }%
1455 }

```

**TODO:** describe

**eql@composetag@anchor**

```

1456 \def\eql@composetag@anchor{%
1457 \ifdefined\eql@tags@tag
1458 \def\@currentcounter{equation}%
1459 \ifdefined\eql@tags@ref
1460 \let\@currentlabel\eql@tags@ref
1461 \else
1462 \protected@edef\@currentlabel{\p@equation\eql@tags@tag}%
1463 \fi
1464 \eql@tags@glabel@step
1465 \edef\@currentHref{\eql@tags@glabel}%
1466 \eql@Hy@anchor\@currentHref
1467 \else

```

```

1468 \refstepcounter{equation}%
1469 \protected@edef\eql@tags@tag{\theequation}%
1470 \fi
1471 \eql@tags@anchor
1472 }

```

**Tag Layout.** **TODO:** describe

```

1473 \def\eql@tags@taglayout@set@direct#1{%
1474 \def\eql@tags@taglayout##1{#1}%
1475 }
1476 \def\eql@tags@taglayout@set#1{%
1477 \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1478 }

```

**TODO:** describe

```

1479 \def\eql@tags@tagform@set@direct#1{%
1480 \def\eql@tags@tagform##1{#1}%
1481 }
1482 \def\eql@tags@tagform@set#1#2#3{%
1483 \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@@italiccorr#3}%
1484 }

1485 \eql@tags@taglayout@set{#1}
1486 \eql@tags@tagform@set({#1})
1487 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1488 \protected\def\tagform{\eql@tags@tagform}
1489 \protected\def\tagbox{\eql@tags@taglayout}
1490 \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:

```

1491 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1492 \def\eql@composetag@tag{%
1493 \eql@tagging@tagbegin
1494 \eql@tags@frame@cmd{%
1495 \eql@tags@taglayout{%
1496 \eql@tags@tagform\eql@tags@tag
1497 \eql@tagging@tagsave
1498 }%
1499 }%
1500 \eql@tagging@tagend
1501 }

```

## 5.7 Tag Composition

**TODO:** describe

```

1502 \def\eql@composetag@measure{%
1503 \ifdefined\eql@tags@tag\else
1504 \stepcounter{equation}%
1505 \let\eql@tags@tag\theequation

```

```

1506 \fi
1507 \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1508 \ifdefined\eql@numbering@multi
1509 \global\let\eql@tags@container\eql@tags@container@clear
1510 \fi
1511 }

```

**TODO:** describe

```

1512 \def\eql@composetag@print{%
1513 \eql@composetag@anchor
1514 \eql@composetag@label
1515 \ifdefined\eql@tags@left
1516 \eql@SK@printlabel@left
1517 \eql@composetag@tag
1518 \else
1519 \eql@composetag@tag
1520 \eql@SK@printlabel@right
1521 \fi
1522 \global\let\eql@tags@container\eql@tags@container@clear
1523 }

```

**TODO:** describe

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

```

1524 \def\eql@tagbox@make#1{%
1525 \setbox\eql@tagbox@\hbox{\eql@strut@tag\@align#1}%
1526 \eql@tagwidth@\wd\eql@tagbox@
1527 \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1528 \eql@tagwidth@\eql@tagwidthmin@
1529 \fi
1530 \advance\eql@tagwidth@\eql@tagsepmin@
1531 }

```

**TODO:** describe

```

1532 \def\eql@tagbox@print@adjustheadroom{%
1533 \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1534 \ifdim\dimen@>\z@
1535 \ifdim\dimen@>\eql@tagpos@headroom@
1536 \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1537 \else
1538 \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1539 \fi
1540 \fi
1541 }

```

**TODO:** describe

```

1542 \def\eql@tagbox@print@adjustfootroom{%
1543 \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1544 \ifdim\dimen@>\z@
1545 \ifdim\dimen@>\eql@tagpos@footroom@
1546 \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1547 \else
1548 \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1549 \fi
1550 \fi
1551 }

```

**TODO:** describe

```

1552 \def\eq\tagbox@print@extendabove{%
1553   \dimen@ \dimexpr \ht\eq\tagbox@+\eq\tagpos@current@-\eq\line@height@ \relax
1554   \ifdim \dimen@ > \z@
1555     \global\eq\appendexpand\eq\display@container{%
1556       \eq\display@aboveextend@ \the\dimen@ \relax}%
1557   \fi
1558 }

```

**TODO:** describe

```

1559 \def\eq\tagbox@print@extendbelow{%
1560   \dimen@ \dimexpr \dp\eq\tagbox@-\eq\tagpos@current@-\eq\line@depth@ \relax
1561   \ifdim \dimen@ > \z@
1562     \global\eq\appendexpand\eq\display@container{%
1563       \eq\display@belowextend@ \the\dimen@ \relax}%
1564   \fi
1565 }

```

**TODO:** describe

```

1566 \def\eq\tagbox@print@prepare{%
1567   \ifdefined\eq\tagpos@reserve
1568     \eq\tagpos@current@\eq\tagpos@plain@
1569   \else
1570     \eq\tagpos@current@\eq\tagpos@raised@
1571   \fi
1572   \ifdim \eq\tagpos@headroom@ > \z@
1573     \eq\tagbox@print@adjustheadroom
1574   \fi
1575   \ifdim \eq\tagpos@footroom@ > \z@
1576     \eq\tagbox@print@adjustfootroom
1577   \fi
1578   \ifnum \eq\row@ = \one
1579     \eq\tagbox@print@extendabove
1580   \fi
1581   \ifnum \eq\row@ = \eq\totalrows@
1582     \eq\tagbox@print@extendbelow
1583   \fi
1584 }

```

**TODO:** describe

```

1585 \def\eq\tagbox@print@tagsright{%
1586   \eq\tagbox@print@prepare
1587   \kern-\wd\eq\tagbox@
1588   \raise\eq\tagpos@current@\box\eq\tagbox@
1589 }

```

**TODO:** describe

```

1590 \def\eq\tagbox@print@tagsleft{%
1591   \eq\display@firstavail@set\z@
1592   \eq\tagbox@print@prepare
1593   \wd\eq\tagbox@\z@
1594   \raise\eq\tagpos@current@\box\eq\tagbox@
1595 }

```

$\tagbox@print@cell$

```

1596 \def\eq\tagbox@print@cell{%
1597   \eq\tagging@tagaddbox

```

```

1598 \ifdefined\eq\@tagsleft
1599 \ifdefined\eq\@tagpos@reserve
1600 \ifdim\eq\@tagwidth@>\dimexpr\eq\@line@avail@+\eq\@tagfuzz@\relax
1601 \let\eq\@tagpos@reserve\eq\@false
1602 \fi
1603 \fi
1604 \if@eqnsw
1605 \eq\@tagbox@print@tagsleft
1606 \fi
1607 \kern\displaywidth
1608 \else
1609 \kern\displaywidth
1610 \ifdefined\eq\@tagpos@reserve
1611 \ifdim\eq\@tagwidth@>%
1612 \dimexpr\displaywidth-\eq\@line@width@+\eq\@tagfuzz@\relax
1613 \let\eq\@tagpos@reserve\eq\@false
1614 \fi
1615 \fi
1616 \if@eqnsw
1617 \eq\@tagbox@print@tagsright
1618 \fi
1619 \fi
1620 }

```

## 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:

```

1621 \eq\@amsmath@undefine\c@parentequation
1622 \eq\@amsmath@undefine\theparentequation
1623 \ifdefined\c@parentequation\else
1624 \newcounter{parentequation}
1625 \fi

```

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

```

1626 \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:

```

1627 \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`:

```

1628 \def\eq@subequations@init{%
1629   \edef\eq@subequations@restorecounter{%
1630     \global\c@equation\the\c@equation\relax}%
1631   \eq@tags@container@block
1632   \ifdefined\eq@tags@tag
1633     \eq@tags@glabel@step
1634     \protected@edef\theHparentequation{\eq@tags@glabel}%
1635     \protected@edef\theparentequation{\eq@tags@tag}%
1636   \else
1637     \advance\c@equation\@ne
1638     \protected@edef\theparentequation{\theequation}%
1639     \ifdefined\theHequation
1640       \protected@edef\theHparentequation{\theHequation}%
1641     \fi
1642   \fi
1643   \global\c@parentequation\c@equation
1644   \global\c@equation\z@
1645   \let\theequation\eq@subequations@template
1646   \def\theHequation{\theHparentequation.\arabic{equation}}%
1647 }
```

`\eq@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:

```

1648 \def\eq@subequations@close{%
1649   \ifnum\c@equation=\z@
1650     \eq@subequations@restorecounter
1651   \else
1652     \global\c@equation\c@parentequation
1653   \fi
1654 }
```

## 6.2 Environment

`\eq@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@printsubeqlabel`

```

1655 \def\eq@subequations@start{%
1656   \let\eq@tags@container@block\eq@tags@container@clear
1657   \eq@nextopt@process{subequations}%
1658   \eq@subequations@init
1659   \eq@tags@glabel@step
1660   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1661   \eq@Hy@anchor\eq@subequations@currentHref
1662   \edef\eq@subequations@thepage{\thepage}%
1663   \def\@currentcounter{equation}%
1664   \let\@currentHref\eq@subequations@currentHref
1665   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1666   \eq@tags@container@block
1667   \ifdefined\eq@tags@name
1668     \let\@currentlabelname\eq@tags@name
1669   \else
```

```

1670 \let\@currentlabelname\eq\@tags\@name\@generic
1671 \fi
1672 \let\eq\@subequations\@active\eq\@true
1673 \ifdefined\eq\@tags\@label
1674 \eq\@SK\@label\eq\@tags\@label
1675 \fi
1676 \ignorespaces
1677 }

```

`\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:

```

1678 \def\eq\@subequations\@end{%
1679 \ifnum\c\@equation>\z@
1680 \eq\@tags\@container\@block
1681 \ifdefined\eq\@tags\@label
1682 \begingroup
1683 \def\@currentcounter{equation}%
1684 \let\thepage\eq\@subequations\@thepage
1685 \let\@currentHref\eq\@subequations\@currentHref
1686 % \TODO how about tag* ?! also within equations!
1687 \protected\edef\@currentlabel{\p\@equation\theparentequation}%
1688 \ifdefined\eq\@tags\@name
1689 \let\@currentlabelname\eq\@tags\@name
1690 \else
1691 \let\@currentlabelname\eq\@tags\@name\@generic
1692 \fi
1693 \expandafter\eq\@label\@clean\expandafter{\eq\@tags\@label}%
1694 \endgroup
1695 \fi
1696 \fi
1697 \eq\@subequations\@close
1698 }

```

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

```

1699 \newenvironment{eq\@subequations}{%
1700 \eq\@verbose\@info\eq\@verbose\@msg\@enterenv
1701 \eq\@subequations\@testall\eq\@subequations\@start%
1702 }{%
1703 \eq\@subequations\@end
1704 \ignorespacesafterend
1705 \eq\@verbose\@info\eq\@verbose\@msg\@leaveenv
1706 }

```

**TODO:** describe

```

1707 \def\eq\@subequations\@testall{\eq\@parseopt\@main\eq\@subequations\@parse}
1708 \def\eq\@subequations\@parse{%
1709 \ifx\eq\@parseopt\@token[%]
1710 \let\eq\@parseopt\@next\eq\@parseopt\@opt
1711 \fi
1712 \ifx\eq\@parseopt\@token\eq\@atxi
1713 \let\eq\@parseopt\@next\eq\@parseopt\@label
1714 \fi
1715 \ifx\eq\@parseopt\@token\eq\@atxii
1716 \let\eq\@parseopt\@next\eq\@parseopt\@label
1717 \fi

```

```

1718 \ifx\eql@parseopt@token\label
1719 \let\eql@parseopt@next\eql@parseopt@end
1720 \fi
1721 }

```

## 6.3 Subequation Scheme

**TODO:** describe

```

1722 \def\eql@numbering@subeq@init{%
1723 \let\eql@save@theequation\theequation
1724 \let\eql@save@theHequation\theHequation
1725 \eql@subequations@init
1726 \let\eql@tags@container@parent\eql@tags@container@block
1727 \let\eql@tags@container@block\eql@tags@container@clear
1728 }

```

**TODO:** describe

```

1729 \def\eql@numbering@subeq@test{%
1730 \ifnum\eql@tagrows@<\tw@
1731 \let\eql@tags@container@block\eql@tags@container@parent
1732 \let\eql@numbering@subeq@use\eql@false
1733 \let\theequation\eql@save@theequation
1734 \let\theHequation\eql@save@theHequation
1735 \eql@subequations@restorecounter
1736 \fi
1737 }

```

**TODO:** describe

```

1738 % \TODO note must not use setcounter here (when calc is loaded)
1739 \def\eql@numbering@subeq@close{%
1740 \eql@subequations@close
1741 }

```

## 7 Display Equations Support

**TODO:** describe

```

1742 \let\eql@display@injectbefore\@undefined
1743 \let\eql@display@injectafter\@undefined
1744 \let\eql@interline@container\@undefined
1745 \def\eql@interline@container@clear{%
1746 \eql@displaybreak@open@\@MM
1747 \eql@vspaceskip@\z@skip
1748 }

```

### 7.1 Display Breaks

**TODO:** describe

erdisplaylinepenalty

```

1749 \interdisplaylinepenalty\@M

```



`\eqldisplaybreak@open` **TODO:** isn't this the opposite order than `\@getpen`?

```
1750 \def\eqldisplaybreak@open#1{%
1751   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1752 }
```

**TODO:** allow a displaybreak before equations

```
1753 \protected\def\eqldisplaybreak@default{%
1754   \eqldisplaybreak@warning{Invalid use of \string\displaybreak}{}%
1755   \eqldisplaybreak@teststaroropt@loose\@gobble\eqldisplaybreak@opt}%
1756 \eqldisplaybreak@after{\let\eqldisplaybreak@default\displaybreak}
1757 \eqldisplaybreak@let\displaybreak\eqldisplaybreak@default
```

```
1758 \newcount\eqldisplaybreak@pen@
1759 \newcount\eqldisplaybreak@prepen@
1760 \newcount\eqldisplaybreak@postpen@
```

**TODO:** describe

```
1761 \protected\def\eqldisplaybreak{%
1762   \relax
1763   \eqldisplaybreak@protecttwo\eqldisplaybreak@teststaroropt@tight
1764   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1765 }
```

```
1766 \def\eqldisplaybreak@star#1{%
1767   \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1768     \eqldisplaybreak@open@the\numexpr#1\relax\relax}%
1769 }
```

```
1770 \def\eqldisplaybreak@level[#1]{%
1771   \ifnum#1<\z@
1772     \global\eqldisplaybreak@append\eqldisplaybreak@interline@container{\eqldisplaybreak@open@\@MM}%
1773   \else
1774     \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1775       \eqldisplaybreak@open@-\@getpen{#1}\relax}%
1776   \fi
1777 }
```

**TODO:** describe

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

**TODO:** describe

```
1785 \def\eqldisplaybreak@post#1{%
1786   \ifnum#1<\z@
1787     \eqldisplaybreak@postpen@\@MM
1788   \else
1789     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
1790   \fi
1791 }
```

**TODO:** describe

```

1792 \def\eqldisplaybreak@inter#1{%
1793   \ifnum#1<\z@
1794     \interdisplaylinepenalty\@M
1795   \else
1796     \interdisplaylinepenalty\eqldgetdsp@pen{#1}\relax
1797   \fi
1798 }

```

## 7.2 Explicit Vertical Space

**TODO:** describe

`\eqlvspaceskip@` (*skip*)

```

1799 \newskip\eqlvspaceskip@

1800 \let\eqlvspace@org\vspace
1801 \def\eqlvspace{%
1802   \ifvmode
1803     \expandafter\eqlvspace@immediate
1804   \else
1805     \expandafter\eqlvspace@line
1806   \fi
1807 }

```

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

```

1808 \def\eqlvspace@line{%
1809   \eqldifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
1810 }
1811 \def\eqlvspace@add#1{%
1812   \global\eqldappendexpand\eqldinterline@container{%
1813     \advance\eqlvspaceskip@\the\glueexpr#1\relax\relax}}
1814 \def\eqlvspace@addfixedbefore#1{%
1815   \global\eqldappendexpand\eqldinterline@container{%
1816     \noexpand\eqldappend\noexpand\eqldisplay@injectbefore{%
1817       \skip@\the\glueexpr#1\relax\relax
1818       \penalty\@M
1819       \vskip\skip@
1820       \global\advance\eqldline@interline@\skip@
1821     }%
1822   }%
1823 }
1824 \def\eqlvspace@addfixedafter#1{%
1825   \global\eqldappendexpand\eqldinterline@container{%
1826     \noexpand\eqldappend\noexpand\eqldisplay@injectafter{%
1827       \dimen@\prevdepth
1828       \hrule\@height\z@
1829       \skip@\the\glueexpr#1\relax\relax
1830       \penalty\@M
1831       \vskip\skip@
1832       \global\advance\eqldline@interline@\skip@
1833       \prevdepth\dimen@
1834     }%
1835   }%
1836 }

```

**TODO:** careful to not expand `\eqldisplay@container` after measure

```

1837 \def\eqlvspace@immediate{%
1838   \noalign\bgroup
1839     \eq@ifstar@loose\eqlvspace@fixed\eqlvspace@discardable
1840 }
1841 \def\eqlvspace@fixed#1{%
1842   \skip@\glueexpr#1\relax
1843   \ifnum\eql@row@=\@ne
1844     \global\eql@appendexpand\eqldisplay@container{%
1845       \advance\eql@abovespace@\the\skip@\relax}%
1846   \else\ifnum\eql@row@>\eql@totalrows@
1847     \global\eql@appendexpand\eqldisplay@container{%
1848       \advance\eql@belowspace@\the\skip@\relax}%
1849   \else
1850     \dimen@\prevdepth
1851     \hrule\@height\z@
1852     \penalty\@M
1853     \vskip\skip@
1854     \global\advance\eql@line@interline@\skip@
1855     \prevdepth\dimen@
1856   \fi\fi
1857 \egroup
1858 }
1859 \def\eqlvspace@discardable#1{%
1860   \skip@\glueexpr#1\relax
1861   \ifnum\eql@row@=\@ne
1862     \global\eql@appendexpand\eqldisplay@container{%
1863       \advance\eql@abovespace@\the\skip@\relax}%
1864   \else\ifnum\eql@row@>\eql@totalrows@
1865     \global\eql@appendexpand\eqldisplay@container{%
1866       \advance\eql@belowspace@\the\skip@\relax}%
1867   \else
1868     \vskip\skip@
1869     \global\advance\eql@line@interline@\skip@
1870   \fi\fi
1871 \egroup
1872 }

```

### 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  
`\eql@strutbox@` of a normal `\strut` minus `\normallineskiplimit` according to M. Spivak.

```

1873 \newbox\eql@strutbox@
1874 \def\eql@strut@depth{.3}
1875 \def\eql@strut{\copy\eql@strutbox@}
1876 \let\eql@strut@cell\eql@strut
1877 \let\eql@strut@tag\eql@strut
1878 \def\eql@strut@make{%
1879   \setbox\eql@strutbox@\hbox{%
1880     \@tempdimb\dimexpr
1881       \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
1882     \@tempdima\dimexpr
1883       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
1884     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@

```

```

1885 }
1886 }
1887 \AtBeginDocument{\eql@strut@make}

```

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

```

1888 \def\eql@spread@set{%
1889   \ifdefined\eql@spread@reset
1890     \lineskip\normallineskip
1891     \lineskiplimit\normallineskiplimit
1892     \baselineskip\normalbaselineskip
1893   \fi
1894   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
1895     +\normalbaselineskip-\baselineskip\relax
1896   \ifdim\eql@spread@>\z@
1897     \openup\eql@spread@
1898     \ifdefined\spread@equation
1899       \let\spread@equation\@empty
1900     \fi
1901   \fi
1902 }

```

## 7.4 Entry and Exit

**TODO:** describe

```

1903 \let\eql@beamerbasecolor@fix\@empty
1904 \AddToHook{package/beamerbasecolor/after}{%
1905   \def\eql@beamerbasecolor@fix{%
1906     \donotcolorouterdisplaymaths
1907     \donotcoloroutermaths
1908     \beamer@setdisplaymathcolor
1909   }%
1910 }

```

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

```

1911 \newskip\eql@abovespace@
1912 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

1913 \def\eql@display@enter{%
1914   \if@noskipsec\leavevmode\par\fi
1915   \ifvmode
1916     \eql@prevdepth@\prevdepth
1917     \nointerlineskip
1918     \noindent
1919   \else
1920     \eql@prevdepth@\maxdimen
1921   \fi
1922   \eql@beamerbasecolor@fix
1923 }

```

`\eql@display@adjust`

```

1924 \def\eql@display@adjust{%
1925   \ifdefined\eql@display@linewidth

```

```

1926 \displaywidth\glueexpr\eqldisplay@linewidth\relax
1927 \advance\displaywidth-\displayindent
1928 \fi
1929 \ifdefined\eqldisplay@marginleft
1930 \advance\displaywidth\displayindent
1931 \displayindent\glueexpr\eqldisplay@marginleft\relax
1932 \advance\displaywidth-\displayindent
1933 \fi
1934 \ifdefined\eqldisplay@marginright
1935 \advance\displaywidth-\glueexpr\eqldisplay@marginright\relax
1936 \fi
1937 \ifdim\displaywidth<\z@
1938 \displaywidth\z@
1939 \fi
1940 }

```

\eqldisplay@init

```

1941 \def\eqldisplay@init{%
1942 \let\displaybreak\eqldisplaybreak
1943 \let\eqlvspace@org\vspace
1944 \let\vspace\eqlvspace
1945 \let\eqncontrol\eql@control
1946 \let\eqldisplay@injectbefore\@empty
1947 \let\eqldisplay@injectafter\@empty
1948 \let\eqnpunct\eql@punct@setnext
1949 \eql@spread@set
1950 \eql@strut@make
1951 \let\eql@frame@cmd\@undefined
1952 }

```

\eqldisplay@print

```

1953 \def\eqldisplay@print{%
1954 \eql@punct@top@set
1955 \let\eqldisplay@container\@empty
1956 \eqldisplay@firstavail@\z@
1957 \eqldisplay@aboveextend@\z@
1958 \eqldisplay@belowextend@\z@
1959 \global\let\eql@interline@container\eql@interline@container@clear
1960 }

```

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

```

1961 \def\eqldisplay@halign@init#1{%
1962 \eql@row@\z@
1963 \eql@prevgraf@\prevgraf
1964 \everycr{\noalign{%
1965 \global\advance\eql@row@\@ne
1966 \prevgraf\numexpr\prevgraf+\@ne\relax
1967 #1%
1968 }}%
1969 }

```

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

```

1970 \def\eqldisplay@halign@start{%
1971 \prevgraf\numexpr\eql@prevgraf+\@ne\relax
1972 \ifdim\eql@prevdepth>=\maxdimen\else

```

```

1973     \prevdepth\eql@prevdepth@
1974   \fi
1975   \ifdim\prevdepth=-\@m\p@\else
1976     \ifdefined\eql@display@height
1977       \skip@\baselineskip
1978       \advance\skip@-\glueexpr\eql@display@height\relax
1979       \advance\skip@-\prevdepth\relax
1980       \ifdim\skip@<\lineskiplimit
1981         \skip@\lineskip
1982       \fi
1983       \advance\skip@-\eql@spread@\relax
1984       \vskip\skip@
1985       \nointerlineskip
1986     \else
1987       \vskip-\eql@spread@\relax
1988     \fi
1989   \fi
1990 }

```

**TODO:** describe

```

1991 \def\eql@display@vspace{%
1992   \advance\abovedisplayskip\eql@abovespace@
1993   \advance\belowdisplayskip\eql@belowspace@
1994 }

```

**TODO:** describe

```

1995 \def\eql@display@vspace@native{%
1996   \advance\abovedisplayskip\eql@abovespace@
1997   \advance\belowdisplayskip\eql@belowspace@
1998   \advance\abovedisplayshortskip\eql@abovespace@
1999   \advance\belowdisplayshortskip\eql@belowspace@
2000 }

```

**TODO:** describe

```

2001 \def\eql@display@penalty{%
2002   \ifnum\eql@displaybreak@postpen@=\@MM\else
2003     \postdisplaypenalty\eql@displaybreak@postpen@
2004   \fi
2005   \ifnum\eql@displaybreak@pen@=\@MM\else
2006     \postdisplaypenalty\eql@displaybreak@pen@
2007   \fi
2008   \ifnum\eql@displaybreak@prepen@=\@MM\else
2009     \predisplayskip\eql@displaybreak@prepen@
2010   \fi
2011 }

```

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

```

2012 \def\eql@display@halign@end{%
2013   \eql@interline@container
2014   \eql@display@injectbefore
2015   \global\eql@prevgraf=\numexpr\prevgraf+\@ne\relax
2016   \ifdefined\eql@display@depth
2017     \prevdepth\glueexpr\eql@display@depth\relax
2018   \fi
2019 }

```

`\eql@display@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?

```

2020 \def\eqldisplay@close{%
2021   \eqldisplay@container
2022   \ifdim\eqldisplay@firstavail@<\z@
2023     \eqldisplay@firstavail@ \z@
2024   \fi
2025   \eqldisplay@mode@leave@ \z@
2026   \ifdim\eqldisplay@prevdepth@=\maxdimen
2027     \ifdim\predisplay@size=-\maxdimen
2028       \eqldisplay@mode@above@\eqldisplay@mode@cont@above\relax
2029       \eqldisplay@mode@below@\eqldisplay@mode@cont@below\relax
2030     \else
2031       \eqldisplay@mode@above@\z@
2032       \eqldisplay@mode@below@\z@
2033       \advance\eqldisplay@firstavail@\displayindent
2034       \ifdim\eqldisplay@firstavail@>\predisplay@size
2035         \ifcase\eqldisplay@mode@short\relax
2036         \or
2037           \eqldisplay@mode@above@\@ne
2038         \or
2039           \eqldisplay@mode@above@\@ne
2040           \ifnum\eqldisplay@totalrows@=\@ne
2041             \eqldisplay@mode@below@\@ne
2042           \fi
2043         \or
2044           \eqldisplay@mode@above@\@ne
2045           \eqldisplay@mode@below@\@ne
2046         \fi
2047       \fi
2048     \fi
2049   \else
2050     \ifdim\eqldisplay@prevdepth@=-\@m\p@
2051       \eqldisplay@mode@above@\eqldisplay@mode@top@above\relax
2052       \eqldisplay@mode@below@\eqldisplay@mode@top@below\relax
2053     \else
2054       \eqldisplay@mode@above@\eqldisplay@mode@par@above\relax
2055       \eqldisplay@mode@below@\eqldisplay@mode@par@below\relax
2056     \fi
2057   \fi
2058   \ifcase\eqldisplay@mode@above@
2059   \or\or\or
2060     \predisplaypenalty\z@
2061   \or
2062     \predisplaypenalty\z@
2063   \fi
2064   \ifcase\eqldisplay@mode@below@
2065   \or\or\or
2066     \eqldisplay@mode@leave@\@ne
2067   \or
2068     \eqldisplay@mode@leave@\tw@
2069   \fi
2070   \ifdefined\eqldisplay@force@above
2071     \eqldisplay@mode@above@\eqldisplay@force@above\relax
2072   \fi
2073   \ifdefined\eqldisplay@force@below
2074     \eqldisplay@mode@below@\eqldisplay@force@below\relax
2075   \fi

```

```

2076 \ifdefined\eq@skip@force@leave
2077   \eq@skip@mode@leave@\eq@skip@force@leave\relax
2078 \fi
2079 \ifnum\eq@skip@mode@leave@>\z@
2080   \postdisplaypenalty\z@
2081 \fi
2082 \ifcase\eq@skip@mode@above@
2083   \abovedisplayskip\glueexpr\eq@skip@long@above\relax
2084 \or
2085   \abovedisplayskip\glueexpr\eq@skip@short@above\relax
2086 \or
2087   \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
2088 \or
2089   \abovedisplayskip\glueexpr\eq@skip@par@above\relax
2090 \or
2091   \abovedisplayskip\glueexpr\eq@skip@top@above\relax
2092 \or
2093   \abovedisplayskip\z@skip
2094 \or
2095   \abovedisplayskip\glueexpr\eq@skip@med@above\relax
2096 \or
2097   \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
2098 \fi
2099 \ifcase\eq@skip@mode@below@
2100   \belowdisplayskip\glueexpr\eq@skip@long@below\relax
2101 \or
2102   \belowdisplayskip\glueexpr\eq@skip@short@below\relax
2103 \or
2104   \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
2105 \or
2106   \belowdisplayskip\glueexpr\eq@skip@par@below\relax
2107 \or
2108   \belowdisplayskip\glueexpr\eq@skip@top@below\relax
2109 \or
2110   \belowdisplayskip\z@skip
2111 \or
2112   \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2113 \or
2114   \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2115 \fi
2116 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
2117 \eq@interline@container
2118 \advance\eq@belowspace@\eq@vspaceskip@
2119 \eq@display@penalty
2120 \eq@display@vspace
2121 \skip@\glueexpr\eq@skip@tag@above\relax
2122 \ifdim\skip@>\abovedisplayskip
2123   \skip@\abovedisplayskip
2124 \fi
2125 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
2126 \ifdim\abovedisplayskip<\skip@
2127   \abovedisplayskip\skip@
2128 \fi
2129 \skip@\glueexpr\eq@skip@tag@below\relax
2130 \ifdim\skip@>\belowdisplayskip
2131   \skip@\belowdisplayskip
2132 \fi
2133 \ifdim\eq@display@belowextend@>\z@

```



```

2134 \advance\belowdisplayskip-\eql@display@belowextend@\relax
2135 \ifdim\belowdisplayskip<\skip@
2136 \belowdisplayskip\skip@
2137 \fi
2138 \fi
2139 }

```

**TODO:** describe

```

2140 \def\eql@display@leave{%
2141 \prevgraf\eql@prevgraf@
2142 \ifcase\eql@skip@mode@leave@
2143 \or
2144 \endgraf
2145 \or
2146 \endgraf
2147 \prevdepth-\@m\p@
2148 \fi
2149 }

```

**TODO:** describe

```

2150 \def\eql@display@nest{%
2151 \let\displaybreak\eql@displaybreak@default
2152 \let\intertext\eql@intertext@default
2153 \let\vspace\eql@vspace@org
2154 }

```

**TODO:** describe

```

2155 \def\eql@display@restore{%
2156 \let\label\eql@label@org
2157 \let>tag\eql>tag@default
2158 \let\raisetag\eql@raisetag@default
2159 \let\displaybreak\eql@displaybreak@default
2160 \let\intertext\eql@intertext@default
2161 \let\vspace\eql@vspace@org
2162 \let\eqnpunct\eql@punct@addopt
2163 \let\eql@punct@block\@undefined
2164 }

```

**TODO:** describe

```

2165 \eql@append\@arrayparboxrestore{%
2166 \eql@display@restore
2167 \ifdefined\eql@ampproof@active
2168 \eql@amprevert
2169 \fi
2170 \@displayfalse
2171 }

```

## 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)

```

2172 \def\eql@stack@enable{%
2173 \let\eql@stack@save@equations\eql@stack@save@equations@
2174 \let\eql@stack@save@box\eql@stack@save@box@

```

2175 }

**TODO:** describe

2176 \let\eq\stack@save@equations\eq\stack@enable

2177 \let\eq\stack@save@box\eq\stack@enable

2178 \let\eq\stack@restore\@empty

**TODO:** describe

2179 \def\eq\stack@save@reg#1{\global#1\the#1\relax}

2180 \def\eq\stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}

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

\eq\@prevgraf@, \eq\@skip@mode@leave@, \eq\@shape@lastrow, \eq\@frame@prevcmd

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

2181 \def\eq\stack@save@equations@{%

2182 \let\eq\stack@numbering@eqnswinit\eq\@numbering@eqnswinit

2183 \let\eq\stack@cell@container\eq\@cell@container

2184 \let\eq\stack@tags@container\eq\@tags@container

2185 \let\eq\stack@interline@container\eq\@interline@container

2186 \let\eq\stack@dimensions@tab\eq\@dimensions@tab

2187 \let\eq\stack@block@container\eq\@display@container

2188 \let\eq\stack@punct@top\eq\@punct@top

2189 \edef\eq\stack@restore{%

2190 \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi

2191 \eq\stack@save@let\eq\stack@numbering@eqnswinit\eq\@numbering@eqnswinit

2192 \eq\stack@save@let\eq\stack@cell@container\eq\@cell@container

2193 \eq\stack@save@let\eq\stack@tags@container\eq\@tags@container

2194 \eq\stack@save@let\eq\stack@interline@container\eq\@interline@container

2195 \eq\stack@save@let\eq\stack@dimensions@tab\eq\@dimensions@tab

2196 \eq\stack@save@let\eq\stack@block@container\eq\@display@container

2197 \eq\stack@save@let\eq\stack@punct@top\eq\@punct@top

2198 \eq\stack@save@reg\eq\@column@

2199 \eq\stack@save@reg\eq\@totalcolumns@

2200 \eq\stack@save@reg\eq\@line@avail@

2201 \eq\stack@save@reg\eq\@line@pos@

2202 \eq\stack@save@reg\eq\@line@width@

2203 \eq\stack@save@reg\eq\@line@depth@

2204 \eq\stack@save@reg\eq\@line@height@

2205 \eq\stack@save@reg\eq\@line@prevdepth@

2206 \eq\stack@save@reg\eq\@line@interline@

2207 \eq\stack@save@reg\eq\@totalheight@

2208 \eq\stack@save@reg\eq\@tagwidth@max@

2209 \eq\stack@save@reg\eq\@tagpos@row@

2210 \eq\stack@save@reg\eq\@row@

2211 \eq\stack@save@reg\eq\@tagrows@

2212 }%

2213 }

**TODO:** describe

2214 \def\eq\stack@save@box@{%

2215 \let\eq\stack@cell@container\eq\@cell@container

2216 \edef\eq\stack@restore{%

2217 \eq\stack@save@let\eq\stack@cell@container\eq\@cell@container

2218 \eq\stack@save@reg\eq\@row@

2219 }%

2220 }

## 8 Multi-Line Support

**TODO:** describe

### 8.1 Measure Support

**TODO:** describe

```
2221 \def\eq@measure@init#1#2{%
2222   \eq@dimensions@reset
2223   \let\eq@display@container\@empty
2224   \eq@numbering@measure@init
2225   \eq@row@\z@
2226   \eq@totalheight@\z@
2227   \eq@totalrows@\@M
2228   \eq@line@prevdepth@-\@m\p@
2229   \eq@line@interline@\z@
2230   \tabskip\z@skip
2231   \everycr{\noalign{%
2232     \global\advance\eq@row@\@ne
2233     #1%
2234   }}%
2235   \eq@punct@top@set
2236   \global\let\eq@interline@container\eq@interline@container@clear
2237   \eq@measure@savestate
2238   \eq@display@halign@letcr{#2}%
2239 }
```

**TODO:** describe

```
2240 \def\eq@measure@tag{%
2241   \eq@tagwidth@\z@
2242   \ifdefined\eq@numbering@multi
2243     \if@eqnsw
2244       \eq@tags@container
2245       \eq@tagbox@make\eq@composetag@measure
2246       \ifdefined\eq@tagpos@reserve\else
2247         \eq@tagwidth@\z@
2248       \fi
2249     \fi
2250   \fi
2251 }
```

**TODO:** describe

```
2252 \def\eq@measure@endrow{%
2253   \ifdim\eq@line@prevdepth@=-\@m\p@\else
2254     \dimen@\dimexpr\baselineskip-\eq@line@height@-\eq@line@prevdepth@\relax
2255     \ifdim\dimen@<\lineskiplimit
2256       \dimen@\lineskip
2257     \fi
2258     \advance\eq@line@interline@\dimen@
2259   \fi
2260   \eq@dimensions@endrow
2261   \ifdim\eq@tagwidth@>\eq@tagwidth@max@
2262     \global\eq@tagwidth@max@\eq@tagwidth@
2263   \fi
2264   \ifdim\eq@tagwidth@>\z@
```

```

2265 \global\advance\eql@tagrows@\@ne
2266 \fi
2267 \global\advance\eql@totalheight@\dimexpr
2268 \eql@line@interline@+\eql@line@height@+\eql@line@depth@
2269 \global\eql@line@interline@\z@
2270 \global\eql@line@prevdepth@\eql@line@depth@
2271 }

```

**TODO:** describe

```

2272 \def\eql@measure@close{%
2273 \advance\eql@row@-\tw@
2274 \eql@totalrows@\eql@row@
2275 \ifnum\eql@totalrows@>\z@
2276 \eql@dimensions@get\@ne
2277 \eql@topheight@\dimexpr\eql@line@height@+\eql@line@interline@\relax
2278 \eql@dimensions@get\eql@totalrows@
2279 \eql@bottomdepth@\eql@line@depth@
2280 \fi
2281 \eql@numbering@measure@blocktag
2282 \begingroup
2283 \eql@tags@container
2284 \if@eqnsw
2285 \eql@tagbox@make\eql@composetag@measure
2286 \ifdefined\eql@tagpos@reserve\else
2287 \eql@tagwidth@\z@
2288 \fi
2289 \eql@dimensions@saveblocktag
2290 \else
2291 \eql@dimensions@savenoblocktag
2292 \eql@numbering@warnunused
2293 \fi
2294 \endgroup
2295 \eql@dimensions@get\z@
2296 \eql@measure@restorestate
2297 }

```

measure@restorestate

eql@measure@savestate

```

2298 \let\eql@measure@restorestate\@empty
2299 \def\eql@measure@savestate{%
2300 \begingroup
2301 \def\@elt##1{%
2302 \global\csname c@##1\endcsname\the\csname c@##1\endcsname}%
2303 \global\edef\@gtempa{\cl@@ckpt}%
2304 \endgroup
2305 \let\eql@measure@restorestate\@gtempa
2306 }

```

## 8.2 Line Breaks

**TODO:** describe

\eql@display@cr

```

2307 \def\eql@display@cr{%
2308 \let\eql@punct@termcr\eql@false
2309 \eql@ampprotect\eql@display@cr@testall\eql@display@cr@process}

```

```

1@display@cr@testall TODO: describe
eq1@display@cr@parse
2310 \def\eq1@display@cr@testall{\eq1@parseopt@aux\eq1@display@cr@parse}
2311 \def\eq1@display@cr@parse{%
2312   \ifx\eq1@parseopt@token[%
2313     \let\eq1@parseopt@next\eq1@display@cr@parse@vspace
2314   \fi
2315   \ifx\eq1@parseopt@token*%
2316     \let\eq1@parseopt@next\eq1@display@cr@parse@star
2317   \fi
2318   \ifx\eq1@parseopt@token.%
2319     \let\eq1@parseopt@next\eq1@parseopt@punctpass
2320   \fi
2321   \ifx\eq1@parseopt@token,%
2322     \let\eq1@parseopt@next\eq1@parseopt@punctpass
2323   \fi
2324   \ifx\eq1@parseopt@token~%
2325     \let\eq1@parseopt@next\eq1@display@cr@parse@cont
2326   \fi
2327   \ifx\eq1@parseopt@token'%
2328     \let\eq1@parseopt@next\eq1@parseopt@punctnext
2329   \fi
2330   \ifx\eq1@parseopt@token!%
2331     \let\eq1@parseopt@next\eq1@parseopt@puncttermcr
2332   \fi
2333   \ifx\eq1@parseopt@token&%
2334     \let\eq1@parseopt@next\eq1@parseopt@end
2335   \fi
2336 }
2337 \def\eq1@display@cr@parse@vspace[#1]{\eq1@vspace@add{#1}\eq1@parseopt@peek}
2338 \def\eq1@display@cr@parse@star#1{\eq1@displaybreak@star\@M\eq1@parseopt@peek}
2339 \def\eq1@display@cr@parse@cont#1{\numbernext\eqnpunct~\eq1@parseopt@peek}

1@display@cr@process

2340 \def\eq1@display@cr@process{%
2341   \ifdefined\eq1@punct@term@cr\eq1@punct@apply@top\fi
2342   \eq1@display@endline
2343   \cr

2344   \noalign{%
2345     \eq1@interline@container
2346     \eq1@display@injectbefore
2347     \ifnum\eq1@displaybreak@pen@=\@MM
2348       \penalty\interdisplaylinepenalty
2349     \else
2350       \penalty\eq1@displaybreak@pen@
2351     \fi
2352     \vskip\eq1@vspaceskip@
2353     \global\advance\eq1@line@interline@\eq1@vspaceskip@
2354     \eq1@display@injectafter
2355     \global\let\eq1@interline@container\eq1@interline@container@clear
2356   }%
2357 }

display@halign@letcr

2358 \def\eq1@display@halign@letcr#1{%
2359   \let\\eq1@display@cr

```

```

2360 \let\eqldisplay@endline#1%
2361 }

```

### 8.3 Intertext

**TODO:** describe

**TODO:** revert in everymath?

```

2362 \def\eqlintertext@default{\eql@error{Invalid use of \string\intertext}}
2363 \eql@amsmath@let\intertext\eqlintertext@default

```

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

```

2364 \def\eqlintertext@process{%
2365   \eqldisplay@endline
2366   \cr
2367   \ifmeasuring@
2368     \expandafter\@gobble
2369   \else
2370     \expandafter\eqlintertext@print
2371   \fi
2372 }

```

**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

```

2373 \def\eqlintertext@print#1{%
2374   \noalign{%
2375     \eqldisplay@halign@end
2376     \let\eql@skip@force@below\z@
2377     \let\eql@skip@force@above\z@
2378     \eql@setkeys{intertext}\eqlintertext@opt
2379     \openup-\eql@spread@
2380     \penalty\postdisplaypenalty
2381     \ifcase\eql@skip@force@below\relax
2382       \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2383     \or
2384       \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2385     \or
2386       \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2387     \or
2388       \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2389     \or
2390       \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2391     \or
2392       \advance\eql@vspaceskip@\z@skip
2393     \or
2394       \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2395     \or
2396       \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax
2397     \fi
2398     \vskip\eql@vspaceskip@
2399     \global\let\eql@interline@container\eql@interline@container@clear
2400     \vbox{%
2401       \@parboxrestore
2402       \ifdim
2403         \ifdim@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2404       \else
2405         \parshape\@ne

```

```

2406      \@totalleftmargin\linewidth
2407      \fi
2408      \noindent
2409      \prevgraf\eql@prevgraf@
2410      \ignorespaces
2411      #1%
2412      \par
2413      \global\eql@prevgraf@\prevgraf
2414    }%
2415    \penalty\predisplaypenalty
2416    \ifcase\eql@skip@force@above\relax
2417      \vskip\glueexpr\eql@skip@long@above\relax
2418    \or
2419      \vskip\glueexpr\eql@skip@short@above\relax
2420    \or
2421      \vskip\glueexpr\eql@skip@cont@above\relax
2422    \or
2423      \vskip\glueexpr\eql@skip@par@above\relax
2424    \or
2425      \vskip\glueexpr\eql@skip@top@above\relax
2426    \or
2427      \vskip\z@skip
2428    \or
2429      \vskip\glueexpr\eql@skip@med@above\relax
2430    \or
2431      \vskip\glueexpr\eql@skip@custom@above\relax
2432    \fi
2433    % \eql@prevdepth@\maxdimen
2434    \eql@prevdepth@\z@
2435    \eql@display@halign@start
2436  }
2437 }

```

**TODO:** describe

```

2438 \newenvironment{eql@intertext}{%
2439   \eql@testopt@tight\eql@intertext@{}}%
2440 }{%
2441   \aftergroup\eql@intertext@after
2442   \ignorespacesafterend
2443 }

```

**TODO:** describe

```

2444 \def\eql@intertext@env{intertext}
2445 \def\eql@intertext@[#1]{%
2446   \global\def\eql@intertext@opt{#1}%
2447   \ifx\@currenvir\eql@intertext@env
2448     \expandafter\eql@scan@env\expandafter\eql@intertext@inject
2449   \else
2450     \expandafter\eql@intertext@process
2451   \fi
2452 }

```

**TODO:** describe

```

2453 \def\eql@intertext@inject{%
2454   \global\edef\eql@intertext@after{%
2455     \noexpand\eql@intertext@process{%
2456       \ifx\eql@scan@body\eql@scan@body@dump
2457         \eql@scan@body@dump

```

```

2458     \else
2459         \noexpand\scantokens{\eql@scan@body@dump}%
2460     \fi
2461 }%
2462 }%
2463 }

```

## 8.4 Line Marks

**TODO:** describe

```

2464 \def\eql@markline@pos@below{below}
2465 \def\eql@markline@pos@bottom{bottom}
2466 \def\eql@markline@pos@baseline{baseline}
2467 \let\eql@markline@pos\eql@markline@pos@baseline
2468 \let\eql@markline@shift\z@
2469 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2470 \def\eql@markline@symbol{}

```

**TODO:** describe

```

2471 \def\eql@markline@select#1{%
2472     \let\eql@markline@shift\z@
2473     \eql@setkeys{markline}{#1}%
2474     \eql@markline@print
2475 }

```

**TODO:** describe

```

2476 \def\eql@markline@print{%
2477     \dimen@=\dimexpr\eql@markline@shift\relax
2478     \ifx\eql@markline@pos\eql@markline@pos@below
2479         \ifdim\dimen@=\z@\else
2480             \penalty\@M
2481             \vskip-\dimen@
2482         \fi
2483         \nointerlineskip
2484         \penalty\@M
2485         \vbox{\hfill\hbox{\eql@markline@symbol}}}%
2486     \else
2487         \ifx\eql@markline@pos\eql@markline@pos@baseline
2488             \advance\dimen@\prevdepth
2489         \fi
2490         \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}}%
2491         \dimen@\prevdepth
2492         \ht\z@\z@
2493         \dp\z@\z@
2494         \nointerlineskip
2495         \penalty\@M
2496         \vbox{\hfill\box\z@}%
2497         \prevdepth\dimen@
2498     \fi
2499 }

```

**TODO:** describe

```

2500 \def\eql@markline@inject#1{%
2501     \let\eql@markline@push\eql@false
2502     \ifx\eql@markline@pos\eql@markline@pos@below\else

```



```

2503 \ifdefined\eql@tagsleft\else
2504 \ifx\eql@equations@main\eql@multi@main
2505 \ifdefined\eql@numbering@multi
2506 \if@eqnsw
2507 \let\eql@markline@push\eql@true
2508 \fi
2509 \else
2510 \ifnum\eql@row@=\eql@tagpos@row@
2511 \let\eql@markline@push\eql@true
2512 \fi
2513 \fi
2514 \else
2515 \if@eqnsw
2516 \let\eql@markline@push\eql@true
2517 \fi
2518 \fi
2519 \fi
2520 \fi
2521 \ifdefined\eql@markline@push
2522 \global\eql@append\eql@interline@container{%
2523 \eql@append\eql@display@injectbefore{\eql@markline@select{push,#1}}}%
2524 \else
2525 \global\eql@append\eql@interline@container{%
2526 \eql@append\eql@display@injectbefore{\eql@markline@select{#1}}}%
2527 \fi
2528 }

```

**TODO:** describe

```

2529 \def\eql@markline@amsthm@opt[#1]{\eql@markline@inject{qed,#1}}
2530 \def\eql@markline@amsthm@staropt[#1]{\eql@markline@inject{qed,push,#1}}
2531 \def\eql@markline@amsthm@qed{\eql@teststaropt@tight
2532 \eql@markline@amsthm@staropt\eql@markline@amsthm@opt{}}
2533 \def\eql@markline@amsthm@register#1{\eql@letcs{#1@qed}\eql@markline@amsthm@qed}
2534 \def\eql@markline@amsthm@move#1#2{%
2535 \AddToHook{package/amsthm/after}{%
2536 \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:

```

2537 \newcount\eql@shape@pos@
2538 \newdimen\eql@shape@amount@
2539 \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

$\eql@marginright@$  (*dimen*)

$\eql@centeroffset@$  (*dimen*) 2540 \newdimen\eql@marginleft@

```

2541 \newdimen\eql@marginright@
2542 \newdimen\eql@marginleft@min@
2543 \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:

```

2544 \def\eql@shape@tab@default{default}
2545 \def\eql@shape@tab@left{left}
2546 \def\eql@shape@tab@center{center}
2547 \def\eql@shape@tab@right{right}
2548 \def\eql@shape@tab@first{first}
2549 \def\eql@shape@tab@hanging{hanging}
2550 \def\eql@shape@tab@steps{steps}

```

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

```

2551 \let\eql@shape@tab@def\eql@shape@tab@default
2552 \let\eql@shape@tab@\eql@shape@tab@default
2553 \let\eql@shape@tab@l\eql@shape@tab@left
2554 \let\eql@shape@tab@c\eql@shape@tab@center
2555 \let\eql@shape@tab@r\eql@shape@tab@right
2556 \let\eql@shape@tab@rc\eql@shape@tab@first
2557 \let\eql@shape@tab@indent\eql@shape@tab@first
2558 \let\eql@shape@tab@hang\eql@shape@tab@hanging
2559 \let\eql@shape@tab@lc\eql@shape@tab@hanging
2560 \let\eql@shape@tab@outdent\eql@shape@tab@hanging
2561 \let\eql@shape@tab@lcr\eql@shape@tab@steps

```

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

```

2562 \let\eql@shape@mode\eql@shape@tab@default

```

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

```

2563 \def\eql@shape@set#1{%
2564   \ifcsname eql@shape@tab@#1\endcsname
2565     \expandafter\let\expandafter\eql@shape@mode
2566       \csname eql@shape@tab@#1\endcsname
2567   \else
2568     \eql@error{shape '#1' unknown: setting to default}%
2569     \let\eql@shape@mode\eql@shape@tab@default
2570   \fi
2571 }

```

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

```

2572 \def\eql@shape@layoutcenter@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2573 \def\eql@shape@layoutcenter@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}
2574 \def\eql@shape@layoutcenter@right{\eql@shape@pos@tw@\eql@shape@amount@z@}
2575 \let\eql@shape@layoutcenter@default\eql@shape@layoutcenter@center
2576 \def\eql@shape@layoutleft@left{\eql@shape@pos@z@\eql@shape@amount@z@}
2577 \def\eql@shape@layoutleft@center{\eql@shape@pos@\@ne\eql@shape@amount@z@}

```

```

2578 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2579 \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):

```

2580 \def\eq@shape@layoutcenter@first{%
2581   \eq@shape@pos@z@
2582   \eq@shape@amount@z@
2583   \ifnum\eq@totalrows@>\@ne
2584     \ifnum\eq@row@=\@ne
2585       \eq@shape@amount@\eq@indent@
2586     \fi
2587   \fi
2588 }
2589 \def\eq@shape@layoutleft@first{%
2590   \eq@shape@pos@z@
2591   \eq@shape@amount@z@
2592   \ifnum\eq@totalrows@>\@ne
2593     \ifnum\eq@row@=\@ne
2594       \eq@shape@amount@\eq@indent@
2595     \fi
2596   \fi
2597 }

```

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:

```

2598 \def\eq@shape@layoutcenter@hanging{%
2599   \eq@shape@pos@z@
2600   \eq@shape@amount@\eq@indent@
2601   \ifnum\eq@totalrows@>\@ne
2602     \ifnum\eq@row@=\@ne
2603       \eq@shape@amount@z@
2604     \fi
2605   \fi
2606 }
2607 \def\eq@shape@layoutleft@hanging{%
2608   \eq@shape@pos@z@
2609   \eq@shape@amount@z@
2610   \ifnum\eq@totalrows@>\@ne
2611     \ifnum\eq@row@=\@ne
2612       \eq@shape@amount@-\eq@indent@
2613     \fi
2614   \fi
2615 }

```

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:

```

2616 \def\eq@shape@layoutcenter@steps{%
2617   \eq@shape@amount@z@
2618   \eq@shape@pos@\@ne
2619   \ifnum\eq@totalrows@>\@ne
2620     \ifnum\eq@row@=\@ne
2621       \eq@shape@pos@z@
2622     \fi
2623     \ifnum\eq@row@=\eq@totalrows@

```

```

2624     \eq@shape@pos@tw@
2625     \fi
2626 \fi
2627 }
2628 \def\eq@shape@layoutleft@steps{%
2629   \eq@shape@pos@z@
2630   \eq@shape@amount@z@
2631   \ifnum\eq@totalrows@>\@ne
2632     \ifnum\eq@row@=\@ne
2633       \eq@shape@amount@-\eq@indent@
2634     \fi
2635     \ifnum\eq@row@=\eq@totalrows@
2636       \eq@shape@amount@\eq@indent@
2637     \fi
2638 \fi
2639 }

```

`\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`:

```

2640 \let\eq@shape@eval\undefined
2641 \def\eq@shape@select{%
2642   \expandafter\let\expandafter\eq@shape@eval
2643     \csname eq@shape%
2644       @\ifdefined\eq@layoutleft layoutleft\else layoutcenter\fi
2645       @\eq@shape@mode\endcsname
2646 }

```

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

```

2647 \protected\def\eq@shape@alignleft{%
2648   \global\eq@append\eq@cell@container{\eq@shape@pos@z@}%
2649   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2650 \protected\def\eq@shape@aligncenter{%
2651   \global\eq@append\eq@cell@container{\eq@shape@pos@\@ne}%
2652   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2653 \protected\def\eq@shape@alignright{%
2654   \global\eq@append\eq@cell@container{\eq@shape@pos@tw@}%
2655   \eq@ampprotect\eq@shape@align@testpar\eq@shape@alignamount@opt}
2656 \def\eq@shape@align@testpar#1{%
2657   \eq@ifstar@tight{#1[\eq@indent@]}%
2658   {\eq@ifnextgobble@tight{!}{#1[-\eq@indent@]}%
2659   {\eq@testopt@tight{#1}\z@}}
2660 \def\eq@shape@alignamount@opt[#1]{\eq@shape@alignamount@set{#1}}

```

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

```

2661 \protected\def\eq@shape@alignamount{%
2662   \eq@ampprotecttwo\eq@ifstar@tight
2663   \eq@shape@alignamount@set\eq@shape@alignamount@add}
2664 \def\eq@shape@alignamount@add#1{%
2665   \global\eq@appendexpand\eq@cell@container{%
2666     \advance\eq@shape@amount@the\glueexpr#1\relax\relax}}
2667 \def\eq@shape@alignamount@set#1{%
2668   \global\eq@appendexpand\eq@cell@container{%
2669     \eq@shape@amount@the\glueexpr#1\relax\relax}}
2670 \def\eq@shape@align@enable{%
2671   \let\shoveleft\eq@shape@alignleft

```

```

2672 \let\shovecenter\eq\shape@aligncenter
2673 \let\shoveright\eq\shape@alignright
2674 \let\shoveby\eq\shape@alignamount
2675 }

```

**TODO:** describe

```

2676 \protected\def\eq\shape@align@default{%
2677   \eq\warn@here{\shove...}%
2678   \eq\ampprotect\eq\shape@align@testpar\eq@gobbleopt}
2679 \protected\def\eq\shape@alignamount@default{%
2680   \eq\warn@here{\shove...}%
2681   \eq\ampprotecttwo\eq\ifstar@tight\@gobble\@gobble}
2682 \def\eq\shape@align@disable{%
2683   \let\shoveleft\eq\shape@align@default
2684   \let\shovecenter\eq\shape@align@default
2685   \let\shoveright\eq\shape@align@default
2686   \let\shoveby\eq\shape@alignamount@default
2687 }

```

### 9.3 Width Data

$\text{width@block@}$  (*dimen*)

```

2688 \newdimen\eq\tagwidth@block@
2689 \newdimen\eq\tagheight@block@
2690 \newdimen\eq\tagdepth@block@

```

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

```

2691 \let\eq\dimensions@tab\@empty

```

$\text{eq@dimensions@reset}$

```

2692 \def\eq\dimensions@reset{%
2693   \let\eq\dimensions@tab\@empty
2694   \eq\tagwidth@max@z@
2695   \eq\tagrows@z@
2696 }

```

$\text{eq@dimensions@add}$

```

2697 \def\eq\dimensions@add#1{%
2698   \global\eq\appendexpand\eq\dimensions@tab{#1}%
2699 }

```

$\text{eq@dimensions@addreg}$

```

2700 \def\eq\dimensions@addreg#1{#1\the#1\relax}

```

$\text{eq@dimensions@startrow}$

```

2701 \def\eq\dimensions@startrow{%
2702   \eq\dimensions@add{\eq\dimensions@addreg\eq\row@}%
2703 }

```

$\text{eq@dimensions@savecell}$

```

2704 \def\eq\dimensions@savecell{%

```

```

2705 \eqldimensions@add{%
2706   \eqldimensions@addreg\eql@shape@pos@
2707   \eqldimensions@addreg\eql@cellwidth@
2708   \eqldimensions@addreg\eql@shape@amount@
2709   \noexpand\eqldimensions@cellcall
2710 }%
2711 }

```

l@dimensions@savesep

```

2712 \def\eqldimensions@savesep{%
2713   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2714 }

```

ql@dimensions@endrow

```

2715 \def\eqldimensions@endrow{%
2716   \eqldimensions@add{,%
2717     \eqldimensions@addreg\eql@tagwidth@
2718     \eqldimensions@addreg\eql@line@height@
2719     \eqldimensions@addreg\eql@line@depth@
2720     \eqldimensions@addreg\eql@line@interline@
2721   };}%
2722 }

```

ensions@saveblocktag

```

2723 \def\eqldimensions@saveblocktag{%
2724   \eqldimensions@add{\eql@row@0\relax,%
2725     \eql@tagwidth@block@\the\eql@tagwidth@\relax
2726     \eql@tagheight@block@\the\ht\eql@tagbox@\relax
2727     \eql@tagdepth@block@\the\dp\eql@tagbox@\relax
2728     \eqldimensions@addreg\eql@tagpos@shift@
2729     \let\noexpand\eql@tagpos@reserve\ifdefined\eql@tagpos@reserve
2730     \noexpand\eql@true\else\noexpand\eql@false\fi
2731   };}%
2732   \global\eql@tagwidth@max@\eql@tagwidth@
2733   \global\eql@tagrows@\@ne
2734 }

```

sions@savenoblocktag

```

2735 \def\eqldimensions@savenoblocktag{%
2736   \eqldimensions@add{\eql@row@0\relax,;%
2737 }

```

\eqldimensions@for

```

2738 \def\eqldimensions@for#1{%
2739   \def\eqldimensions@forcall{#1}%
2740   \expandafter\eqldimensions@forstep\eqldimensions@tab
2741 }

```

l@dimensions@forstep

```

2742 \def\eqldimensions@forstep\eql@row@#1\relax#2,#3;%
2743   \eql@row@#1\relax
2744   \ifnum\eql@row@=\z@\else
2745     #3%

```

```

2746 \def\eqldimensions@cells{#2}%
2747 \eqldimensions@forall
2748 \expandafter\eqldimensions@forstep
2749 \fi
2750 }

```

\eqldimensions@get

```

2751 \def\eqldimensions@get#1{%
2752 \eqldimensions@row#1\relax
2753 \expandafter\eqldimensions@getdef\expandafter{\the\eqldimensions@row}%
2754 \expandafter\eqldimensions@getparse\eqldimensions@tab\@nil
2755 }

```

\eqldimensions@getdef

```

2756 \def\eqldimensions@getdef#1{%
2757 \def\eqldimensions@getparse
2758 ##1\eqldimensions@row#1\relax##2,##3;##4\@nil{%
2759 ##3%
2760 \def\eqldimensions@cells{##2}%
2761 }%
2762 }

```

\eqldimensions@colwidth@tab

```

2763 \let\eqldimensions@colwidth@tab\@empty

```

\eqldimensions@colwidth@get

```

2764 \def\eqldimensions@colwidth@get#1{%
2765 \ifcase\expandafter#1\eqldimensions@colwidth@tab\else\z@\fi
2766 }

```

\eqldimensions@colwidth@save

```

2767 \def\eqldimensions@colwidth@save#1{%
2768 \edef\eqldimensions@colwidth@tab{%
2769 \noexpand\or\the#1%
2770 \unexpanded\expandafter{\eqldimensions@colwidth@tab}%
2771 }%
2772 }

```

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

```

2773 \def\eqldimensions@calc{%
2774 \eqldimensions@column@\z@
2775 \eqldimensions@line@pos@\z@
2776 \eqldimensions@line@possep@\z@
2777 \eqldimensions@line@avail@\eqldimensions@totalwidth@
2778 \eqldimensions@line@availsep@\eqldimensions@intercolumns@
2779 \eqldimensions@line@width@\z@
2780 \eqldimensions@line@widthsep@\z@
2781 \let\eqldimensions@cellcall\eqldimensions@calc@call

```

```

2782 \let\eqldimensions@sepcall\eqldimensions@calc@callsep
2783 \eqldimensions@cells
2784 }

```

ensions@calc@callsep Callback for each intercolumn space.

```

2785 \def\eqldimensions@calc@callsep{%
2786 \advance\eqlline@possep@ \@ne
2787 }%

```

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

```

2788 \def\eqldimensions@calc@call{%
2789 \advance\eqllcolumn@ \@ne
2790 \ifnum\eqlltotalcolumns@=\@ne
2791 \dimen@ \eqlltotalwidth@
2792 \else
2793 \dimen@ \eqllcolwidth@get\eqllcolumn@ \relax
2794 \fi
2795 \ifdim\eqllcellwidth@>\z@
2796 \ifdim\eqllline@width@=\z@
2797 \eqllline@avail@ \eqllline@pos@
2798 \eqllline@availsep@ \eqllline@possep@
2799 \ifcase\eqllshape@pos@
2800 \or
2801 \advance\eqllline@avail@ \dimexpr
2802 (\dimen@-\eqllcellwidth@+\eqllcenteroffset@)/\tw@ \relax
2803 \or
2804 \advance\eqllline@avail@ \dimexpr \dimen@-\eqllcellwidth@ \relax
2805 \fi
2806 \advance\eqllline@avail@ \eqllshape@amount@
2807 \fi
2808 \eqllline@width@ \eqllline@pos@
2809 \eqllline@widthsep@ \eqllline@possep@
2810 \ifcase\eqllshape@pos@
2811 \advance\eqllline@width@ \eqllcellwidth@
2812 \or
2813 \advance\eqllline@width@ \dimexpr
2814 (\dimen@+\eqllcellwidth@+\eqllcenteroffset@)/\tw@ \relax
2815 \or
2816 \advance\eqllline@width@ \dimen@
2817 \fi
2818 \advance\eqllline@width@ \eqllshape@amount@
2819 \fi
2820 \advance\eqllline@pos@ \dimen@
2821 }

```

## 9.4 Best Line Selection

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

```

2822 \let\eqllnumbering@best@auto\eqllfalse

```

g@best@row@ (*counter*)

g@best@space@ (*dimen*)

bering@best@use (*bool*)

```

2823 \newcount\eqllnumbering@best@row@

```



```

2824 \newdimen\eql@numbering@best@space@
2825 \let\eql@numbering@best@use\eql@false

```

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

```

2826 \def\eql@numbering@best@find{%
2827   \eql@numbering@best@row@ \z@
2828   \eql@numbering@best@space@ \z@
2829   \eql@dimensions@for{%
2830     \eql@dimensions@calc
2831     \ifdefined\eql@tagsleft
2832       \dimen@ \eql@line@avail@
2833     \else
2834       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
2835     \fi
2836     \ifdim \dimen@ > \eql@numbering@best@space@
2837       \eql@numbering@best@row@ \eql@row@
2838       \eql@numbering@best@space@ \dimen@
2839     \fi
2840   }%
2841   \ifnum \eql@numbering@best@row@ > \z@
2842     \eql@tagpos@row@ \eql@numbering@best@row@
2843     \let\eql@tagpos@continuous\eql@false
2844     \eql@tagpos@prevrow@ \z@
2845   \fi
2846 }

```

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

```

2847 \def\eql@numbering@best@test#1{%
2848   \eql@dimensions@get#1%
2849   \eql@dimensions@calc
2850   \ifdefined\eql@tagsleft
2851     \dimen@ \dimexpr \eql@line@avail@
2852       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
2853   \else
2854     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
2855       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
2856   \fi
2857   \ifdim \dimen@ < \eql@tagwidth@block@
2858     \let\eql@numbering@best@use\eql@true
2859   \fi
2860 }

```

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

```

2861 \def\eql@numbering@best@eval{%
2862   \ifdefined\eql@numbering@best@auto
2863     \ifdefined\eql@numbering@best@use\else
2864       \ifdefined\eql@numbering@multi\else
2865         \ifnum \eql@tagpos@row@ > \z@
2866           \eql@numbering@best@test\eql@tagpos@row@
2867         \fi
2868         \ifnum \eql@tagpos@prevrow@ > \z@
2869           \eql@numbering@best@test\eql@tagpos@prevrow@
2870         \fi
2871       \fi
2872     \fi
2873   \fi

```

```

2874 \ifdefined\eql@numbering@best@use
2875   \eql@numbering@best@find
2876 \fi
2877 }

```

## 9.5 Tag Margin

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

djust@calc@tagmargin

```

2878 \def\eql@adjust@calc@tagmargin{%
2879   \ifdefined\eql@tagmargin@val
2880     \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
2881   \else
2882     \eql@tagmargin@\eql@tagwidth@max@
2883     \ifdim\eql@tagmargin@>\z@
2884       \advance\eql@tagmargin@-\eql@tagsepmin@
2885     \fi
2886   \fi

2887   \dimen@\eql@tagrows@\p@
2888   \ifnum\eql@totalrows@=\@ne
2889     \ifnum\eql@tagrows@=\@ne
2890       \advance\dimen@1sp\relax
2891     \fi
2892   \fi
2893   \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
2894     \eql@tagmargin@\z@
2895   \fi

2896   \@tempdima\dimexpr\displaywidth
2897     -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
2898   \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
2899   \ifdim\@tempdimb>\z@
2900     \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
2901       \eql@tagmargin@\z@
2902     \fi
2903   \fi
2904 }

```

## 9.6 Single Column

ql@adjust@calc@lines

```

2905 \def\eql@adjust@calc@lines{%
2906   \eql@totalcolumns@\@ne
2907   \eql@intercolumns@\z@
2908   \eql@colsep@\z@
2909   \ifdefined\eql@layoutleft
2910     \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2911     \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2912     \ifdim\eql@marginleft@<\eql@marginleft@min@
2913       \eql@marginleft@\eql@marginleft@min@
2914     \fi
2915     \dimen@\glueexpr\eql@layoutleftmarginmax\relax

```

```

2916 \ifdim\eql@marginleft@>\dimen@
2917 \eql@marginleft@\dimen@
2918 \fi
2919 \eql@marginright@\z@
2920 \eql@centeroffset@\z@
2921 \else
2922 \eql@adjust@calc@tagmargin
2923 \ifdefined\eql@paddingleft@val
2924 \eql@marginleft@\dimexpr
2925 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2926 -\glueexpr\eql@paddingleft@val\relax\relax
2927 \ifdim\eql@marginleft@<\z@
2928 \eql@marginleft@\z@
2929 \fi
2930 \else
2931 \eql@marginleft@\z@
2932 \fi
2933 \ifdefined\eql@paddingright@val
2934 \eql@marginright@\dimexpr
2935 (\displaywidth-\eql@totalwidth@-\eql@tagmargin@)/\tw@
2936 -\glueexpr\eql@paddingright@val\relax\relax
2937 \ifdim\eql@marginright@<\z@
2938 \eql@marginright@\z@
2939 \fi
2940 \else
2941 \eql@marginright@\z@
2942 \fi
2943 \ifdim\eql@tagmargin@>\z@
2944 \ifdefined\eql@tagsleft
2945 \ifdim\eql@marginleft@<\eql@tagsepmin@
2946 \eql@marginleft@\eql@tagsepmin@
2947 \fi
2948 \advance\eql@marginleft@\eql@tagmargin@
2949 \advance\eql@centeroffset@\eql@tagmargin@
2950 \else
2951 \ifdim\eql@marginright@<\eql@tagsepmin@
2952 \eql@marginright@\eql@tagsepmin@
2953 \fi
2954 \advance\eql@marginright@\eql@tagmargin@
2955 \advance\eql@centeroffset@-\eql@tagmargin@
2956 \fi
2957 \fi
2958 \eql@marginleft@min@\z@
2959 \eql@centeroffset@\dimexpr\eql@marginright@-\eql@marginleft@
2960 \ifdefined\eql@tagsleft+ \else - \fi \eql@tagmargin@ \relax
2961 \fi

2962 \eql@totalwidth@\dimexpr\displaywidth
2963 -\eql@marginleft@-\eql@marginright@ \relax
2964 }

```

## 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:

```
2965 \def\eql@adjust@calc@columns{%
```

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

```
2966 \ifnum\eql@totalcolumns@<\thr@@
2967 \eql@totalcolumns@\tw@
2968 \let\eql@columns@fulllength\eql@false
2969 \fi
```

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

```
2970 \eql@intercolumns@\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
2971 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

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

```
2972 \ifdefined\eql@layoutleft
2973 \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
2974 \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
2975 \ifdim\eql@marginleft@<\eql@marginleft@min@
2976 \eql@marginleft@\eql@marginleft@min@
2977 \fi
2978 \else
```

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

```
2979 \eql@adjust@calc@tagmargin
2980 \ifdefined\eql@columns@fulllength
2981 \ifdim\eql@tagmargin@>\z@
2982 \advance\eql@tagmargin@\eql@tagsepmin@
2983 \fi
2984 \fi
2985 \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
2986 -\eql@intercolumns@\eql@colsepmin@\relax
2987 \eql@tagmargin@\z@
2988 \fi
2989 \eql@marginleft@min@\z@
2990 \fi
```

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

```
2991 \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:

```
2992 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
2993 \ifdefined\eql@layoutleft
2994 \advance\eql@colsep@-\eql@marginleft@
2995 \else
2996 \advance\eql@colsep@-\eql@tagmargin@
2997 \fi
2998 \count@\eql@intercolumns@
2999 \ifdefined\eql@columns@fulllength\else
```

```

3000     \ifdefined\eql@layoutleft
3001       \advance\count@\@ne
3002     \else
3003       \advance\count@\tw@
3004     \fi
3005   \fi
3006   \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:

```

3007   \ifdim\eql@colsep@<\eql@colsepmin@
3008     \eql@colsep@\eql@colsepmin@
3009   \else
3010     \ifdefined\eql@columns@fulllength\else
3011       \dimen@\glueexpr\eql@colsepmax@val\relax
3012       \ifdim\eql@colsep@>\dimen@
3013         \eql@colsep@\dimen@
3014       \fi
3015     \fi
3016   \fi
3017 \else

```

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

```

3018   \eql@colsep@\eql@colsepmin@
3019 \fi

```

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

```

3020 \ifdefined\eql@layoutleft

```

Set the default value:

```

3021   \ifdim\eql@colsep@=\eql@colsepmin@

```

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

```

3022     \dimen@\dimexpr\displaywidth-\eql@totalwidth@
3023     -\eql@intercolumns@\eql@colsep@\relax
3024   \ifdim\dimen@<\eql@marginleft@
3025     \ifdim\dimen@<\eql@marginleft@min@
3026       \eql@marginleft@\eql@marginleft@min@
3027     \else
3028       \eql@marginleft@\dimen@
3029     \fi
3030   \fi
3031 \fi
3032 \else

```

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

```

3033   \ifdefined\eql@columns@fulllength
3034     \eql@marginleft@\z@

```

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

```

3035   \else
3036     \eql@marginleft@\dimexpr(\displaywidth-\eql@totalwidth@

```

```

3037         -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@\relax
3038     \ifdim\eql@marginleft@<\z@
3039         \eql@marginleft@\z@
3040     \fi
3041 \fi

```

Add tag margin in case of left tags:

```

3042     \ifdefined\eql@tagsleft
3043         \advance\eql@marginleft@\eql@tagmargin@
3044     \fi
3045 \fi

```

Find the best row for tag placement:

```

3046 \eql@numbering@best@eval

```

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

```

3047 \ifdefined\eql@tagsleft
3048     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3049 \else
3050     \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3051 \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

```

3052 \ifdefined\eql@numbering@multi
3053     \eql@dimensions@for{%
3054         \ifdim\eql@tagwidth@>\z@
3055             \eql@dimensions@calc
3056             \eql@adjust@columns@test
3057         \fi
3058     }%
3059 \else
3060     \ifnum\eql@tagpos@row@>\z@
3061         \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3062             \eql@dimensions@get\eql@tagpos@row@
3063             \eql@tagwidth@\eql@tagwidth@block@
3064             \eql@dimensions@calc
3065             \eql@adjust@columns@test
3066         \fi
3067     \fi
3068     \ifnum\eql@tagpos@prevrow@>\z@
3069         \eql@dimensions@get\eql@tagpos@prevrow@
3070         \eql@tagwidth@\eql@tagwidth@block@
3071         \eql@dimensions@calc
3072         \eql@adjust@columns@test
3073     \fi
3074 \fi

```

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

```

3075 \advance\eql@totalwidth@\dimexpr
3076     \eql@intercolumns@\eql@colsep@+\eql@marginleft@\relax
3077 }

```

## Placement for Right Tags.

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

```
3078 \def\eql@adjust@columns@test@tagsright{%
```

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

```
3079 \@tempdima\dimexpr\displaywidth-\eql@line@width@-\eql@tagwidth@\relax
```

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

```
3080 \ifdim\@tempdima<\dimexpr
3081 \eql@marginleft@+\eql@line@widthsep@\eql@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).

```
3082 \ifdim\@tempdima<\dimexpr
3083 \eql@marginleft@min@+\eql@line@widthsep@\eql@colsepmin@\relax\else
```

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

```
3084 \eql@adjust@columns@modify@tagsright
3085 \fi
3086 \fi
3087 }
```

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

```
3088 \def\eql@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.

```
3089 \ifnum\eql@line@widthsep@>\z@
3090 \dimen@=\@tempdima
3091 \count@=\eql@line@widthsep@
3092 \ifdefined\eql@layoutleft
3093 \advance\dimen@-\eql@marginleft@
3094 \else
3095 \ifdefined\eql@columns@fulllength\else
3096 \advance\count@\@ne
3097 \fi
3098 \fi
3099 \divide\dimen@\count@
```

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

```
3100 \ifdim\dimen@<\eql@colsep@
3101 \ifdim\dimen@<\eql@colsepmin@
3102 \eql@colsep@\eql@colsepmin@
3103 \else
3104 \eql@colsep@\dimen@
3105 \fi
3106 \fi
3107 \fi
```

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

```

3108 \dimen@ \dimexpr \@tempdima - \eq@line@widthsep@ \eq@colsep@ \relax
3109 \ifdim \eq@marginleft@ > \dimen@
3110 \eq@marginleft@ \dimen@
3111 \fi
3112 }

```

### Placement for Left Tags.

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

```

3113 \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.

```

3114 \count@ \numexpr \eq@intercolumns@ - \eq@line@availsep@ \relax
3115 \@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.

```

3116 \ifdim \@tempdima > \dimexpr
3117 \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).

```

3118 \ifdim \eq@tagwidth@ < %
3119 \ifdefined \eq@layoutleft
3120 \glueexpr \eq@layoutleftmarginmax \relax
3121 \else
3122 \displaywidth
3123 \fi

```

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

```

3124 \ifdim \@tempdima > \dimexpr
3125 \displaywidth - \eq@totalwidth@ - \count@ \eq@colsepmin@ \relax \else

```

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

```

3126 \eq@adjust@columns@modify@tagsleft
3127 \fi
3128 \fi
3129 \fi
3130 }

```

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

```

3131 \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.

```

3132 \ifnum \count@ > \z@

```



```

3133 \dimen@ \dimexpr \displaywidth - \eq@totalwidth - \@tempdima \relax
3134 \ifdefined \eq@columns@fulllength \else
3135 \advance \count@ \@ne
3136 \fi
3137 \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.

```

3138 \ifdim \dimen@ < \eq@colsep@
3139 \ifdim \dimen@ < \eq@colsepmin@
3140 \dimen@ \eq@colsepmin@
3141 \fi
3142 \advance \dimen@ - \eq@colsep@
3143 \advance \eq@marginleft@ - \eq@intercolumns@ \dimen@
3144 \advance \eq@colsep@ \dimen@
3145 \fi
3146 \fi

```

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

```

3147 \dimen@ \dimexpr \@tempdima - \eq@line@availsep@ \eq@colsep@ \relax
3148 \ifdim \eq@marginleft@ < \dimen@
3149 \eq@marginleft@ \dimen@
3150 \fi
3151 }

```

## 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:

```

3152 \ifdefined \inf@bad \else%
3153 \newcount \inf@bad
3154 \inf@bad 1000000 \relax
3155 \fi

```

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

```

3156 \let \eq@restore@hfuzz \empty
3157 \def \eq@save@hfuzz { \edef \eq@restore@hfuzz { \hfuzz \the \hfuzz \relax } }

```

`\eq@alignbadness@` The registers `\eq@alignbadness@` and `\eq@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:

```

3158 \newcount \eq@alignbadness@
3159 \newcount \eq@tagbadness@
3160 \newcount \eq@arrange@badness@
3161 \eq@alignbadness@ \inf@bad
3162 \eq@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@`:

```
3163 \def\eql@arrange@try#1{%
3164   \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
3165     \setbox\eql@cellbox@\hbox to\displaywidth{%
3166       \unhbox\eql@cellbox@\unkern\kern#1}%
3167     \eql@arrange@badness@\badness
3168   \else
3169     \eql@arrange@badness@\m@ne
3170   \fi
3171 }
```

`\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

```
3172 \def\eql@arrange@print#1#2{%
3173   \eql@restore@hfuzz
3174   \if@eqnsw
3175     \ifdefined\eql@tagsleft
3176       \eql@tagbox@print@tagsleft
3177     \fi
3178   \fi
3179   \hbox to\displaywidth{%
3180     #1%
3181     \unhbox\eql@cellbox@\unkern
3182     #2%
3183     \eql@tagging@mathaddlast
3184   }%
3185   \if@eqnsw
3186     \ifdefined\eql@tagsleft\else
3187       \eql@tagbox@print@tagsright
3188     \fi
3189   \fi
3190 }
```

`\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:

```
3191 \def\eql@arrange@print@alignleft#1#2{%
3192   \eql@display@firstavail@set{\dimexpr#1\relax}%
3193   \eql@arrange@print{\kern#1}{\kern#2}%
3194 }

3195 \def\eql@arrange@print@alignright#1#2{%
3196   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3197   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3198 }

3199 \def\eql@arrange@print@aligncenter#1{%
3200   \eql@display@firstavail@set{\dimexpr
3201     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3202   \ifdim#1>\z@
3203     \eql@arrange@print{\kern#1\hfil}{}%
3204   \fi
3205 }
```

```

3204 \else
3205   \eql@arrange@print{\hfil}{\kern-#1}%
3206 \fi
3207 }

```

`\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:

```

3208 \def\eql@arrange@init{%
3209   \eql@save@hfuzz
3210   \hfuzz\maxdimen
3211   \eql@shape@select
3212 }

```

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

```

3213 \def\eql@arrange@print@line{%
3214   \eql@tagging@tagaddbox
3215   \csname eql@arrange%
3216     @ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3217     @init\endcsname
3218   \csname eql@arrange%
3219     @ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3220     @ifdefined\eql@tagpos@reserve
3221     \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3222     notag\fi\endcsname
3223 }

```

### 10.3 Central Alignment

**TODO:** describe

```

3224 \def\eql@arrange@aligncenter@init{%
3225   \eql@tagging@aligncenter
3226   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3227   +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3228 }

```

**TODO:** describe

```

3229 \def\eql@arrange@aligncenter@notag{%
3230   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3231     \ifdim\eql@line@offset@<\eql@marginleft@min@
3232       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3233     \else
3234       \eql@line@offset@
3235     \fi
3236   \eql@arrange@print@aligncenter\eql@line@offset@
3237 \else
3238   \ifdim\eql@line@offset@<\eql@marginleft@min@
3239     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3240   \else
3241     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3242   \fi
3243 \fi
3244 }

```

**TODO:** describe

```

3245 \def\eq@arrange@aligncenter@tagsright{%
3246   \ifdim\dimexpr\displaywidth-\eq@cellwidth@>\relax%
3247     \ifdim\eq@line@offset@<\dimexpr\eq@marginleft@min@-\eq@tagwidth@>\relax
3248       \dimexpr\tw@\eq@marginleft@min@-\eq@line@offset@>\relax
3249     \else
3250       \dimexpr\tw@\eq@tagwidth@+\eq@line@offset@>\relax
3251     \fi
3252   \eq@arrange@print@aligncenter\eq@line@offset@
3253 \else
3254   \eq@arrange@try{\dimexpr\eq@tagwidth@+\eq@marginleft@min@>\relax}%
3255   \ifnum\eq@arrange@badness@<\eq@tagbadness@
3256     \ifdim\eq@line@offset@<\dimexpr\eq@marginleft@min@-\eq@tagwidth@>\relax
3257       \eq@arrange@print@alignleft\eq@marginleft@min@\eq@tagwidth@
3258     \else
3259       \eq@arrange@print@alignright\eq@marginleft@min@\eq@tagwidth@
3260     \fi
3261   \else
3262     \let\eq@tagpos@reserve\eq@false
3263     \eq@arrange@aligncenter@notag
3264   \fi
3265 \fi
3266 }

3267 \def\eq@arrange@aligncenter@tagsleft{%
3268   \ifdim\eq@tagwidth@>\eq@marginleft@min@
3269     \ifdim\dimexpr\displaywidth-\eq@cellwidth@>\relax%
3270       \ifdim\eq@line@offset@<\eq@tagwidth@
3271         \dimexpr\tw@\eq@tagwidth@-\eq@line@offset@>\relax
3272       \else
3273         \eq@line@offset@
3274       \fi
3275     \eq@arrange@print@aligncenter\eq@line@offset@
3276   \else
3277     \eq@arrange@try\eq@tagwidth@
3278     \ifnum\eq@arrange@badness@<\eq@tagbadness@
3279       \ifdim\eq@line@offset@<\eq@tagwidth@
3280         \eq@arrange@print@alignleft\eq@tagwidth@\z@
3281       \else
3282         \eq@arrange@print@alignright\eq@tagwidth@\z@
3283       \fi
3284     \else
3285       \let\eq@tagpos@reserve\eq@false
3286       \eq@arrange@aligncenter@notag
3287     \fi
3288   \fi
3289 \else
3290   \eq@arrange@aligncenter@notag
3291 \fi
3292 }

```

## 10.4 Left Alignment

```

3293 \def\eq@arrange@alignleft@init{%
3294   \eq@tagging@alignleft
3295   \eq@line@offset@\dimexpr\eq@marginleft@+\eq@shape@amount@>\relax
3296   \ifdim\eq@line@offset@<\eq@marginleft@min@
3297     \eq@line@offset@\eq@marginleft@min@

```

```

3298 \fi
3299 }

3300 \def\eql@arrange@alignleft@notag{%
3301 \ifdim\eql@line@offset@>\eql@marginleft@min@
3302 \eql@arrange@try\eql@line@offset@
3303 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3304 \eql@arrange@print@alignleft\eql@line@offset@\z@
3305 \else
3306 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3307 \fi
3308 \else
3309 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3310 \fi
3311 }

3312 \def\eql@arrange@alignleft@tagsright{%
3313 \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3314 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3315 \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3316 \else
3317 \ifdim\eql@line@offset@>\eql@marginleft@min@
3318 \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3319 \fi
3320 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3321 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3322 \else
3323 \let\eql@tagpos@reserve\eql@false
3324 \eql@arrange@alignleft@notag
3325 \fi
3326 \fi
3327 }

3328 \def\eql@arrange@alignleft@tagsleft{%
3329 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3330 \ifdim\eql@line@offset@>\eql@tagwidth@
3331 \eql@arrange@try\eql@line@offset@
3332 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3333 \eql@arrange@print@alignleft\eql@line@offset@\z@
3334 \else
3335 \eql@arrange@try\eql@tagwidth@
3336 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3337 \eql@arrange@print@alignright\eql@tagwidth@\z@
3338 \else
3339 \let\eql@tagpos@reserve\eql@false
3340 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3341 \fi
3342 \fi
3343 \else
3344 \eql@arrange@try\eql@tagwidth@
3345 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3346 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3347 \else
3348 \let\eql@tagpos@reserve\eql@false
3349 \eql@arrange@alignleft@notag
3350 \fi
3351 \fi
3352 \else
3353 \eql@arrange@alignleft@notag
3354 \fi

```

3355 }

## 10.5 Right Alignment

```
3356 \def\eql@arrange@alignright@init{%
3357   \eql@tagging@alignright
3358   \eql@line@offset@dimexpr\eql@marginright@-\eql@shape@amount@\relax
3359   \ifdim\eql@line@offset@<\z@
3360     \eql@line@offset@\z@
3361   \fi
3362 }
```

**TODO:** describe

```
3363 \def\eql@arrange@alignright@notag{%
3364   \ifdim\eql@line@offset@>\z@
3365     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3366     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3367       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3368     \else
3369       \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3370     \fi
3371   \else
3372     \eql@arrange@print@alignright\eql@marginleft@min@\z@
3373   \fi
3374 }
```

**TODO:** describe

```
3375 \def\eql@arrange@alignright@tagsright{%
3376   \ifdim\eql@line@offset@>\eql@tagwidth@
3377     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@line@offset@\relax}%
3378     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3379       \eql@arrange@print@alignright\eql@marginleft@min@\eql@line@offset@
3380     \else
3381       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3382       \ifnum\eql@arrange@badness@<\eql@tagbadness@
3383         \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3384       \else
3385         \let\eql@tagpos@reserve\eql@false
3386         \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3387       \fi
3388     \fi
3389   \else
3390     \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@\relax}%
3391     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3392       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3393     \else
3394       \let\eql@tagpos@reserve\eql@false
3395       \eql@arrange@alignright@notag
3396     \fi
3397   \fi
3398 }
```

**TODO:** describe

```
3399 \def\eql@arrange@alignright@tagsleft{%
3400   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3401     \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3402     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3403       \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
```

```

3404 \else
3405 \ifdim\eql@line@offset@>\z@
3406 \eql@arrange@try\eql@tagwidth@
3407 \fi
3408 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3409 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3410 \else
3411 \let\eql@tagpos@reserve\eql@false
3412 \eql@arrange@alignright@notag
3413 \fi
3414 \fi
3415 \else
3416 \eql@arrange@alignright@notag
3417 \fi
3418 }

```

## 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`

```

3419 \def\eql@box@cr{%
3420 \ifmmode\else\unskip\fi
3421 \eql@vspaceskip@\z@skip
3422 \let\eql@punct@termcr\eql@false
3423 \eql@amprotect\eql@box@cr@testall\eql@box@cr@process}

```

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

`\eql@box@cr@parse`

```

3424 \def\eql@box@cr@testall{\eql@parseopt@aux\eql@box@cr@parse}
3425 \def\eql@box@cr@parse{%
3426 \ifx\eql@parseopt@token[%
3427 \let\eql@parseopt@next\eql@parseopt@vspace
3428 \fi
3429 \ifx\eql@parseopt@token*%
3430 \let\eql@parseopt@next\eql@parseopt@gobble
3431 \fi
3432 \ifx\eql@parseopt@token.%
3433 \let\eql@parseopt@next\eql@parseopt@punctpass
3434 \fi
3435 \ifx\eql@parseopt@token,%
3436 \let\eql@parseopt@next\eql@parseopt@punctpass
3437 \fi
3438 \ifx\eql@parseopt@token~%
3439 \let\eql@parseopt@next\eql@parseopt@punctpass
3440 \fi
3441 \ifx\eql@parseopt@token'
3442 \let\eql@parseopt@next\eql@parseopt@punctnext

```

```

3443 \fi
3444 \ifx\eql@parseopt@token!%
3445   \let\eql@parseopt@next\eql@parseopt@puncttermcr
3446 \fi
3447 \ifx\eql@parseopt@token&%
3448   \let\eql@parseopt@next\eql@parseopt@end
3449 \fi
3450 }

```

`\eql@box@cr@process`

```

3451 \def\eql@box@cr@process{%
3452   \ifdefined\eql@punct@term@cr\eql@punct@apply@top\fi
3453   \eql@box@endline
3454   \expandafter\eql@box@cr@vskip\expandafter{\the\eql@vspaceskip@}%
3455 }

```

`\eql@box@endline`

```

3456 \def\eql@box@endline{%
3457   \eql@punct@apply@line
3458   \eql@hook@lineout
3459 }

```

`\eql@box@cr@vskip`

```

3460 \def\eql@box@cr@vskip#1{%
3461   \eql@box@lastcell
3462   \cr
3463   \noalign{%
3464     \vskip#1\relax
3465   }%
3466 }

```

## 11.2 Stacked Mode

```

3467 \def\eql@box@lastcell@stacked{&\omit\kern-2\eql@colsep@}

```

**TODO:** templates

```

3468 \def\eql@box@open@stacked{%
3469   \eql@shape@align@enable
3470   \let\eql@box@lastcell\eql@box@lastcell@stacked
3471   \everycr{\noalign{%
3472     \eql@verbose@info\eql@verbose@msg@startline
3473     \global\advance\eql@row@\@ne
3474   }}%
3475   \tabskip\z@skip
3476   \halign\bgroup
3477     &%
3478     \global\let\eql@cell@container\@empty
3479     \setbox\eql@cellbox@\hbox{%
3480       \eql@strut@cell
3481       \@lign
3482       $\m@th\eql@mathstyle
3483       \eql@hook@colin
3484       ##%
3485       \eql@punct@apply@col
3486       \eql@hook@colout

```



```

3487         \eql@tagging@mathsave
3488     $%
3489     \eql@tagging@mathaddlast
3490 }%
3491 \ifdefined\eql@shape@lastrow
3492     \eql@totalrows@\eql@row@
3493 \fi
3494 \eql@shape@eval
3495 \eql@cell@container
3496 \ifdefined\eql@frame@cmd
3497     \ifcase\eql@shape@pos@
3498         \eql@frame@measure
3499         \advance\eql@shape@amount@-\eql@frame@margin@
3500     \or\or
3501         \eql@frame@measure
3502         \advance\eql@shape@amount@+\eql@frame@margin@
3503     \fi
3504     \eql@frame@print
3505 \fi
3506 \ifcase\eql@shape@pos@
3507     \kern\eql@shape@amount@
3508     \box\eql@cellbox@
3509     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3510     -\eql@shape@amount@+\@flushglue\relax
3511     \eql@tagging@alignleft
3512 \or
3513     \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3514     \box\eql@cellbox@
3515     \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3516     \eql@tagging@aligncenter
3517 \or
3518     \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3519     +\eql@shape@amount@+\@flushglue\relax
3520     \box\eql@cellbox@
3521     \kern-\eql@shape@amount@
3522     \eql@tagging@alignright
3523 \fi
3524 \tabskip\eql@colsep@\relax
3525 \crrc
3526 \noalign{%
3527     \global\let\eql@shape@lastrow\eql@false
3528     \eql@hook@blockbefore
3529 }%
3530 \eql@hook@blockin
3531 }
3532 \def\eql@mode@stacked{\let\eql@box@open\eql@box@open@stacked}

```

### 11.3 Aligned Mode

```

3533 \def\eql@box@lastcell@odd{%
3534     &\omit
3535     \eql@prevwidth@\wd\eql@cellbox@
3536     \let\eql@frame@cmd\eql@frame@prevcmd
3537     \ifdefined\eql@frame@cmd
3538         \eql@frame@measure
3539         \advance\eql@prevwidth@\eql@frame@margin@
3540         \eql@frame@print
3541     \fi

```

```

3542 \kern-\eql@prevwidth@
3543 \unhbox\eql@cellbox@
3544 \hfil
3545 &\omit\kern-\eql@colsep@
3546 }%
3547 \def\eql@box@lastcell@even{&\omit\kern-\eql@colsep@}

3548 \def\eql@verbose@msg@startline@aligned{starting new line}
3549 \def\eql@box@open@aligned{%
3550 % \TODO templates
3551 \eql@shape@align@disable
3552 \let\eql@box@lastcell\@empty
3553 \everycr{\noalign{%
3554 \eql@verbose@info\eql@verbose@msg@startline@aligned
3555 }}%
3556 \tabskip\z@skip
3557 \halign\bgroup
3558 &%
3559 \let\eql@box@lastcell\eql@box@lastcell@odd
3560 \global\let\eql@cell@container\@empty
3561 \global\setbox\eql@cellbox@\hbox{%
3562 \eql@strut@cell
3563 \@lign
3564 $\m@th\eql@mathstyle
3565 \eql@hook@colin
3566 ##%
3567 \eql@punct@apply@next
3568 \eql@class@innerleft
3569 \eql@hook@innerleft
3570 \eql@tagging@mathsave
3571 $%
3572 \eql@tagging@mathaddlast
3573 }%
3574 \eql@cell@container
3575 \hfil
3576 \kern\wd\eql@cellbox@
3577 \ifdefined\eql@frame@cmd
3578 \eql@frame@measure
3579 \kern\eql@frame@margin@
3580 \fi
3581 \global\let\eql@frame@prevcmd\eql@frame@cmd
3582 \tabskip\z@skip
3583 &%
3584 \eql@prevwidth@\wd\eql@cellbox@
3585 \let\eql@box@lastcell\eql@box@lastcell@even
3586 \let\eql@frame@cmd\eql@frame@prevcmd
3587 \global\let\eql@cell@container\@empty
3588 \setbox\eql@cellbox@\hbox{%
3589 \unhbox\eql@cellbox@
3590 \eql@strut@cell
3591 \@lign
3592 $\m@th\eql@mathstyle
3593 \eql@hook@innerright
3594 \eql@class@innerright@sel
3595 ##%
3596 \eql@punct@apply@col
3597 \eql@hook@colout
3598 \eql@tagging@mathsave
3599 $%

```

```

3600     \eql@tagging@mathaddlast
3601 }%
3602 \eql@cell@container
3603 \ifdefined\eql@frame@cmd
3604     \eql@frame@measure
3605     \advance\eql@prevwidth@\eql@frame@margin@
3606     \eql@frame@print
3607 \fi
3608 \kern-\eql@prevwidth@
3609 \unhbox\eql@cellbox@
3610 \hfil
3611 \tabskip\eql@colsep@\relax
3612 \crrr
3613 \noalign{%
3614     \eql@hook@blockbefore
3615 }%
3616 \eql@hook@blockin
3617 }
3618 \def\eql@mode@aligned{\let\eql@box@open\eql@box@open@aligned}

```

## 11.4 Cases Mode

**TODO:** describe

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

```

3619 \def\eql@box@lastcell@cases{&}%
3620 \let\eql@box@cases@condtext\eql@false
3621 \let\eql@box@cases@condintro\@empty
3622 \def\eql@verbose@msg@startline@cases{starting new line}
3623 \def\eql@box@open@cases{%
3624     \eql@shape@align@disable
3625     \let\eql@box@lastcell\@empty
3626     \everycr{\noalign{%
3627         \eql@verbose@info\eql@verbose@msg@startline@cases
3628     }}%
3629     \tabskip\z@skip
3630     \halign\bgroup
3631         \let\eql@box@lastcell\eql@box@lastcell@cases
3632         \global\let\eql@cell@container\@empty
3633         \global\setbox\eql@cellbox@\hbox{%
3634             \eql@strut@cell
3635             \@lign
3636             $\m@th\eql@mathstyle
3637             \eql@hook@colin
3638             ##%
3639             \eql@punct@apply@next
3640             \eql@tagging@mathsave
3641             $%
3642             \eql@tagging@mathaddlast
3643         }%
3644     \eql@cell@container
3645     \unhbox\eql@cellbox@
3646     \hfil
3647     \eql@tagging@alignleft
3648     \tabskip\eql@colsep@\relax

```

```

3649    &%
3650    \let\eql@box@lastcell\@empty
3651    \global\let\eql@cell@container\@empty
3652    \setbox\eql@cellbox@\hbox{%
3653        \unhbox\eql@cellbox@
3654        \eql@strut@cell
3655        \@lign
3656        $\m@th\eql@mathstyle
3657        \ifdefined\eql@box@cases@condtext
3658            \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
3659        \eql@box@cases@condintro
3660        ##%
3661        \eql@punct@apply@col
3662        \egroup
3663        \eql@hook@colout
3664        \eql@tagging@mathsave
3665        $%
3666        \eql@tagging@mathaddlast
3667    }%
3668    \eql@cell@container
3669    \unhbox\eql@cellbox@
3670    \hfil
3671    \eql@tagging@alignleft
3672    \tabskip\z@skip
3673    \crrr
3674    \noalign{%
3675        \eql@hook@blockbefore
3676    }%
3677    \eql@hook@blockin
3678 }

3679 \def\eql@mode@cases{\let\eql@box@open\eql@box@open@cases}

```

## 11.5 Main

```

3680 \let\eql@box@box\vcenter
3681 \let\eql@box@open\@undefined
3682 \let\eql@box@frame\@firstofone
3683 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

3684 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
3685     \left#1##1\right#2}}
3686 \def\eql@box@getdim{\setbox\@ne\null\ht\@ne\ht\z@\dp\@ne\dp\z@}
3687 \def\eql@box@deldim#1{\hbox{$\m@th\null\delimiterspace\z@\left#1
3688     \ifx\eql@box@box\vcenter\vcenter{\box\@ne}\else\box\@ne\fi\right.$}}
3689 \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
3690     \eql@box@getdim\eql@box@deldim#1##1}}
3691 \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
3692     \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!

```

3693 \def\eql@box@close{%
3694     \ifvmode\else
3695         \ifmmode\else\unskip\fi

```

```

3696 \global\let\eq@shape@lastrow\eq@true
3697 \eq@punct@apply@block
3698 \ifdefined\eq@punct@term@box
3699 \eq@punct@apply@top
3700 \fi
3701 \eq@box@endline
3702 \eq@box@lastcell
3703 \cr
3704 \fi
3705 \noalign{%
3706 \eq@hook@blockafter
3707 \global\let\eq@shape@lastrow\eq@false
3708 }%
3709 \eq@tagging@tablesaveinner
3710 \egroup
3711 }

```

\eq@box@vcenter

```

3712 \def\eq@box@vcenter#1{%
3713 \ifmmode
3714 \vcenter{#1}%
3715 \else
3716 $\m@th\vcenter{#1}$%
3717 \fi
3718 }

```

\eq@box@start

```

3719 \let\eq@box@endmath\eq@false
3720 \def\eq@box@start{%
3721 \relax
3722 \ifmmode
3723 \let\eq@box@endmath\eq@false
3724 \else
3725 \let\eq@box@endmath\eq@true
3726 \expandafter$%$
3727 \fi
3728 \eq@stack@save@box
3729 \let\eq@frame@cmd\@undefined
3730 \let\eq@layoutleft\eq@false
3731 \eq@row@z@
3732 \eq@totalrows@\@M
3733 \eq@shape@select
3734 \setbox\z@\ifx\eq@box@box\vcenter
3735 \expandafter\vbox
3736 \else
3737 \expandafter\eq@box@box
3738 \fi\bgroup
3739 \let\eqnpunct\eq@punct@setnext
3740 \eq@display@nest
3741 \let\\eq@box@cr
3742 \eq@spread@set
3743 \eq@strut@make
3744 \eq@box@open
3745 }

```

\eq@box@end

```

3746 \def\eql@box@end{%
3747   \eql@box@close
3748   \egroup
3749   \eql@box@frame{%
3750     \ifdefined\eql@display@marginleft
3751       \hskip\glueexpr\eql@display@marginleft\relax
3752     \fi
3753     \ifx\eql@box@box\vcenter
3754       \eql@box@vcenter{\unvbox\z@}%
3755     \else
3756       \box\z@
3757     \fi
3758     \eql@tagging@tableaddinner
3759     \ifdefined\eql@display@marginright
3760       \hskip\glueexpr\eql@display@marginright\relax
3761     \fi
3762   }%
3763   \eql@stack@restore
3764   \ifdefined\eql@box@endmath
3765     \expandafter$%$
3766   \fi
3767 }

```

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

```

3768 \def\eql@box@main{%
3769   \expandafter\eql@box@start
3770   \eql@scan@body
3771   \eql@box@end
3772 }

```

## 11.6 Options Processing

```

3773 \def\eql@box@testall{\eql@parseopt@main\eql@box@parse}
3774 \def\eql@box@parse{%
3775   \ifx\eql@parseopt@token[%]
3776     \let\eql@parseopt@next\eql@parseopt@opt
3777   \fi
3778   \ifx\eql@parseopt@token.%
3779     \let\eql@parseopt@next\eql@parseopt@punctpass
3780   \fi
3781   \ifx\eql@parseopt@token,%
3782     \let\eql@parseopt@next\eql@parseopt@punctpass
3783   \fi
3784   \ifx\eql@parseopt@token~%
3785     \let\eql@parseopt@next\eql@parseopt@punctpass
3786   \fi
3787   \ifx\eql@parseopt@token'
3788     \let\eql@parseopt@next\eql@parseopt@punctopt
3789   \fi
3790   \ifx\eql@parseopt@token!%
3791     \let\eql@parseopt@next\eql@parseopt@puncttermopt
3792   \fi
3793   \ifx\eql@parseopt@token=%
3794     \let\eql@parseopt@next\eql@parseopt@lines
3795   \fi
3796   \ifx\eql@parseopt@token|%

```

```

3797 \let\eql@parseopt@next\eql@parseopt@columns
3798 \fi
3799 \ifx\eql@parseopt@token<%
3800 \let\eql@parseopt@next\eql@parseopt@ampeq
3801 \fi
3802 \ifx\eql@parseopt@token>%
3803 \let\eql@parseopt@next\eql@parseopt@eqamp
3804 \fi
3805 }

```

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

```

3806 \def\eql@box@processopt{%
3807 \let\eql@box@frame\@firstofone
3808 \let\eql@display@marginleft\@undefined
3809 \let\eql@display@marginright\@undefined
3810 \let\eql@punct@term@box\eql@false
3811 \let\eql@punct@block\@undefined
3812 \eql@nextopt@process{equationsbox}%
3813 \let\eql@punct@next\@undefined
3814 \eql@colsep@\glueexpr\eql@box@colsep\relax
3815 \ifdefined\eql@paddingleft@val
3816 \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
3817 \else
3818 \eql@paddingleft@z@
3819 \fi
3820 \ifdefined\eql@paddingright@val
3821 \eql@paddingright@\glueexpr\eql@paddingright@val\relax
3822 \else
3823 \eql@paddingright@z@
3824 \fi
3825 \eql@indent@\glueexpr\eql@indent@val\relax
3826 }

```

## 11.7 Environment

`equationsbox` (*env.*)

```

3827 \newenvironment{equationsbox}{%
3828 \eql@verbose@info\eql@verbose@msg@enterenv
3829 \eql@ampprotect\eql@box@testall\eql@box@env@start
3830 }{%
3831 \ifdefined\eql@box@doscan\else
3832 \expandafter\eql@box@end
3833 \fi
3834 \eql@verbose@info\eql@verbose@msg@leaveenv
3835 }

```

`\eql@box@env@start`

```

3836 \def\eql@box@env@start{%
3837 \eql@box@processopt
3838 \ifdefined\eql@box@doscan
3839 \expandafter\eql@scan@env\expandafter\eql@box@main
3840 \else
3841 \expandafter\eql@box@start
3842 \fi
3843 }

```

`\eql@box@ang@open`

```
3844 \def\eql@box@ang@open{%
3845   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
3846   \begin{equationsbox@ang}%
3847   \eql@verbose@info\eql@verbose@msg@enterenv
3848   \let\>\eql@box@ang@close
3849   \eql@ampprotect\eql@box@testall\eql@box@ang@start
3850 }
```

`\eql@box@ang@start` Process arguments and start handling the box:

```
3851 \def\eql@box@ang@start{%
3852   \eql@box@processopt
3853   \ifdefined\eql@box@doscan
3854     \expandafter\eql@scan@ang\expandafter\eql@box@main
3855   \else
3856     \expandafter\eql@box@start
3857   \fi
3858 }
```

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

```
3859 \def\eql@box@ang@close{%
3860   \eql@ampprotect\eql@end@testall\eql@box@ang@end}
```

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

```
3861 \def\eql@box@ang@end{%
3862   \ifdefined\eql@box@doscan\else
3863     \expandafter\eql@box@end
3864   \fi
3865   \eql@verbose@info\eql@verbose@msg@leaveenv
3866   \end{equationsbox@ang}%
3867   \ignorespaces
3868 }
```

## 12 Single-Line Equation

**TODO:** describe

### 12.1 Native Mode

```
3869 \def\eql@single@start@native{%
3870   \eql@display@init
3871   \eql@display@print
3872   \let\raisetag\eql@raisetag@default
3873   \eql@shape@align@disable
3874   \eql@hook@eqin
3875   % \mathopen{}%
3876 }
```

**TODO:** describe

```
3877 \def\eql@single@end@native{%
3878   % \mathclose{}%
3879   \eql@tags@container
3880   \eql@numbering@single@eval
```



```

3881 \if@eqnsw
3882   \ifdefined\eql@tagsleft
3883     \leqno
3884   \else
3885     \eqno
3886   \fi
3887   \eql@composetag@print
3888 \fi
3889 \eql@interline@container
3890 \advance\eql@belowspace@\eql@vspaceskip@
3891 \eql@display@container
3892 \eql@display@penalty
3893 \eql@display@vspace@native
3894 }%

```

## 12.2 Print

```

3895 \def\eql@single@start@print{%
3896   \eql@display@init
3897   \eql@display@print
3898   \eql@shape@align@enable
3899   \eql@totalrows@\@ne
3900   \eql@row@\@ne
3901   \eql@arrange@init
3902   \global\let\eql@cell@container\@empty
3903   \prevgraf\numexpr\prevgraf+\@ne\relax
3904   \setbox\eql@cellbox@\hbox\bgroup
3905     \eql@restore@hfuzz
3906     \eql@strut@cell
3907     $\m@th\eql@mathstyle%$
3908     \eql@hook@eqin
3909 }
3910 \def\eql@single@end@print{%
3911   \eql@tagging@mathsave
3912   $%$
3913   \hfil
3914   \kern\z@
3915   \egroup
3916   \prevgraf\numexpr\prevgraf-\@ne\relax
3917   \eql@shape@eval
3918   \eql@cell@container
3919   \ifdefined\eql@frame@cmd
3920     \eql@frame@adjust
3921   \fi
3922   \eql@cellwidth@\wd\eql@cellbox@
3923   \eql@line@height@\ht\eql@cellbox@
3924   \eql@line@depth@\dp\eql@cellbox@
3925   \eql@totalwidth@\eql@cellwidth@
3926   \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
3927   \eql@topheight@\eql@line@height@
3928   \eql@bottomdepth@\eql@line@depth@
3929   \eql@tags@container
3930   \eql@numbering@single@eval
3931   \if@eqnsw
3932     \eql@tagbox@make\eql@composetag@print
3933     \eql@tagrows@\@ne

```

```

3934 \ifdefined\eql@tagpos@reserve\else
3935 \eql@tagwidth@z@
3936 \fi
3937 \eql@tagheight@block@ht\eql@tagbox@
3938 \eql@tagdepth@block@dp\eql@tagbox@
3939 \else
3940 \eql@numbering@warnunused
3941 \eql@tagwidth@z@
3942 \eql@tagrows@z@
3943 \fi
3944 \eql@tagwidth@max@\eql@tagwidth@
3945 \eql@tagpos@single@eval
3946 \eql@tagpos@print@line@eval

3947 \eql@intercolumns@z@
3948 \eql@adjust@calc@lines

3949 \eql@display@halign@init{}%
3950 \halign{##\crr
3951 \noalign{\eql@display@halign@start}%
3952 \eql@arrange@print@line
3953 \cr
3954 \noalign{\eql@display@halign@end}%
3955 \eql@tagging@tablesavelines
3956 }%
3957 \eql@tagpos@print@line@end
3958 \eql@display@close
3959 }

```

## 13 Multi-Line with Single Column

**TODO:** outline sequence of calls

### 13.1 Measure

**TODO:** describe

```

3960 \def\eql@lines@measure@line@begin{%
3961 \eql@verbose@info\eql@verbose@msg@startline
3962 \eql@numbering@measure@line@begin
3963 \eql@hook@linein
3964 }

```

**TODO:** describe

```

3965 \def\eql@lines@measure@line@end{%
3966 \eql@punct@apply@line
3967 \eql@hook@lineout
3968 }

```

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

```

3969 \def\eql@lines@measure@cell{%
3970 \ifdefined\eql@frame@cmd
3971 \ifcase\eql@shape@pos@
3972 \eql@frame@measure
3973 \advance\eql@shape@amount@-\eql@frame@margin@
3974 \or\or

```

```

3975     \eql@frame@measure
3976     \advance\eql@shape@amount@+\eql@frame@margin@
3977     \fi
3978     \eql@frame@print
3979   \fi
3980   \eql@cellwidth@\wd\eql@cellbox@
3981   \eql@line@height@\ht\eql@cellbox@
3982   \eql@line@depth@\dp\eql@cellbox@
3983   \eql@dimensions@startrow
3984   \eql@dimensions@savcell
3985   \kern\eql@cellwidth@
3986 }

```

\eql@lines@measure

```

3987 \def\eql@lines@measure{%
3988   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
3989   \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
3990   \eql@totalrows@\@M
3991   \eql@shape@select

3992   \setbox\z@\vbox{\measuring@true\halign{%
3993     \global\let\eql@cell@container\@empty
3994     \setbox\eql@cellbox@\hbox{%
3995       \eql@strut@cell
3996       \@lign
3997       $\m@th\eql@mathstyle
3998       \eql@hook@colin
3999       ##%
4000       \eql@punct@apply@col
4001       \eql@hook@colout
4002       $%
4003     }%
4004     \ifdefined\eql@shape@lastrow
4005       \eql@totalrows@\eql@row@
4006     \fi
4007     \eql@shape@eval
4008     \eql@cell@container
4009     \eql@lines@measure@cell
4010     \eql@measure@tag
4011     \eql@measure@endrow
4012   \crr

4013   \noalign{%
4014     \global\let\eql@shape@lastrow\eql@false
4015     \eql@hook@blockbefore
4016   }%
4017   \eql@hook@blockin
4018   \eql@scan@body
4019   \ifvmode\else
4020     \global\let\eql@shape@lastrow\eql@true
4021     \eql@punct@apply@block
4022     \eql@hook@blockout
4023     \eql@display@endline
4024     \cr
4025   \fi
4026   \omit
4027   \cr
4028   \noalign{%

```

```

4029     \eql@hook@blockafter
4030     \global\let\eql@shape@lastrow\eql@false
4031   }%
4032 }}%

4033 \eql@measure@close

4034 \setbox\z@\vbox{%
4035   \unvbox\z@
4036   \unpenalty
4037   \global\setbox\@ne\lastbox
4038 }%
4039 \eql@totalwidth@\wd\@ne

4040 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
4041 }

```

## 13.2 Column Placement

**TODO:** describe Find the best row for tag placement:

```

4042 \def\eql@lines@adjust{%
4043   \eql@tagpos@adjust@eval
4044   \eql@adjust@calc@lines
4045   \eql@numbering@best@eval
4046 }

```

## 13.3 Print

**TODO:** describe

mes@print@line@begin

```

4047 \def\eql@lines@print@line@begin{%
4048   \eql@verbose@info\eql@verbose@msg@startline
4049   \eql@numbering@print@line@begin
4050   \eql@hook@linein
4051 }

```

**TODO:** describe

```

4052 \def\eql@lines@print@line@end{%
4053   \eql@punct@apply@line
4054   \eql@hook@lineout
4055 }

```

**TODO:** describe

```

4056 \def\eql@lines@print@line@adjust{%
4057   \ifdefined\eql@frame@cmd
4058     \ifcase\eql@shape@pos@
4059       \eql@frame@measure
4060       \advance\eql@shape@amount@-\eql@frame@margin@
4061     \or\or
4062       \eql@frame@measure
4063       \advance\eql@shape@amount@+\eql@frame@margin@
4064     \fi
4065     \eql@frame@adjust
4066   \fi

```

```

4067 \eql@cellwidth@\wd\eql@cellbox@
4068 \eql@line@height@\ht\eql@cellbox@
4069 \eql@line@depth@\dp\eql@cellbox@
4070 \eql@numbering@print@line@eval
4071 \if@eqnsw
4072   \eql@tagbox@make\eql@composetag@print
4073 \fi
4074 \eql@tagpos@print@line@eval
4075 \eql@arrange@print@line
4076 \eql@tagpos@print@line@end
4077 }

```

**TODO:** describe

```

4078 \def\eql@lines@print{%
4079   \eql@verbose@infoarg\eql@verbose@msg@center\eql@lines@print
4080   \eql@arrange@init
4081   \eql@display@halign@init\eql@lines@print@line@begin
4082   \eql@display@halign@letcr\eql@lines@print@line@end
4083   \tabskip\z@skip

4084   \halign{%
4085     \global\let\eql@cell@container\@empty
4086     \setbox\eql@cellbox@\hbox{%
4087       \eql@restore@hfuzz
4088       \eql@strut@cell
4089       \@lign
4090       $\m@th\eql@mathstyle
4091       \eql@hook@colin
4092       ##%
4093       \eql@punct@apply@col
4094       \eql@hook@colout
4095       \eql@tagging@mathsave
4096       $%
4097       \hfil
4098       \kern\z@
4099     }%
4100     \eql@shape@eval
4101     \eql@cell@container
4102     \eql@lines@print@line@adjust
4103   \crrr

4104   \noalign{%
4105     \eql@display@halign@start
4106     \eql@numbering@print@block@begin
4107     \eql@hook@blockbefore
4108   }%
4109   \eql@hook@blockin
4110   \eql@scan@body
4111   \ifvmode\else
4112     \relax
4113     \eql@punct@apply@block
4114     \eql@hook@blockout
4115     \eql@display@endline
4116     \cr
4117   \fi
4118   \noalign{%
4119     \eql@hook@blockafter
4120     \eql@display@halign@end
4121     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@print

```

```

4122     }%
4123     \eql@tagging@tablesavelines
4124 }%
4125 }

```

## 14 Multi-Line with Multiple Columns

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

### 14.1 Support

**TODO:** describe

```

\eql@columns@add@amp
@columns@completerow
4126 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4127 \def\eql@columns@completerow{%
4128   \count@ \numexpr\eql@totalcolumns@+\@ne-\eql@column@ \relax
4129   \edef\eql@tmp{%
4130     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4131   \eql@tmp
4132 }

4133 \def\eql@columns@overfull{%
4134   \dimen@ \eql@line@width@
4135   \advance \dimen@ -\hfuzz
4136   \ifdim \dimen@ > \displaywidth
4137     \setbox \z@ \hbox to \displaywidth {\hbox to \eql@line@width@ {\hfil}}%
4138     \wd \z@ \z@
4139     \ht \z@ \eql@line@height@
4140     \dp \z@ \eql@line@depth@
4141     \box \z@
4142   \fi
4143 }

```

### 14.2 Transpose

**TODO:** describe

**TODO:** describe

```

4144 \let\eql@transpose@active\eql@false
4145 \def\eql@transpose@end{\eql@transpose@end}
4146 \def\eql@transpose@skip{&\eqnpunct~}
4147 \def\eql@transpose@complete{%
4148   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}

TODO: describe

4149 \def\eql@transpose{%
4150   \eql@totalcolumns@\z@
4151   \eql@totalrows@\z@
4152   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
4153   \eql@scan@reg@{}%
4154   \eql@row@\z@
4155   \eql@transpose@output@row
4156 }

```

**TODO:** describe

```
4157 \def\eql@transpose@save@col#1{%  
4158   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%  
4159     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}
```

**TODO:** describe

```
4160 \def\eql@transpose@scan@col#1\&{%  
4161   \def\eql@tmpa{#1}%  
4162   \ifx\eql@tmpa\eql@transpose@end\else  
4163     \advance\eql@totalcolumns@\@ne  
4164     \eql@row@\z@  
4165     \let\eql@transpose@data@col\@empty  
4166     \eql@transpose@scan@row#1\\eql@transpose@end\\%  
4167     \ifnum\eql@row@>\eql@totalrows@  
4168       \eql@totalrows@\eql@row@  
4169     \fi  
4170     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%  
4171     \expandafter\eql@transpose@scan@col  
4172   \fi  
4173 }
```

**TODO:** describe

```
4174 \def\eql@transpose@append@row#1{%  
4175   \advance\eql@row@\@ne  
4176   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

**TODO:** describe

```
4177 \def\eql@transpose@scan@row#1\\{%  
4178   \def\eql@tmpa{#1}%  
4179   \ifx\eql@tmpa\eql@transpose@end\else  
4180     \ifx\eql@transpose@active+  
4181       \eql@transpose@scan@cell#1&\eql@transpose@end&%  
4182     \else  
4183       \eql@transpose@append@row{#1}%  
4184     \fi  
4185     \expandafter\eql@transpose@scan@row  
4186   \fi  
4187 }
```

**TODO:** describe

```
4188 \def\eql@transpose@scan@cell#1&#2&{%  
4189   \def\eql@tmpa{#2}%  
4190   \ifx\eql@tmpa\eql@transpose@end  
4191     \eql@transpose@append@row{#1}%  
4192   \else  
4193     \eql@transpose@append@row{#1&#2}%  
4194     \expandafter\eql@transpose@scan@cell@next  
4195   \fi  
4196 }
```

**TODO:** describe

```
4197 \def\eql@transpose@scan@cell@next#1&{%  
4198   \def\eql@tmpa{#1}%  
4199   \ifx\eql@tmpa\eql@transpose@end\else  
4200     \eql@transpose@append@row{&#1}%  
4201     \expandafter\eql@transpose@scan@cell@next
```

```

4202 \fi
4203 }

```

**TODO:** describe

```

4204 \def\eql@transpose@output@row{%
4205   \ifnum\eql@row@<\eql@totalrows@
4206     \advance\eql@row@\@ne
4207     \eql@column@\z@
4208     \eql@transpose@output@col
4209     \ifnum\eql@row@<\eql@totalrows@
4210       \eql@scan@addto\\%
4211     \fi
4212     \expandafter\eql@transpose@output@row
4213 \fi
4214 }

```

**TODO:** describe

```

4215 \def\eql@transpose@output@col{%
4216   \ifnum\eql@column@<\eql@totalcolumns@
4217     \advance\eql@column@\@ne
4218     \csname eql@transpose@data@col@\the\eql@column@\endcsname
4219     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4220     \ifnum\eql@column@<\eql@totalcolumns@
4221       \eql@scan@addto{\eql@transpose@complete}%
4222     \fi
4223     \expandafter\eql@transpose@output@col
4224 \fi
4225 }

```

## 14.3 Measure

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

s@measure@line@begin

```

4226 \def\eql@columns@measure@line@begin{%
4227   \eql@verbose@info\eql@verbose@msg@startline
4228   \global\eql@column@\z@
4229   \global\eql@line@height@\z@
4230   \global\eql@line@depth@\z@
4231   \eql@numbering@measure@line@begin
4232   \eql@hook@linein
4233 }

4234 \def\eql@columns@measure@cell{%
4235   \eql@cellwidth@\wd\eql@cellbox@
4236   \ifdefined\eql@frame@cmd
4237     \eql@frame@measure
4238     \advance\eql@cellwidth@\eql@frame@margin@
4239   \fi
4240   \ifdim\ht\eql@cellbox@>\eql@line@height@
4241     \global\eql@line@height@\ht\eql@cellbox@
4242   \fi
4243   \ifdim\dp\eql@cellbox@>\eql@line@depth@
4244     \global\eql@line@depth@\dp\eql@cellbox@
4245   \fi
4246   \ifnum\eql@column@=\@ne

```



```

4247 \eqldimensions@startrow
4248 \fi
4249 \ifodd\eql@column@
4250 \eql@shape@pos@tw@
4251 \else
4252 \eql@shape@pos@z@
4253 \fi
4254 \eql@shape@amount@z@
4255 \eqldimensions@savecell
4256 \ifodd\eql@column@ \else
4257 \eqldimensions@savesep
4258 \fi
4259 \kern\eql@cellwidth@
4260 }

```

columns@measure@line@end

```

4261 \def\eql@columns@measure@line@end{%
4262 \eql@punct@apply@line
4263 \eql@hook@lineout
4264 &\omit
4265 \ifnum\eql@column@>\eql@totalcolumns@
4266 \global\eql@totalcolumns@\eql@column@
4267 \fi

```

**TODO:** not sure whether saving the last cell value makes sense, but rather not increase  $\backslash\text{eql@totalcolumns@}$  because that will disable the fallback to lines mode. **TODO:** additional column in width table is accounted for in column table

```

4268 \ifdefined\eql@frame@cmd
4269 \advance\eql@column@\@ne
4270 \wd\eql@cellbox@z@
4271 \eql@columns@measure@cell
4272 \fi
4273 \eql@measure@tag
4274 \eql@measure@endrow
4275 }

```

$\backslash\text{eql@columns@measure}$

```

4276 \def\eql@columns@measure{%
4277 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4278 \eql@totalcolumns@z@
4279 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end

4280 \setbox\z@\vbox{\measuring@true\halign{%
4281 &%
4282 \global\advance\eql@column@\@ne
4283 \global\let\eql@cell@container\@empty
4284 \global\setbox\eql@cellbox@\hbox{%
4285 \eql@strut@cell
4286 \@lign
4287 $\m@th\eql@mathstyle
4288 \eql@hook@colin
4289 ##%
4290 \eql@punct@apply@next
4291 \eql@class@innerleft
4292 \eql@hook@innerleft
4293 $%

```

```

4294     }%
4295     \eql@cell@container
4296     \hfil
4297     \eql@columns@measure@cell
4298     \global\let\eql@frame@prevcmd\eql@frame@cmd
4299     &%
4300     \eql@prevwidth@\wd\eql@cellbox@
4301     \let\eql@frame@cmd\eql@frame@prevcmd
4302     \global\advance\eql@column@\@ne
4303     \global\let\eql@cell@container\@empty
4304     \setbox\eql@cellbox@\hbox{%
4305         \eql@strut@cell
4306         \@lign
4307         $\m@th\eql@mathstyle
4308         \eql@hook@innerright
4309         \eql@class@innerright@sel
4310         ##%
4311         \eql@punct@apply@col
4312         \eql@hook@colout
4313         $%
4314     }%
4315     \eql@cell@container
4316     \eql@columns@measure@cell
4317     \hfil
4318     \cr
4319     \noalign{%
4320         \eql@hook@blockbefore
4321     }%
4322     \eql@hook@blockin
4323     \eql@scan@body
4324     \ifvmode\else
4325         \eql@punct@apply@block
4326         \eql@hook@blockout
4327         \eql@display@endline
4328         \cr
4329     \fi
4330     \noalign{%
4331         \eql@hook@blockafter
4332     }%

```

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

```

4333     \omit
4334     \eql@column@\@ne
4335     \eql@columns@completerow
4336     \cr
4337     }}%
4338     \eql@measure@close
4339     \setbox\z@\vbox{%
4340         \unvbox\z@
4341         \unpenalty
4342         \global\setbox\@ne\lastbox
4343     }%
4344     \eql@totalwidth@\wd\@ne

```

**TODO:** why not recycle box contents altogether?!

```

4345 \let\eql@colwidth@tab\@empty
4346 \loop
4347   \setbox\@ne\hbox{%
4348     \unhbox\@ne
4349     \unskip
4350     \global\setbox\thr@@\lastbox
4351   }%
4352 \ifhbox\thr@@
4353   \eql@colwidth@save{\wd\thr@@}%
4354 \repeat

4355 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@measure
4356 }

```

## 14.4 Columns Placement

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

```

4357 \def\eql@columns@adjust{%
4358   \ifodd\eql@totalcolumns@
4359     \advance\eql@totalcolumns@\@ne
4360   \fi
4361   \eql@tagpos@adjust@eval
4362   \eql@adjust@calc@columns
4363 }

```

## 14.5 Print

**TODO:** describe

ms@print@line@begin

```

4364 \def\eql@columns@print@line@begin{%
4365   \eql@verbose@info\eql@verbose@msg@startline
4366   \global\eql@column@\z@
4367   \global\eql@line@pos@\eql@marginleft@
4368   \global\eql@line@width@\z@
4369   \global\eql@line@avail@\eql@totalwidth@
4370   \global\eql@line@height@\z@
4371   \global\eql@line@depth@\z@
4372   \eql@numbering@print@line@begin
4373   \eql@hook@linein
4374 }

```

l@columns@print@cell

```

4375 \def\eql@columns@print@cell{%
4376   \eql@cellwidth@\wd\eql@cellbox@
4377   \ifodd\eql@column@
4378     \ifdefined\eql@frame@cmd
4379       \eql@frame@measure
4380       \advance\eql@cellwidth@\eql@frame@margin@
4381     \fi
4382     \dimen@\z@
4383   \else
4384     \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```
4385 \ifdefined\eql@frame@cmd
4386 \eql@frame@measure
4387 \advance\eql@cellwidth@\eql@frame@margin@
4388 \advance\eql@prevwidth@\eql@frame@margin@
4389 \eql@frame@print
4390 \fi
```

update height and depth

```
4391 \ifdim\ht\eql@cellbox@>\eql@line@height@
4392 \global\eql@line@height@\ht\eql@cellbox@
4393 \fi
4394 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4395 \global\eql@line@depth@\dp\eql@cellbox@
4396 \fi
```

print box

```
4397 \kern-\eql@prevwidth@
4398 \unhbox\eql@cellbox@
4399 \dimen@-\eql@cellwidth@
4400 \fi
```

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

```
4401 \advance\dimen@\eql@colwidth@get\eql@column@\relax
4402 \kern\dimen@
```

update available and used space

```
4403 \dimen@\eql@colwidth@get\eql@column@\relax
4404 \ifdim\eql@cellwidth@>\z@
4405 \ifdim\eql@line@width@=\z@
4406 \eql@line@avail@\eql@line@pos@
4407 \ifodd\eql@column@
4408 \advance\eql@line@avail@\dimen@
4409 \advance\eql@line@avail@-\eql@cellwidth@
4410 \fi
4411 \global\eql@line@avail@\eql@line@avail@
4412 \fi
4413 \eql@line@width@\eql@line@pos@
4414 \ifodd\eql@column@
4415 \advance\eql@line@width@\dimen@
4416 \else
4417 \advance\eql@line@width@\eql@cellwidth@
4418 \fi
4419 \global\eql@line@width@\eql@line@width@
4420 \fi
4421 \advance\eql@line@pos@\dimen@
4422 \ifodd\eql@column@\else
4423 \advance\eql@line@pos@\eql@colsep@
4424 \fi
4425 \global\eql@line@pos@\eql@line@pos@
4426 }
```

```
4427 \def\eql@columns@print@trailright{%
4428 &\omit
4429 \eql@prevwidth@\wd\eql@cellbox@
```

```

4430 \let\eql@frame@cmd\eql@frame@prevcmd
4431 \global\advance\eql@column@\@ne
4432 \eql@columns@print@cell
4433 }

```

lums@print@line@end

```

4434 \def\eql@columns@print@line@end{%
4435 \eql@punct@apply@line
4436 \eql@hook@lineout
4437 % \TODO add an even column with empty stuff if box processing deferred
4438 \ifodd\eql@column@
4439 \expandafter\eql@columns@print@trailright
4440 \fi
4441 \eql@columns@completerow
4442 \eql@columns@print@tag
4443 }

```

ql@columns@print@tag

```

4444 \def\eql@columns@print@tag{%
4445 \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4446 \eql@display@firstavail@set\eql@line@avail@
4447 \eql@columns@overfull
4448 \eql@numbering@print@line@eval
4449 \if@eqnsw
4450 \eql@tagbox@make\eql@composetag@print
4451 \fi
4452 \eql@tagpos@print@line@eval
4453 \eql@tagbox@print@cell
4454 \eql@tagpos@print@line@end
4455 }

```

\eql@columns@print

```

4456 \def\eql@columns@print{%
4457 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
4458 \eql@shape@align@disable
4459 \eql@display@halign@init\eql@columns@print@line@begin
4460 \eql@display@halign@letcr\eql@columns@print@line@end
4461 \tabskip\eql@marginleft@

4462 \halign{%
4463 &%
4464 \global\advance\eql@column@\@ne
4465 \global\let\eql@cell@container\@empty
4466 \global\setbox\eql@cellbox@\hbox{%
4467 \eql@strut@cell
4468 \@lign
4469 $\m@th\eql@mathstyle
4470 \eql@hook@colin
4471 ##%
4472 \eql@punct@apply@next
4473 \eql@class@innerleft
4474 \eql@hook@innerleft
4475 \eql@tagging@mathsave
4476 $%

```

```

4477     \eql@tagging@mathaddlast
4478 }%
4479 \eql@cell@container
4480 \hfil
4481 \eql@columns@print@cell
4482 \global\let\eql@frame@prevcmd\eql@frame@cmd
4483 \tabskip\z@skip
4484 &%
4485 \eql@prevwidth@\wd\eql@cellbox@
4486 \let\eql@frame@cmd\eql@frame@prevcmd
4487 \global\advance\eql@column@\@ne
4488 \global\let\eql@cell@container\@empty
4489 \setbox\eql@cellbox@\hbox{%
4490   \unhbox\eql@cellbox@
4491   \eql@strut@cell
4492   \@lign
4493   $\m@th\eql@mathstyle
4494   \eql@hook@innerright
4495   \eql@class@innerright@sel
4496   ##%
4497   \eql@punct@apply@col
4498   \eql@hook@colout
4499   \eql@tagging@mathsave
4500   $%
4501   \eql@tagging@mathaddlast
4502 }%
4503 \eql@cell@container
4504 \eql@columns@print@cell
4505 \hfil
4506 \tabskip\eql@colsep@\relax
4507 \crrc

4508 \noalign{%
4509   \eql@display@halign@start
4510   \eql@numbering@print@block@begin
4511   \eql@hook@blockbefore
4512 }%
4513 \eql@hook@blockin
4514 \eql@scan@body
4515 \ifvmode\else
4516   \relax
4517   \eql@punct@apply@block
4518   \eql@hook@blockout
4519   \eql@display@endline
4520   \cr
4521 \fi
4522 \noalign{%
4523   \eql@hook@blockafter
4524   \eql@display@halign@end
4525   \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
4526 }%
4527 \eql@tagging@tablesalign
4528 }%
4529 }

```

## 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.

```
4530 \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:

```
4531 \def\eql@scan@body@dump{\the\eql@scan@reg@}
4532 \def\eql@scan@body@rescan{%
4533   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
4534 \let\eql@scan@body\eql@scan@body@dump
```

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

```
4535 \long\def\eql@scan@addto#1{\eql@scan@reg@\expandafter{\the\eql@scan@reg@#1}}
```

#### Scan Modifiers at End.

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

```
4536 \def\eql@scan@testend{%
4537   \eql@ampprotect\eql@end@testall\eql@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.

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

```
4538 \def\eql@scan@env#1{%
4539   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@env
4540   \def\eql@scan@end{#1}\expandafter\end\expandafter{\@currenvir}}%
4541   \eql@scan@reg@{}\def\eql@scan@stack{b}%
```

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

```
4542   \edef\eql@scan@iterate{\expandafter\noexpand\csname\@currenvir\endcsname}%
```

```

4543 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
4544 \ifdefined\eql@scan@par
4545   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
4546 \else
4547   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
4548 \fi
4549 \eql@scan@iterate
4550 }

```

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

```

4551 \long\def\eql@scan@env@iterate#1\end#2{%
4552   \edef\eql@scan@stack{%
4553     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
4554   \ifx\@empty\eql@scan@stack
4555     \checkend{#2}%
4556     \eql@scan@addto{#1}%
4557     \expandafter\let\eql@scan@iterate\eql@scan@env@org
4558     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@env
4559     \expandafter\eql@scan@testend
4560   \else
4561     \eql@scan@addto{#1\end{#2}}%
4562     \expandafter\eql@scan@iterate
4563   \fi
4564 }

```

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

```

4565 \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.

```

4566 \long\def\eql@scan@env@count#1\begin#2{%
4567   \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
4568 }

```

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

```

4569 \def\eql@scan@env@cancel{%
4570   \@namedef{end\@currenvir}{\ignorespacesafterend}%
4571 }

```

**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 `\]`:

```

4572 \def\eql@scan@sqr#1{%
4573   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
4574   \def\eql@scan@end{#1}\}%

```



```

4575 \eql@scan@reg@{ }\def\eql@scan@stack{b}%
4576 \let\eql@scan@sqr@org\[%\]
4577 \ifdefined\eql@scan@par
4578   \let\[\eql@scan@sqr@iterate%\]
4579 \else
4580   \let\[\eql@scan@sqr@iterate@nopar%\]
4581 \fi
4582 \[%\]
4583 }

```

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

```

4584 \long\def\eql@scan@sqr@iterate#1\]{%
4585   \edef\eql@scan@stack{%
4586     \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
4587   \ifx\@empty\eql@scan@stack
4588     \let\[\eql@scan@sqr@org%\]
4589     \eql@scan@addto{#1}%
4590     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
4591     \expandafter\eql@scan@testend
4592   \else
4593     \eql@scan@addto{#1\]}%
4594     \expandafter\[%\]
4595   \fi
4596 }

```

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

```

4597 \def\eql@scan@sqr@iterate@nopar#1\]{\eql@scan@sqr@iterate#1\]}

```

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

```

4598 \long\def\eql@scan@sqr@count#1\[#2{%\]
4599   \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
4600 }

```

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

```

4601 \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 `\>`:

```

4602 \def\eql@scan@ang#1{%
4603   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
4604   \def\eql@scan@end{#1\>}%
4605   \eql@scan@reg@{ }\def\eql@scan@stack{b}%
4606   \let\eql@scan@ang@org\<%\>
4607   \ifdefined\eql@scan@par
4608     \let\<\eql@scan@ang@iterate%\>
4609   \else
4610     \let\<\eql@scan@ang@iterate@nopar%\>
4611   \fi
4612   \<%\>
4613 }

```

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

```

4614 \long\def\eql@scan@ang@iterate#1\>{%
4615   \edef\eql@scan@stack{%
4616     \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
4617   \ifx\@empty\eql@scan@stack
4618     \let\<\eql@scan@ang@org%\>
4619     \eql@scan@addto{#1}%
4620     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
4621     \expandafter\eql@scan@testend
4622   \else
4623     \eql@scan@addto{#1\>}%
4624     \expandafter\<%\>
4625   \fi
4626 }
```

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

```

4627 \def\eql@scan@ang@iterate@nopar#1\>{\eql@scan@ang@iterate#1\>}
```

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

```

4628 \long\def\eql@scan@ang@count#1\<#2{%\>
4629   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
4630 }
```

## 15.2 Options Processing

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

```

4631 \def\eqlequations@testall{\eql@parseopt@main\eqlequations@parse}
4632 \def\eqlequations@parse{%
4633   \ifx\eql@parseopt@token*%
4634     \let\eql@parseopt@next\eql@parseopt@nonumber
4635   \fi
4636   \ifx\eql@parseopt@token!%
4637     \let\eql@parseopt@next\eql@parseopt@donumber
4638   \fi
4639   \ifx\eql@parseopt@token/%
4640     \let\eql@parseopt@next\eql@parseopt@transpose
4641   \fi
4642   \ifx\eql@parseopt@token[%]
4643     \let\eql@parseopt@next\eql@parseopt@opt
4644   \fi
4645   \ifx\eql@parseopt@token\eql@atxi
4646     \let\eql@parseopt@next\eql@parseopt@label
4647   \fi
4648   \ifx\eql@parseopt@token\eql@atxii
4649     \let\eql@parseopt@next\eql@parseopt@label
4650   \fi
4651   \ifx\eql@parseopt@token.%
4652     \let\eql@parseopt@next\eql@parseopt@punctpass
4653   \fi
4654   \ifx\eql@parseopt@token,%
4655     \let\eql@parseopt@next\eql@parseopt@punctpass
```

```

4656 \fi
4657 \ifx\eql@parseopt@token~%
4658   \let\eql@parseopt@next\eql@parseopt@punctpass
4659 \fi
4660 \ifx\eql@parseopt@token'%
4661   \let\eql@parseopt@next\eql@parseopt@punctopt
4662 \fi
4663 \ifx\eql@parseopt@token-%
4664   \let\eql@parseopt@next\eql@parseopt@single
4665 \fi
4666 \ifx\eql@parseopt@token=%
4667   \let\eql@parseopt@next\eql@parseopt@lines
4668 \fi
4669 \ifx\eql@parseopt@token|%
4670   \let\eql@parseopt@next\eql@parseopt@columns
4671 \fi
4672 \ifx\eql@parseopt@token<%
4673   \let\eql@parseopt@next\eql@parseopt@ampeq
4674 \fi
4675 \ifx\eql@parseopt@token>%
4676   \let\eql@parseopt@next\eql@parseopt@eqamp
4677 \fi
4678 \ifx\eql@parseopt@token\label
4679   \let\eql@parseopt@next\eql@parseopt@end
4680 \fi
4681 \ifx\eql@parseopt@token\begin
4682   \let\eql@parseopt@next\eql@parseopt@end
4683 \fi
4684 }

```

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

```

4685 \def\eql@end@testall{\eql@parseopt@main\eql@end@parse}
4686 \def\eql@end@parse{%
4687   \ifx\eql@parseopt@token.%
4688     \let\eql@parseopt@next\eql@parseopt@punctpass
4689   \fi
4690   \ifx\eql@parseopt@token,%
4691     \let\eql@parseopt@next\eql@parseopt@punctpass
4692   \fi
4693   \ifx\eql@parseopt@token~%
4694     \let\eql@parseopt@next\eql@parseopt@punctpass
4695   \fi
4696   \ifx\eql@parseopt@token'%
4697     \let\eql@parseopt@next\eql@parseopt@punctblock
4698   \fi
4699   \ifx\eql@parseopt@token!%
4700     \let\eql@parseopt@next\eql@parseopt@puncttermbox
4701   \fi
4702 }

```

**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:

```

4703 \def\eql@equations@processopt{%
4704   \let\eql@tags@container@block\eql@tags@container@clear
4705   \let\eql@tags@frame@cmd\@firstofone

```

```

4706 \let\eql@skip@force@above\@undefined
4707 \let\eql@skip@force@below\@undefined
4708 \let\eql@skip@force@leave\@undefined
4709 \let\eql@display@linewidth\@undefined
4710 \let\eql@display@marginleft\@undefined
4711 \let\eql@display@marginright\@undefined
4712 \eql@abovespace@\z@skip
4713 \eql@belowspace@\z@skip
4714 \eql@displaybreak@prepen@\@MM
4715 \eql@displaybreak@postpen@\@MM
4716 \eql@nextopt@process{equations}%
4717 \let\eql@punct@next\@undefined
4718 \eql@indent@\glueexpr\eql@indent@val\relax
4719 \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
4720 }

```

### 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:

```

4721 \def\eql@single@cr@error{%
4722   \eql@error{Cannot use '\string\\' within display equation.
4723     Please switch to equations environment}}%
4724 }

```

`\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:

```

4725 \def\eql@single@start{%
4726   \eql@display@enter
4727   \eql@tagging@start
4728   \eql@dollar@begin
4729   \eql@display@adjust
4730   \eql@numbering@init
4731   \eql@stack@save@equations
4732   \eql@numbering@single@init
4733   \ifdefined\eql@single@cr@mode
4734     \let\\\eql@single@cr@mode
4735   \fi
4736   \ifdefined\eql@single@native
4737     \let\eql@single@start@sel\eql@single@start@native
4738     \let\eql@single@end@sel\eql@single@end@native
4739   \else
4740     \let\eql@single@start@sel\eql@single@start@print
4741     \let\eql@single@end@sel\eql@single@end@print
4742   \fi
4743   \eql@single@start@sel
4744 }

```

`\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:

```

4745 \def\eql@single@end{%
4746   \eql@punct@apply@block
4747   \eql@hook@eqout
4748   \eql@single@end@sel
4749   \global\eql@punct@top@reset
4750   \eql@stack@restore
4751   \eql@dollar@dollar@end
4752   \eql@tagging@end
4753   \eql@display@leave
4754 }

```

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

```

4755 \def\eql@single@main{%
4756   \expandafter\eql@single@start
4757   \eql@scan@body
4758   \eql@single@end
4759 }

```

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

```

4760 \def\eql@mode@single{%
4761   \ifdefined\eql@single@doscan
4762     \let\eql@equations@main\eql@single@main
4763   \else
4764     \let\eql@equations@main\@undefined
4765   \fi
4766 }

```

## 15.4 Multi-Line Main

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

```

4767 \let\eql@multi@linesmode\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

```

4768 \def\eql@multi@main{%
4769   \eql@display@enter
4770   \eql@tagging@start
4771   \eql@dollar@dollar@begin
4772   \eql@display@adjust
4773   \eql@numbering@init
4774   \eql@stack@save@equations
4775   \ifdefined\eql@transpose@active
4776     \ifdefined\eql@multi@linesmode\else
4777       \eql@transpose
4778     \fi
4779   \fi
4780   \ifdefined\eql@numbering@subeq@use
4781     \eql@numbering@subeq@init
4782   \fi
4783   \eql@display@init
4784   \let\intertext\eql@intertext

```

```

4785 \let\endintertext\endeql@intertext
4786 \eq@shape@align@enable

```

Now measure the given multi-line equations body:

```

4787 \ifdefined\eq@multi@linesmode
4788   \eq@lines@measure
4789 \else
4790   \ifdefined\eq@ampproof@active
4791     \eq@ampproof
4792   \fi
4793   \eq@columns@measure
4794 \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 `\eq@lines@measure`:

```

4795 \ifdefined\eq@numbering@subeq@use
4796   \eq@numbering@subeq@test
4797 \fi
4798 \ifdefined\eq@multi@linesmode\else
4799   \ifdefined\eq@multi@linesfallback
4800     \ifnum\eq@totalcolumns@=\@ne
4801       \let\eq@multi@linesmode\eq@true
4802       \ifx\eq@multi@linesfallback\z@\else
4803         \eq@lines@measure
4804       \fi
4805     \fi
4806   \fi
4807 \fi

```

Adjust the multi-line equations body:

```

4808 \ifdefined\eq@multi@linesmode
4809   \eq@lines@adjust
4810 \else
4811   \eq@columns@adjust
4812 \fi

```

Now print the multi-line equations body:

```

4813 \eq@display@print
4814 \eq@numbering@print@init
4815 \ifdefined\eq@multi@linesmode
4816   \eq@lines@print
4817 \else
4818   \eq@columns@print
4819 \fi
4820 \eq@display@close

```

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

```

4821 \ifdefined\eq@numbering@subeq@use
4822   \eq@numbering@subeq@close
4823 \fi
4824 \global\eq@punct@top@reset
4825 \eq@stack@restore
4826 \eq@dollar@dollar@end

```

```

4827 \eql@tagging@end
4828 \eql@display@leave
4829 }

```

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

```

\eql@mode@lines
4830 \def\eql@mode@columns{%
4831   \let\eql@equations@main\eql@multi@main
4832   \let\eql@multi@linesmode\eql@false
4833 }
4834 \def\eql@mode@lines{%
4835   \let\eql@equations@main\eql@multi@main
4836   \let\eql@multi@linesmode\eql@true
4837 }

```

## 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 `\eql@equations@start`:

```

4838 \newenvironment{equations}{%
4839   \eql@verbose@info\eql@verbose@msg@enterenv
4840   \ifmmode
4841     \eql@error@mathmode{\string\begin{\@currenvir}}%
4842     \expandafter\eql@scan@env\expandafter\eql@scan@env@cancel
4843   \else
4844     \expandafter\eql@ampprotect\expandafter\eql@equations@testall
4845     \expandafter\eql@equations@env@start
4846   \fi
4847 }{%
4848   \ifdefined\eql@equations@main\else
4849     \expandafter\eql@single@end
4850   \fi
4851   \ignorespacesafterend
4852   \eql@verbose@info\eql@verbose@msg@leaveenv
4853 }
4854 \eql@markline@amsthm@register{equations}
4855 \eql@tagging@register@luamml{equations}

```

`@equations@env@start` The macro `\eql@equations@env@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`:

```

4856 \def\eql@equations@env@start{%
4857   \eql@equations@processopt
4858   \ifdefined\eql@equations@main
4859     \expandafter\eql@scan@env\expandafter\eql@equations@main
4860   \else
4861     \expandafter\eql@single@start
4862   \fi
4863 }

```

## Square Brackets.

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

```
4864 \newenvironment{equations@sqr}{}{}
4865 \eq@markline@amsthm@register{equations@sqr}
4866 \eq@tagging@register@luamml{equations@sqr}
```

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

```
4867 \protected\def\eq@equations@sqr@open{%
4868   \ifmmode
4869     \eq@error@mathmode{\string\[...\string\]}%
4870     \expandafter\eq@scan@sqr\expandafter\eq@scan@sqrang@cancel
4871   \else
4872     \expandafter\eqnaddopt\expandafter{\eq@equations@sqr@opt}%
4873     \begin{equations@sqr}%
4874     \eq@verbose@info\eq@verbose@msg@enterenv
4875     \let\]\eq@equations@sqr@close
4876     \expandafter\eq@ampprotect\expandafter\eq@equations@testall
4877     \expandafter\eq@equations@sqr@start
4878   \fi
4879 }
```

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

```
4880 \def\eq@equations@sqr@start{%
4881   \eq@equations@processopt
4882   \ifdefined\eq@equations@main
4883     \expandafter\eq@scan@sqr\expandafter\eq@equations@main
4884   \else
4885     \expandafter\eq@single@start
4886   \fi
4887 }
```

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

```
4888 \protected\def\eq@equations@sqr@close{%
4889   \eq@ampprotect\eq@end@testall\eq@equations@sqr@end}
```

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

```
4890 \def\eq@equations@sqr@end{%
4891   \ifdefined\eq@equations@main\else
4892     \expandafter\eq@single@end
4893   \fi
4894   \eq@verbose@info\eq@verbose@msg@leaveenv
4895   \end{equations@sqr}%
4896   \ignorespaces
4897 }
```

**TODO:** describe

```
\eq@sqr@open
\eq@sqr@close
```



```

4898 \let\eql@sqr@open\eql@equations@sqr@open
4899 \protected\def\eql@sqr@close{%
4900   \eql@error{'\string\}' may only close '\string\[']%
4901 }

```

## Angle Brackets.

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

```

4902 \newenvironment{equations@ang}{}{}
4903 \newenvironment{equationsbox@ang}{}{}
4904 \eql@markline@amsthm@register{equations@ang}
4905 \eql@tagging@register@luamml{equations@ang}

```

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

```

4906 \protected\def\eql@ang@open{%
4907   \ifmmode
4908     \expandafter\eql@box@ang@open
4909   \else
4910     \expandafter\eql@equations@ang@open
4911   \fi
4912 }

```

`\l@equations@ang@open`

```

4913 \def\eql@equations@ang@open{%
4914   \expandafter\eqnaddopt\expandafter{\eql@equations@ang@opt}%
4915   \begin{equations@ang}%
4916   \eql@verbose@info\eql@verbose@msg@enterenv
4917   \let\>\eql@equations@ang@close
4918   \eql@ampprotect\eql@equations@testall\eql@equations@ang@start
4919 }

```

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

```

4920 \protected\def\eql@ang@close{%
4921   \eql@error{'\string\>' may only close '\string\<'}%
4922 }

```

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

```

4923 \def\eql@equations@ang@start{%
4924   \eql@equations@processopt
4925   \ifdefined\eql@equations@main
4926     \expandafter\eql@scan@ang\expandafter\eql@equations@main
4927   \else
4928     \expandafter\eql@single@start
4929   \fi
4930 }

```

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

```

4931 \def\eql@equations@ang@close{%
4932   \eql@ampprotect\eql@end@testall\eql@equations@ang@end}

```

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

```

4933 \def\eqlequations@ang@end{%
4934   \ifdefined\eqlequations@main\else
4935     \expandafter\eql@single@end
4936   \fi
4937   \eql@verbose@info\eql@verbose@msg@leaveenv
4938   \end{equations@ang}%
4939   \ignorespaces
4940 }

```

## 16 Options

### 16.1 Selection Tools

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

```

4941 \def\eql@decide@abovebelow#1#2#3#4#5{%
4942   \eql@decide@select{#1}{#2}{#3}{%
4943     {abovebelow,both,tb}{#4#5},%
4944     {above,top,t}{#4},%
4945     {below,bottom,b}{#5}}%

```

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

```

4946 \def\eql@decide@situation#1#2#3#4{%
4947   \eql@decide@select{#1}{#2}{#3}{%
4948     {{long}{\def#4{0}}},%
4949     {{short}{\def#4{1}}},%
4950     {{cont}{\def#4{2}}},%
4951     {{par}{\def#4{3}}},%
4952     {{top}{\def#4{4}}},%
4953     {{noskip}{\def#4{5}}},%
4954     {{medskip}{\def#4{6}}}}%

```

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

```

4955 \def\eql@decide@delim#1#2#3{%
4956   \eql@decide@select{#1}{#2}{#3}{%
4957     {{,.,\eql@decide@false}{\eql@box@wrap{}{}},%
4958     {{\eql@decide@true,r,round}{\eql@box@delim()}},%
4959     {{s,sqr,square}{\eql@box@delim[]}},%
4960     {{c,curly,braces}{\eql@box@delim\lbrace\rbrace}},%
4961     {{a,ang,angle}{\eql@box@delim\langle\rangle}},%
4962     {{v,vert}{\eql@box@delim\vert\vert}},%
4963     {{d,dvert}{\eql@box@delim\Vert\Vert}},%
4964     {\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:

```

4965 \eql@define@key{equationsbox}{gathered,gather,ga,lines,ln}[]{}%

```

```

4966 \eql@mode@stacked}
4967 \eql@define@key{equationsbox}{aligned,align,al,columns,col}[]{}%
4968 \eql@mode@aligned}
4969 \eql@define@key{equationsbox}{cases}[]{}%
4970 \eql@mode@cases\eql@box@ldelim\lbrace%
4971 \def\eql@box@colsep{\eql@box@condsep}}
4972 \eql@define@key{equationsbox}{matrix}[r]{}%
4973 \eql@mode@stacked\eql@shape@set{center}%
4974 \let\eql@spread@reset\eql@true\def\eql@spread@val{\z@}%
4975 \def\eql@box@colsep{\eql@box@shortsep}%
4976 \let\eql@mathstyle\empty\eql@punct@clear
4977 \eql@decide@delim{#3}{#2}{#1}}
4978 \eql@define@key{equationsbox}{top,t}[]{\let\eql@box@box\vtop}
4979 \eql@define@key{equationsbox}{center,c}[]{\let\eql@box@box\vcenter}
4980 \eql@define@key{equationsbox}{bottom,b}[]{\let\eql@box@box\vbox}
4981 \eql@define@key{equationsbox}{intro}{}%
4982 \def\eql@box@cases@condintro{#1}}
4983 \eql@define@key{equationsbox}{introtext}{}%
4984 \def\eql@box@cases@condintro{%
4985 \ifmmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
4986 \eql@define@key{equationsbox}{textcond}[true]{}%
4987 \eql@decide@select{#3}{#2}{#1}{%‘
4988 {\eql@decide@true,text}{\let\eql@box@cases@condtext\eql@true}},%
4989 {\eql@decide@false,math}{\let\eql@box@cases@condtext\eql@false}}}}
4990 \eql@define@key{setup}{boxangopt}[]{}%
4991 \def\eql@box@ang@opt{columns,#1}}

```

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

```

4992 \eql@define@key{equations}{equation,eq,single,1}[]{\eql@mode@single}
4993 \eql@define@key{equations}{gathered,gather,ga,lines,ln}[]{}%
4994 \eql@mode@lines}
4995 \eql@define@key{equations}{aligned,align,al,columns,col}[]{}%
4996 \eql@mode@columns}
4997 \eql@define@key{equations,setup}{transpose}[true]{}%
4998 \eql@decide@select{#3}{#2}{#1}{%
4999 {\eql@decide@false{\let\eql@transpose@active\eql@false}},%
5000 {\noamp,plain,restricted}{\let\eql@transpose@active\eql@true}},%
5001 {\eql@decide@true,amp,cont}{\let\eql@transpose@active=+}}}}
5002 \eql@define@key{equations}{native}[true]{}%
5003 \eql@decide@bool{#3}{#2}{#1}\eql@single@native%
5004 \ifdefined\eql@single@native\let\eql@layoutleft\eql@false\fi}
5005 \eql@define@key{setup}{native}[true]{}%
5006 \eql@decide@bool{#3}{#2}{#1}\eql@single@native}
5007 \eql@define@key{setup}{scanequation}[true]{}%
5008 \eql@decide@bool{#3}{#2}{#1}\eql@single@doscan}
5009 \eql@define@key{setup}{scanbox}[true]{}%
5010 \eql@decide@bool{#3}{#2}{#1}\eql@box@doscan}
5011 \eql@define@key{setup}{sqropt}[]{}%
5012 \def\eql@equations@sqr@opt{equation,#1}}
5013 \eql@define@key{setup}{angopt}[]{}%
5014 \def\eql@equations@ang@opt{columns,#1}}

```

**TODO:** describe

```

5015 \eql@define@key{control}{restoreexterior}[]{\eql@display@restore}
5016 \eql@define@key{control}{restoreexterior*}[]{\arrayparboxrestore}

```

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

```

5017 \def\eql@keycat{equations,equationsbox,setup}
5018 \eql@define@key\eql@keycat{spread}{%
5019   \let\eql@spread@reset\eql@false\def\eql@spread@val{#1}}
5020 \eql@define@key\eql@keycat{spread*}[Opt]{%
5021   \let\eql@spread@reset\eql@true\def\eql@spread@val{#1}}
5022 \eql@define@key\eql@keycat{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
5023   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
5024   {{cell}}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
5025   {{tag}}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
5026   {\eql@decide@true
5027     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
5028 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5029 \eql@define@key\eql@keycat{displayheight}[strut]{%
5030   \eql@decide@select{#3}{#2}{#1}{%
5031     {\eql@decide@false{\let\eql@display@height\@undefined}},%
5032     {{strut}}{\def\eql@display@height{\ht\eql@strutbox@}}},%
5033     {\relax{\def\eql@display@height{#1}}}}
5034 \eql@define@key\eql@keycat{displaydepth}[strut]{%
5035   \eql@decide@select{#3}{#2}{#1}{%
5036     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
5037     {{strut}}{\def\eql@display@depth{\dp\eql@strutbox@}}},%
5038     {\relax{\def\eql@display@depth{#1}}}}

```

Settings concerning page breaks:

```

5039 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5040   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
5041   {{high,3}{\eql@displaybreak@pre3}},%
5042   {{med,medium,2}{\eql@displaybreak@pre2}},%
5043   {{low,1}{\eql@displaybreak@pre1}},%
5044   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
5045   {{default,inherit,-1}{\eql@displaybreak@pre\m@ne}}}}
5046 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5047   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
5048   {{high,3}{\eql@displaybreak@post3}},%
5049   {{med,medium,2}{\eql@displaybreak@post2}},%
5050   {{low,1}{\eql@displaybreak@post1}},%
5051   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
5052   {{default,inherit,-1}{\eql@displaybreak@post\m@ne}}}}
5053 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5054   \eql@decide@select{#3}{#2}{#1}{%
5055     {{full,4}{\eql@displaybreak@inter4}},%
5056     {{high,3}{\eql@displaybreak@inter3}},%
5057     {{med,medium,2}{\eql@displaybreak@inter2}},%
5058     {{low,1}{\eql@displaybreak@inter1}},%
5059     {{0,\eql@decide@false}{\eql@displaybreak@inter\z@}}}}
5060 \eql@define@key{equations}{prepenalty}{%
5061   \eql@displaybreak@prepen@\numexpr#1\relax}
5062 \eql@define@key{equations}{postpenalty}{%
5063   \eql@displaybreak@postpen@\numexpr#1\relax}
5064 \eql@define@key{equations,setup}{interpenalty}{%
5065   \interdisplaylinepenalty\numexpr#1\relax}

```

**TODO:** describe

```

5066 \eqld@define@key{control}{vspace}[]{\eqlv@space@add{#1}}
5067 \eqld@define@key{control}{vspace*}[]{\eqlv@space@addfixedbefore{#1}}
5068 \eqld@define@key{control}{vspace!}[]{\eqlv@space@addfixedafter{#1}}
5069 \eqld@define@key{control}{break}[4]{\eqldisplaybreak@level[{#1}]}
5070 \eqld@define@key{control}{penalty}[]{\eqldisplaybreak@star{#1}}

```

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

```

5071 \eqld@define@key{equations}{noskip}[both]{%
5072   \eqld@decide@abovebelow{#3}{#2}{#1}%
5073   {\def\eqld@skip@force@above{5}}%
5074   {\def\eqld@skip@force@below{5}}}
5075 \eqld@define@key{equations}{short}[above]{%
5076   \eqld@decide@abovebelow{#3}{#2}{#1}%
5077   {\def\eqld@skip@force@above{1}}%
5078   {\def\eqld@skip@force@below{1}}}
5079 \eqld@define@key{equations}{long}[both]{%
5080   \eqld@decide@abovebelow{#3}{#2}{#1}%
5081   {\def\eqld@skip@force@above{0}}%
5082   {\def\eqld@skip@force@below{0}}}
5083 \eqld@define@key{equations}{medskip}[both]{%
5084   \eqld@decide@abovebelow{#3}{#2}{#1}%
5085   {\def\eqld@skip@force@above{6}}%
5086   {\def\eqld@skip@force@below{6}}}
5087 \eqld@define@key{equations}{par}[par]{%
5088   \eqld@decide@select{#3}{#2}{#1}{%
5089     {\default,\eqld@decide@false}{\let\eqld@skip@force@leave\undefined}},%
5090     {\cont,hmode}{\let\eqld@skip@force@leave\z@}},%
5091     {\par,vmode}{\let\eqld@skip@force@leave\@one
5092       \ifdefined\eqld@skip@force@below\else
5093         \def\eqld@skip@force@below{3}%
5094         \fi}},%
5095     {\top}{\let\eqld@skip@force@leave\tw@
5096       \ifdefined\eqld@skip@force@below\else
5097         \def\eqld@skip@force@below{4}
5098         \fi}}}}

```

Specify vertical spacing explicitly:

```

5099 \eqld@define@key{equations}{skip}{%
5100   \def\eqld@skip@force@above{7}%
5101   \def\eqld@skip@custom@above{#1}%
5102   \let\eqld@skip@force@below\eqld@skip@force@above
5103   \let\eqld@skip@custom@below\eqld@skip@custom@above}
5104 \eqld@define@key{equations}{aboveskip}{%
5105   \def\eqld@skip@force@above{7}%
5106   \def\eqld@skip@custom@above{#1}}
5107 \eqld@define@key{equations}{belowskip}{%
5108   \def\eqld@skip@force@below{7}%
5109   \def\eqld@skip@custom@below{#1}}
5110 \eqld@define@key{equations}{abovespace}{%
5111   \advance\eqld@abovespace@\glueexpr#1\relax}
5112 \eqld@define@key{equations}{belowspace}{%
5113   \advance\eqld@belowspace@\glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5114 \eqld@define@key{intertext}{skip}{%
5115   \def\eqld@skip@force@above{7}%

```

```

5116 \def\eql@skip@custom@above{#1}%
5117 \let\eql@skip@force@below\eql@skip@force@above
5118 \let\eql@skip@custom@below\eql@skip@custom@above}
5119 \eql@define@key{intertext}{aboveskip}{%
5120 \def\eql@skip@force@below{7}%
5121 \def\eql@skip@custom@below{#1}}
5122 \eql@define@key{intertext}{belowskip}{%
5123 \def\eql@skip@force@above{7}%
5124 \def\eql@skip@custom@above{#1}}
5125 \eql@define@key{intertext}{noskip}[both]{%
5126 \eql@decide@abovebelow{#3}{#2}{#1}%
5127 {\def\eql@skip@force@below{5}}%
5128 {\def\eql@skip@force@above{5}}}
5129 \eql@define@key{intertext}{short}[both]{%
5130 \eql@decide@abovebelow{#3}{#2}{#1}%
5131 {\def\eql@skip@force@below{1}}%
5132 {\def\eql@skip@force@above{1}}}
5133 \eql@define@key{intertext}{long}[both]{%
5134 \eql@decide@abovebelow{#3}{#2}{#1}%
5135 {\def\eql@skip@force@below{0}}%
5136 {\def\eql@skip@force@above{0}}}
5137 \eql@define@key{intertext}{medskip}[both]{%
5138 \eql@decide@abovebelow{#3}{#2}{#1}%
5139 {\def\eql@skip@force@below{6}}%
5140 {\def\eql@skip@force@above{6}}}

```

Configure general vertical spacing behaviour for various situations:

```

5141 \eql@define@key{setup}{skip,longskip}{%
5142 \abovedisplayskip\glueexpr#1\relax
5143 \belowdisplayskip\abovedisplayskip
5144 \def\eql@skip@long@above{#1}%
5145 \let\eql@skip@long@below\eql@skip@long@above}
5146 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5147 \abovedisplayskip\glueexpr#1\relax
5148 \def\eql@skip@long@above{#1}}
5149 \eql@define@key{setup}{belowskip,belowlongskip}{%
5150 \belowdisplayskip\glueexpr#1\relax
5151 \def\eql@skip@long@below{#1}}
5152 \eql@define@key{setup}{aboveshortskip}{%
5153 \abovedisplayshortskip\glueexpr#1\relax
5154 \def\eql@skip@short@above{#1}}
5155 \eql@define@key{setup}{belowshortskip}{%
5156 \belowdisplayshortskip\glueexpr#1\relax
5157 \def\eql@skip@short@below{#1}}
5158 \eql@define@key{setup}{tagskip}{%
5159 \def\eql@skip@tag@above{#1}%
5160 \let\eql@skip@tag@below\eql@skip@tag@above}
5161 \eql@define@key{setup}{abovetagskip}{%
5162 \def\eql@skip@tag@above{#1}}
5163 \eql@define@key{setup}{belowtagskip}{%
5164 \def\eql@skip@tag@below{#1}}
5165 \eql@define@key{setup}{medskip}{%
5166 \def\eql@skip@med@above{#1}%
5167 \let\eql@skip@med@below\eql@skip@med@above}
5168 \eql@define@key{setup}{abovemedskip}{%
5169 \def\eql@skip@med@above{#1}}
5170 \eql@define@key{setup}{belowmedskip}{%
5171 \def\eql@skip@med@below{#1}}

```

```

5172 \eqld@define@key{setup}{abovetopskip}{%
5173   \def\eqld@skip@top@above{#1}}
5174 \eqld@define@key{setup}{belowtopskip}{%
5175   \def\eqld@skip@top@below{#1}}
5176 \eqld@define@key{setup}{aboveparskip}{%
5177   \def\eqld@skip@par@above{#1}}
5178 \eqld@define@key{setup}{belowparskip}{%
5179   \def\eqld@skip@par@below{#1}}
5180 \eqld@define@key{setup}{abovecontskip}{%
5181   \eqld@decide@select{#3}{#2}{#1}{%
5182     {\hide}{\def\eqld@skip@cont@above{\eqld@spread@val-\eqld@skip@long@below}}},%
5183     {\relax{\def\eqld@skip@cont@above{#1}}}}}%
5184 \eqld@define@key{setup}{belowcontskip}{%
5185   \def\eqld@skip@cont@below{#1}}
5186 \eqld@define@key{setup}{shortmode}{%
5187   \eqld@decide@select{#3}{#2}{#1}{%
5188     {\eqld@decide@false,never}{\def\eqld@skip@mode@short{0}}},%
5189     {\above,neverbelow,belowoff}{\def\eqld@skip@mode@short{1}}},%
5190     {\belowone,belowsingle}{\def\eqld@skip@mode@short{2}}},%
5191     {\belowall,always,on}{\def\eqld@skip@mode@short{3}}}}}%
5192 \eqld@define@key{setup}{abovecontmode}{%
5193   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@above}
5194 \eqld@define@key{setup}{belowcontmode}{%
5195   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@cont@below}
5196 \eqld@define@key{setup}{aboveparmode}{%
5197   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@above}
5198 \eqld@define@key{setup}{belowparmode}{%
5199   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@par@below}
5200 \eqld@define@key{setup}{abovetopmode}{%
5201   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@above}
5202 \eqld@define@key{setup}{belowtopmode}{%
5203   \eqld@decide@situation{#3}{#2}{#1}\eqld@skip@mode@top@below}

```

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

```

5204 \def\eqld@keycat{equations,subequations}
5205 \eqld@define@key\eqld@keycat{label}{\eqld@tags@addblock@label{#1}}
5206 \eqld@define@key\eqld@keycat{labelname}{\eqld@tags@addblock@name{#1}}
5207 \eqld@define@key\eqld@keycat{tag}{\eqld@tags@addblock@tag{#1}}
5208 \eqld@define@key\eqld@keycat{tag*}{%
5209   \eqld@tags@addblock@tagform@off\eqld@tags@addblock@tag{#1}}
5210 \eqld@define@key\eqld@keycat{taglabel}{\eqld@tags@addblock@ref{#1}}

```

**TODO:** describe

```

5211 \eqld@define@key{control}{label}{\eqld@tags@add@label{#1}}
5212 \eqld@define@key{control}{labelname}{\eqld@tags@add@name{#1}}
5213 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
5214 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
5215 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}
5216 \eqld@define@key{control}{shifftag}{\eqld@tags@add@raiseshift{#1}}
5217 \eqld@define@key{control}{smashtag}{\eqld@tags@add@raisesmash{#1}}
5218 \eqld@define@key{control}{pushtag}{\eqld@tags@add@forceraise}

```

**TODO:** describe

```

5219 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
5220 \eqld@define@key{setup}{autolabel}[true]{%
5221   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}

```

```

5222 \eqld@define@key{setup}{autotag}[true]{%
5223   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

**Tag Spacing.** Configure horizontal spacing for equation tags:

```

5224 \def\eqld@keycat{equations,setup}
5225 \eqld@define@key\eqld@keycat{tagmargin}[auto]{%
5226   \eqld@decide@select{#3}{#2}{#1}{%
5227     {{auto,\eqld@decide@false}}{\let\eqld@tagmargin@val\undefined}},%
5228     {\relax{\def\eqld@tagmargin@val{#1}}}}}%
5229 \eqld@define@key\eqld@keycat{tagmargin*}{%
5230   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
5231 \eqld@define@key\eqld@keycat{tagmarginratio}{%
5232   \eqld@tagmargin@ratio@{dimexpr#1pt\relax}}
5233 \eqld@define@key\eqld@keycat{tagmarginthreshold}{%
5234   \def\eqld@tagmargin@threshold{#1}}
5235 \eqld@define@key\eqld@keycat{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5236 \eqld@define@key\eqld@keycat{mintagwidth}{%
5237   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5238 \eqld@define@key\eqld@keycat{mintagwidth*}{\settowidth\eqld@tagwidthmin@{#1}}
5239 \eqld@define@key\eqld@keycat{tagsnap}{%
5240   \eqld@decide@select{#3}{#2}{#1}{%
5241     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5242     {\relax{\def\eqld@tagpos@snap{#1}}}}}%

```

**Tag Layout.** Configure methods to declare equation tag layout:

```

5243 \def\eqld@keycat{equations,setup}
5244 \eqld@define@key\eqld@keycat{tagbox,taglayout}{%
5245   \eqld@tags@taglayout@set{#1}}
5246 \eqld@define@key\eqld@keycat{tagbox*,taglayout*}{%
5247   \eqld@tags@taglayout@set@direct{#1}}
5248 \eqld@define@key\eqld@keycat{tagform}{%
5249   \eqld@tags@tagform@set{#1}}
5250 \eqld@define@key\eqld@keycat{tagform*}{%
5251   \eqld@tags@tagform@set@direct{#1}}
5252 \eqld@define@key\eqld@keycat{subeqtemplate}{%
5253   \def\eqld@subequations@template####1####2{#1}%
5254   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}
5255 \eqld@define@key{control}{tagbox,taglayout}{%
5256   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
5257 \eqld@define@key{control}{tagbox*,taglayout*}{%
5258   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
5259 \eqld@define@key{control}{tagform}{%
5260   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
5261 \eqld@define@key{control}{tagform*}{####1}{%
5262   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

**Equation Numbering.** Configure equation numbering schemes:

```

5263 \def\eqld@keycat{equations,setup}
5264 \eqld@define@key\eqld@keycat{numberline,number,num,numline,n}[all]{%
5265   \eqld@decide@select{#3}{#2}{#1}{%
5266     {{\eqld@decide@false,0,*}}{\let\eqld@numbering@active\eqld@false}},%
5267     {{\eqld@decide@true,!}}{\let\eqld@numbering@active\eqld@true}},%
5268     {{none,n,-}}{\let\eqld@numbering@mode\eqld@numbering@mode@multi}}

```



```

5269 \let\eql@numbering@active\eql@false}},%
5270 {{single,1}}{\let\eql@numbering@mode\eql@numbering@mode@single
5271 \let\eql@numbering@active\eql@true}},%
5272 {{multi,@}}{\let\eql@numbering@mode\eql@numbering@mode@multi
5273 \let\eql@numbering@active\eql@true}},%
5274 {\relax{\eql@numbering@set{#1}}}}
5275 \eql@define@key\eql@keycat{nonumber,nn,*}[]{}%
5276 \let\eql@numbering@active\eql@false}
5277 \eql@define@key\eql@keycat{donumber,dn,!}[]{}%
5278 \let\eql@numbering@active\eql@true}
5279 \eql@define@key\eql@keycat{tagsleft,leqno}[]{}{\let\eql@tagsleft\eql@true}
5280 \eql@define@key\eql@keycat{tagsright,reqno}[]{}{\let\eql@tagsleft\eql@false}
5281 \eql@define@key\eql@keycat{tags,eqno}{%
5282 \eql@decide@select{#3}{#2}{#1}}{%
5283 {{right,r}}{\let\eql@tagsleft\eql@false}},%
5284 {{left,l}}{\let\eql@tagsleft\eql@true}}}}
5285 \eql@define@key\eql@keycat{evadetag,avoidtag}[true]{%
5286 \eql@decide@bool{#3}{#2}{#1}\eql@numbering@best@auto}
5287 \eql@define@key\eql@keycat{tagbetween}[true]{%
5288 \eql@decide@bool{#3}{#2}{#1}\eql@tagpos@doconvert}

```

**TODO:** describe

```

5289 \eql@define@key{control}{nonumber,nn,*}[]{}{\global\@eqnswfalse}
5290 \eql@define@key{control}{donumber,dn,!}[]{}{\global\@eqnswtrue}
5291 \eql@define@key{control}{numberhere}[]{}{\eql@numberhere}
5292 \eql@define@key{control}{numbernext}[]{}{\eql@numbernext}

```

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

```

5293 \def\eql@keycat{equations,setup}
5294 \eql@define@key\eql@keycat{layout}{\eql@decide@select{#3}{#2}{#1}}{%
5295 {{center,c}}{\let\eql@layoutleft\eql@false}},%
5296 {{left,l}}{\let\eql@layoutleft\eql@true}}}}
5297 \eql@define@key\eql@keycat{center}[]{}{\let\eql@layoutleft\eql@false}
5298 \eql@define@key\eql@keycat{flushleft,left}[]{}{\let\eql@layoutleft\eql@true}
5299 \eql@define@key\eql@keycat{leftmargin}{\def\eql@layoutleftmargin{#1}}
5300 \eql@define@key\eql@keycat{leftmargin*}{%
5301 \settowidth\dimen@{#1}\edef\eql@layoutleftmargin{\the\dimen@}}
5302 \eql@define@key\eql@keycat{minleftmargin}{%
5303 \def\eql@layoutleftmarginmin{#1}}
5304 \eql@define@key\eql@keycat{maxleftmargin}{%
5305 \eql@decide@select{#3}{#2}{#1}}{%
5306 {\eql@decide@false{\def\eql@layoutleftmarginmax{.5\maxdimen}}},%
5307 {\relax{\def\eql@layoutleftmarginmax{#1}}}}}}
5308 \def\eql@keycat{equations,equationsbox}
5309 \eql@define@key\eql@keycat{margin}{%
5310 \def\eql@display@marginleft{#1}\def\eql@display@marginright{#1}}
5311 \eql@define@key\eql@keycat{marginleft}{\def\eql@display@marginleft{#1}}
5312 \eql@define@key\eql@keycat{marginright}{\def\eql@display@marginright{#1}}
5313 \eql@define@key\eql@keycat{linewidth,width}{\def\eql@display@linewidth{#1}}

```

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

```

5314 \def\eql@keycat{equations,setup}

```

```

5315 \eqld@define@key\eqld@keycat{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5316   {\max,full,4}{\eqld@alignbadness@inf@bad}},%
5317   {\high,3}{\eqld@alignbadness@54\relax}},%
5318   {\med,medium,2}{\eqld@alignbadness@18\relax}},%
5319   {\low,1}{\eqld@alignbadness@6\relax}},%
5320   {\0,\eqld@decide@false}{\eqld@alignbadness@z}}}}
5321 \eqld@define@key\eqld@keycat{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5322   {\max,full,4}{\eqld@tagbadness@inf@bad}},%
5323   {\high,3}{\eqld@tagbadness@54\relax}},%
5324   {\med,medium,2}{\eqld@tagbadness@18\relax}},%
5325   {\low,1}{\eqld@tagbadness@6\relax}},%
5326   {\0,\eqld@decide@false}{\eqld@tagbadness@z}}}}
5327 \eqld@define@key\eqld@keycat{alignbadness}{\eqld@alignbadness@numexpr#1\relax}
5328 \eqld@define@key\eqld@keycat{tagbadness}{\eqld@tagbadness@numexpr#1\relax}
5329 \eqld@define@key\eqld@keycat{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5330   {\0,\eqld@decide@false}{\def\eqld@colsepmin@val{Opt}}},%
5331   {\relax{\def\eqld@colsepmin@val{#1}}}}}}
5332 \eqld@define@key\eqld@keycat{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5333   {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5334   {\relax{\def\eqld@colsepmax@val{#1}}}}}}
5335 \eqld@define@key\eqld@keycat{fulllength}[true]{%
5336   \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

```

**TODO:** is boxcolsep vs breakcolsep okay??!

```

5337 \eqld@define@key\eqld@keycat{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5338   {\0,\eqld@decide@false}{\def\eqld@break@line@sep{Opt}}},%
5339   {\relax{\def\eqld@break@line@sep{#1}}}}}}
5340 \eqld@define@key\eqld@keycat{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
5341   {\0,\eqld@decide@false}{\def\eqld@break@line@shortsep{Opt}}},%
5342   {\relax{\def\eqld@break@line@shortsep{#1}}}}}}
5343 \eqld@define@key{equationsbox,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5344   {\0,\eqld@decide@false}{\def\eqld@box@colsep{Opt}}},%
5345   {\short}{\def\eqld@box@colsep{\eqld@box@shortsep}}},%
5346   {\relax{\def\eqld@box@colsep{#1}}}}}%
5347 \let\eqld@break@col@sep\eqld@box@colsep
5348 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5349   {\0,\eqld@decide@false}{\def\eqld@break@col@sep{Opt}}},%
5350   {\relax{\def\eqld@break@col@sep{#1}}}}}%
5351 \let\eqld@colsepmin@val\eqld@box@colsep
5352 \let\eqld@colsepmax@val\eqld@box@colsep
5353 \let\eqld@box@colsep\eqld@break@col@sep
5354 \eqld@define@key\eqld@keycat{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5355   {\0,\eqld@decide@false}{\def\eqld@break@col@shortsep{Opt}}},%
5356   {\relax{\def\eqld@break@col@shortsep{#1}}}}}}
5357 \eqld@define@key{equationsbox,setup}{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5358   {\0,\eqld@decide@false}{\def\eqld@box@shortsep{Opt}}},%
5359   {\relax{\def\eqld@box@shortsep{#1}}}}}}
5360 \eqld@define@key{equationsbox,setup}{condsep}{\eqld@decide@select{#3}{#2}{#1}{%
5361   {\0,\eqld@decide@false}{\def\eqld@box@condsep{Opt}}},%
5362   {\relax{\def\eqld@box@condsep{#1}}}}}}

```

**Horizontal Shape.** Configure horizontal alignment schemes:

```

5363 \def\eqld@keycat{equations,equationsbox,setup}
5364 \eqld@define@key\eqld@keycat{shape}[default]{\eqld@shape@set{#1}}
5365 \eqld@define@key\eqld@keycat{padding,pad}[indent]{%
5366   \eqld@decide@select{#3}{#2}{#1}{%
5367     {\max}{\let\eqld@paddingleft@val\@undefined}},%

```

```

5368     {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5369     {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5370     {\relax{\def\eql@paddingleft@val{#1}}}}}%
5371 \let\eql@paddingright@val\eql@paddingleft@val}
5372 \eql@define@key\eql@keycat{padleft}[indent]{%
5373   \eql@decide@select{#3}{#2}{#1}{%
5374     {{max}}{\let\eql@paddingleft@val\undefined}},%
5375     {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5376     {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5377     {\relax{\def\eql@paddingleft@val{#1}}}}}%
5378 \eql@define@key\eql@keycat{padright}[indent]{%
5379   \eql@decide@select{#3}{#2}{#1}{%
5380     {{max}}{\let\eql@paddingright@val\undefined}},%
5381     {{indent}}{\def\eql@paddingright@val{\eql@indent@val}}},%
5382     {{0,\eql@decide@false}}{\def\eql@paddingright@val{0pt}}},%
5383     {\relax{\def\eql@paddingright@val{#1}}}}}%
5384 \eql@define@key\eql@keycat{indent}[2em]{%
5385   \def\eql@indent@val{#1}}

```

**TODO:** describe

```

5386 \eql@define@key{control}{align}[]{%
5387   \eql@decide@select{#3}{#2}{#1}{%
5388     {{l,left}}{\global\eql@append\eql@cell@container{\eql@shape@pos@z}}},%
5389     {{c,center}}{\global\eql@append\eql@cell@container{\eql@shape@pos@one}}},%
5390     {{r,right}}{\global\eql@append\eql@cell@container{\eql@shape@pos@two}}}}}%
5391 \eql@define@key{control}{shift,shifto}[]{%
5392   \eql@decide@select{#3}{#2}{#1}{%
5393     {{*,indent}}{\eql@shape@alignamount@set{\eql@indent@}}},%
5394     {{!,outdent}}{\eql@shape@alignamount@set{-\eql@indent@}}},%
5395     {\relax{\eql@shape@alignamount@set{#1}}}}}%
5396 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}

```

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

```

5397 \def\eql@keycat{equations,equationsbox,setup}
5398 \eql@define@key\eql@keycat{classout}{\eql@class@innerleft@set{#1}}
5399 \eql@define@key\eql@keycat{classin}{\eql@class@innerright@set{#1}}
5400 \eql@define@key\eql@keycat{classlead,classin*}{\eql@class@innerlead@set{#1}}
5401 \eql@define@key\eql@keycat{ampeq}[]{\eql@class@ampeq}
5402 \eql@define@key\eql@keycat{eqamp}[]{\eql@class@eqamp}
5403 \eql@define@key\eql@keycat{class}{\eql@decide@select{#3}{#2}{#1}{%
5404   {{ampeq,amprel,eqafter,beforerel}}\eql@class@ampeq},%
5405   {{eqamp,relamp,eqbefore,afterrel}}\eql@class@eqamp}}}

```

**Math Styles.** Configure math classes at alignment marker:

```

5406 \eql@define@key\eql@keycat{style}[display]{%
5407   \eql@decide@select{#3}{#2}{#1}{%
5408     {{text,\eql@decide@false}}{\let\eql@mathstyle\empty}},%
5409     {{display,\eql@decide@true}}{\let\eql@mathstyle\displaystyle}}}%

```

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

```

5410 \def\eql@punct@all#1#2#3#4#5\eql@punct@end{%
5411   \def\eql@tmp{#4}\def\eql@tmpa{1}%
5412   \ifx\eql@tmp\eql@tmpa
5413     \ifnum#5=1111\relax

```

```

5414 \eqlopunct@set\eqlopunct@col{#1}%
5415 \eqlopunct@set\eqlopunct@line{#2}%
5416 \eqlopunct@set\eqlopunct@block{#3}%
5417 \else\ifnum#5=111\relax
5418 \eqlopunct@set\eqlopunct@line{#1}%
5419 \eqlopunct@set\eqlopunct@block{#2}%
5420 \else\ifnum#5=11\relax
5421 \eqlopunct@set\eqlopunct@block{#1}%
5422 \else
5423 \eqlopunct@clear
5424 \fi\fi\fi
5425 \else
5426 \eqlopunct@error{Too many arguments to punctall}%
5427 \fi
5428 }

```

**TODO:** describe

```

5429 \def\eqlopunct@keycat{equations,equationsbox,setup}
5430 \eqlopunct@define@key\eqlopunct@keycat{punctsep}{\,\,}{\def\eqlopunct@sep{#1}}
5431 \eqlopunct@define@key\eqlopunct@keycat{punct}{.}{\eqlopunct@set\eqlopunct@block{#1}}
5432 \eqlopunct@define@key\eqlopunct@keycat{punct*}[]{\eqlopunct@set\eqlopunct@block\relax}
5433 \eqlopunct@define@key\eqlopunct@keycat{punctline}{,}{\eqlopunct@set\eqlopunct@line{#1}}
5434 \eqlopunct@define@key\eqlopunct@keycat{punctline*}[]{\eqlopunct@set\eqlopunct@line\relax}
5435 \eqlopunct@define@key\eqlopunct@keycat{punctcol}{,}{\eqlopunct@set\eqlopunct@col{#1}}
5436 \eqlopunct@define@key\eqlopunct@keycat{punctcol*}[]{\eqlopunct@set\eqlopunct@col\relax}
5437 \eqlopunct@define@key\eqlopunct@keycat{punctall}{,}{\eqlopunct@all#111111\eqlopunct@end}
5438 \eqlopunct@define@key\eqlopunct@keycat{punctterm}[true]{%
5439 \eqlopunct@decide@bool{#3}{#2}{#1}\eqlopunct@term@box}

5440 \eqlopunct@define@key{control}{punctsep}{\,\,}{\def\eqlopunct@sep{#1}}
5441 \eqlopunct@define@key{control}{punct}{.}{\eqlopunct@set\eqlopunct@next{#1}}
5442 \eqlopunct@define@key{control}{punct*}[]{\let\eqlopunct@next\relax}
5443 \eqlopunct@define@key{control}{punctapply}[]{\eqlopunct@apply@top}

```

**Frames.** **TODO:** describe

```

5444 \eqlopunct@define@key{equationsbox}{frame}{\fbox}{%
5445 \def\eqlopunct@box@frame{#1}%
5446 \ifx\eqlopunct@box@frame\empty\let\eqlopunct@box@frame\@firstofone\fi}
5447 \eqlopunct@define@key{equationsbox}{wrap}{\eqlopunct@box@wrap#1}
5448 \eqlopunct@define@key{equationsbox}{delim}{r}{\eqlopunct@decide@delim{#3}{#2}{#1}}
5449 \eqlopunct@define@key{equationsbox}{ldelim}{\eqlopunct@box@ldelim#1}
5450 \eqlopunct@define@key{equationsbox}{rdelim}{\eqlopunct@box@rdelim#1}
5451 \eqlopunct@define@key{equationsbox}{lbrace}[]{\eqlopunct@box@ldelim\lbrace}
5452 \eqlopunct@define@key{equationsbox}{rbrace}[]{\eqlopunct@box@rdelim\rbrace}
5453 \eqlopunct@define@key{equationsbox}{lrbrace,lrbraces}[]{\eqlopunct@box@delim\lbrace\rbrace}
5454 \eqlopunct@define@key\eqlopunct@keycat{braces}{lr}{%
5455 \eqlopunct@decide@select{#3}{#2}{#1}{%
5456 {\eqlopunct@decide@false}{\eqlopunct@box@wrap}{}}},%
5457 {\l,left}{\eqlopunct@box@ldelim\lbrace}},%
5458 {\r,right}{\eqlopunct@box@rdelim\rbrace}},%
5459 {\eqlopunct@decide@true,lr,both}{\eqlopunct@box@delim\lbrace\rbrace}}}}

```

**TODO:** describe

```

5460 \eqlopunct@define@key{control}{framecell}{\fbox}{%
5461 \global\eqlopunct@append\eqlopunct@cell@container{\def\eqlopunct@frame@cmd{#1}}}
5462 \eqlopunct@define@key{control}{frametag}{\fbox}{%
5463 \global\eqlopunct@append\eqlopunct@tags@container{\def\eqlopunct@tags@frame@cmd{#1}}}

```

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

```
5464 \eql@define@key{equations,equationsbox}{alt}{}
```

## Injections.

```
5465 \eql@define@key{control}{inject}{%
5466   \global\eql@append\eql@interline@container{%
5467     \eql@append\eql@display@injectbefore{#1}}
5468 \eql@define@key{control}{inject*}{%
5469   \global\eql@append\eql@interline@container{%
5470     \eql@append\eql@display@injectafter{#1}}
5471 \eql@define@key{control}{markline}[]{\eql@markline@inject{#1}}
5472 \eql@define@key{control}{markline*}[]{\eql@markline@inject{push,#1}}
5473 \eql@define@key{control}{qed}[]{\eql@markline@inject{qed,#1}}
5474 \eql@define@key{control}{qed*}[]{\eql@markline@inject{qed,push,#1}}
```

**TODO:** describe

```
5475 \eql@define@key{markline}{pos}{%
5476   \eql@decide@select{#3}{#2}{#1}{%
5477     {{below,push}}{\let\eql@markline@pos\eql@markline@pos@below}},%
5478     {{baseline}}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
5479     {{bottom}}{\let\eql@markline@pos\eql@markline@pos@bottom}}}%
5480 \eql@define@key{markline}{below,push}[]{%
5481   \let\eql@markline@pos\eql@markline@pos@below}
5482 \eql@define@key{markline}{baseline}[]{%
5483   \let\eql@markline@pos\eql@markline@pos@baseline}
5484 \eql@define@key{markline}{bottom}[]{%
5485   \let\eql@markline@pos\eql@markline@pos@bottom}
5486 \eql@define@key{markline}{shift}{\def\eql@markline@shift{#1}}
5487 \eql@define@key{markline}{symbol}{\def\eql@markline@symbol{#1}}
5488 \eql@define@key{markline}{qed}[]{\let\eql@markline@symbol\eql@markline@qed}
5489 \eql@define@key{setup}{marksymbol}{\def\eql@markline@symbol{#1}}
5490 \eql@define@key{setup}{qedsymbol}{\def\eql@markline@qed{#1}}
5491 \eql@define@key{setup}{markpos}{%
5492   \eql@decide@select{#3}{#2}{#1}{%
5493     {{below}}{\let\eql@markline@pos\eql@markline@pos@below}},%
5494     {{baseline}}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
5495     {{bottom}}{\let\eql@markline@pos\eql@markline@pos@bottom}}}%

```

**Global Switches.** Set global switches:

```
5496 \let\eql@multi@linesfallback\eql@false
5497 \let\eql@scan@par\eql@false
5498 \let\eql@single@cr@mode\eql@false
5499 \let\eql@ampproof@active\eql@false
5500 \let\eql@parseopt@warn@main\eql@warn@parseopt
5501 \let\eql@parseopt@warn@aux\@empty

5502 \eql@define@key{equations,setup}{linesfallback}[true]{%
5503   \eql@decide@select{#3}{#2}{#1}{%
5504     {\eql@decide@false{\let\eql@multi@linesfallback\eql@false}},%
5505     {{reuse,lean}}{\let\eql@multi@linesfallback\z@}},%
5506     {{measure,full,\eql@decide@true}}{\let\eql@multi@linesfallback\eql@true}}}%
5507 \eql@define@key{setup}{ampproof}[true]{%
5508   \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
5509 \eql@define@key{equations,setup}{equationcr}{%
```

```

5510 \eqld@decide@select{#3}{#2}{#1}{%
5511   {\eqld@decide@false{\let\eqld@single@cr@mode\eqld@false}},%
5512   {\eqld@decide@true,break}{\let\eqld@single@cr@mode\eqld@break@line}},%
5513   {\error,verbose}{\let\eqld@single@cr@mode\eqld@single@cr@error}}}%
5514 \eqld@define@key{setup}{modifierwarning}[true]{%
5515   \eqld@decide@select{#3}{#2}{#1}{%
5516     {\eqld@decide@false{\let\eqld@parseopt@warn@main\@empty
5517       \let\eqld@parseopt@warn@aux\@empty}},%
5518     {-}{\let\eqld@parseopt@warn@main\eqld@warn@parseopt
5519       \let\eqld@parseopt@warn@aux\@empty}},%
5520     {\eqld@decide@true{\let\eqld@parseopt@warn@main\eqld@warn@parseopt
5521       \let\eqld@parseopt@warn@aux\eqld@warn@parseopt}},%
5522     {\verbose,+}{\let\eqld@parseopt@warn@main\eqld@warn@parseopt@verbose
5523       \let\eqld@parseopt@warn@aux\eqld@warn@parseopt@verbose}}}%
5524 \eqld@define@key{equations,setup}{rescan}[true]{%
5525   \eqld@decide@if{#3}{#2}{#1}%
5526   {\let\eqld@scan@body\eqld@scan@body@rescan}%
5527   {\let\eqld@scan@body\eqld@scan@body@dump}}
5528 \eqld@define@key{equations,equationsbox,setup}{scanpar}[true]{%
5529   \eqld@decide@bool{#3}{#2}{#1}\eqld@scan@par}
5530 \eqld@define@key{setup}{defaults}{%
5531   \eqld@decide@select{#3}{#2}{#1}{%
5532     {\classic}{\eqld@defaults@classic}},%
5533     {\eqnlines}{\eqld@defaults@eqnlines}}}%
5534 \eqld@define@key{equations,equationsbox,setup}{verbose}[true]{%
5535   \eqld@decide@if{#3}{#2}{#1}\eqld@verbose@on\eqld@verbose@off}

```

**Package Options.** Declare choices available at loading of package only: **TODO:** adjust

```

5536 \let\eqld@provide@opt@env\tw@
5537 \let\eqld@provide@opt@amsmathpatch\eqld@false
5538 \let\eqld@provide@opt@backup\eqld@false
5539 \let\eqld@provide@opt@ang\eqld@true
5540 \let\eqld@provide@opt@eqref\eqld@true
5541 \let\eqld@provide@opt@matrix\eqld@true

5542 \eqld@define@key{setup}{amsmathends,amsmathpatch}[true]{%
5543   \eqld@error@packageoption{#2}%
5544   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@amsmathpatch}
5545 \eqld@define@key{setup}{backup}[true]{%
5546   \eqld@error@packageoption{#2}%
5547   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@backup}
5548 \eqld@define@key{setup}{env}[equation]{%
5549   \eqld@error@packageoption{#2}%
5550   \eqld@decide@select{#3}{#2}{#1}{%
5551     {\none,\eqld@decide@false}{\let\eqld@provide@opt@env\z@}},%
5552     {\equation,latex}{\let\eqld@provide@opt@env\@ne}},%
5553     {\amsmath,all,\eqld@decide@true}{\let\eqld@provide@opt@env\tw@}}}%
5554 \eqld@define@key{setup}{ang}[true]{%
5555   \eqld@error@packageoption{#2}%
5556   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@ang}
5557 \eqld@define@key{setup}{eqref}[true]{%
5558   \eqld@error@packageoption{#2}%
5559   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@eqref}
5560 \eqld@define@key{setup}{matrix}[true]{%
5561   \eqld@error@packageoption{#2}%
5562   \eqld@decide@bool{#3}{#2}{#1}\eqld@provide@opt@matrix}

```



Shortcut Options. **TODO:** describe

```

5563 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
5564 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
5565 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
5566 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
5567 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
5568 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
5569 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
5570 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
5571 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
5572 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
5573 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
5574 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
5575 \def\eql@parseopt@puncttermopt#1{\eqnaddopt{punctterm}\eql@parseopt@peek}
5576 \def\eql@parseopt@punctnext#1#2{%
5577   \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
5578 \def\eql@parseopt@punctblock#1#2{%
5579   \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
5580 \def\eql@parseopt@puncttermbox#1{%
5581   \let\eql@punct@term@box\eql@true\eql@parseopt@peek}
5582 \def\eql@parseopt@puncttermcr#1{%
5583   \let\eql@punct@term@cr\eql@true\eql@parseopt@peek}
5584 \def\eql@parseopt@vspace[#1]{%
5585   \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:

```

5586 \def\eql@defaults@classic{%
5587   \eqnlineset{numberline=all}%
5588   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
5589   \eqnlineset{maxcolsep=off}%
5590   \eqnlineset{spread={\jot}}%
5591   \eqnlineset{tagmargin}%
5592   \eqnlineset{tagmarginratio=1}%
5593   \eqnlineset{tagmarginthreshold=0.5}%
5594   \eqnlineset{leftmargin={\leftmargini}}%
5595   \eqnlineset{padding=max}%
5596   \eqnlineset{evadetag=off}%
5597   \eqnlineset{displayheight=off}%
5598   \eqnlineset{displaydepth=off}%
5599   \eqnlineset{shortmode=belowsingle}%
5600   \eqnlineset{abovecontmode=short}%
5601   \eqnlineset{belowcontmode=short}%
5602   \eqnlineset{aboveparmode=long}%
5603   \eqnlineset{belowparmode=long}%
5604   \eqnlineset{abovetopmode=long}%
5605   \eqnlineset{belowtopmode=long}%
5606   \eqnlineset{abovelongskip={\abovedisplayskip}}%
5607   \eqnlineset{belowlongskip={\belowdisplayskip}}%
5608   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%

```

```

5609 \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
5610 \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
5611 \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
5612 \eqnlineset{abovecontskip=0pt}%
5613 \eqnlineset{belowcontskip=0pt}%
5614 \eqnlineset{aboveparskip=0pt}%
5615 \eqnlineset{belowparskip=0pt}%
5616 \eqnlineset{abovetopskip=0pt}%
5617 \eqnlineset{belowtopskip=0pt}%
5618 \eqnlineset{abovetagskip=0pt}%
5619 \eqnlineset{belowtagskip=0pt}%
5620 \eqnlineset{equationcr=off}%
5621 \eqnlineset{linesfallback=false}%
5622 }

```

values based on 10pt vs 12pt

`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):

```

5623 \def\eqn@defaults@eqnlines{%
5624   \eqnlineset{numberline=all}%
5625   \eqnlineset{mintagsep=.5em}%
5626   \eqnlineset{maxcolsep=2em}%
5627   \eqnlineset{spread={0.2\normalbaselineskip}}%
5628   \eqnlineset{tagmargin}%
5629   \eqnlineset{tagmarginratio=.334}%
5630   \eqnlineset{tagmarginthreshold=0.5}%
5631   \eqnlineset{leftmargin={\leftmargini}}%
5632   \eqnlineset{padding=0pt}%
5633   \eqnlineset{evadetag}%
5634   \eqnlineset{displayheight=strut}%
5635   \eqnlineset{displaydepth=strut}%
5636   \eqnlineset{shortmode=above}%
5637   \eqnlineset{abovecontmode=noskip}%
5638   \eqnlineset{belowcontmode=long}%
5639   \eqnlineset{aboveparmode=long}%
5640   \eqnlineset{belowparmode=long}%
5641   \eqnlineset{abovetopmode=noskip}%
5642   \eqnlineset{belowtopmode=long}%
5643   \eqnlineset{longskip={0.75\normalbaselineskip
5644     plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
5645   \eqnlineset{aboveshortskip={0.0\normalbaselineskip
5646     plus 0.25\normalbaselineskip}}%
5647   \eqnlineset{belowshortskip={0.0\normalbaselineskip
5648     plus 0.25\normalbaselineskip}}%
5649   \eqnlineset{medskip={0.4\normalbaselineskip
5650     plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
5651   \eqnlineset{abovecontskip=0pt}%
5652   \eqnlineset{belowcontskip=0pt}%
5653   \eqnlineset{aboveparskip=0pt}%
5654   \eqnlineset{belowparskip=0pt}%
5655   \eqnlineset{abovetopskip=0pt}%
5656   \eqnlineset{belowtopskip=0pt}%
5657   \eqnlineset{abovetagskip={0.25\normalbaselineskip

```



```

5658     minus 0.25\normalbaselineskip}}%
5659 \eqnlineset{belowtagskip={0.25\normalbaselineskip
5660     minus 0.25\normalbaselineskip}}}%
5661 \eqnlineset{equationcr=break}%
5662 \eqnlineset{linesfallback=true}%
5663 }

```

## 16.4 Component Selection

The following routines provide several additional math environments beyond `equations`. They also backup and overwrite the original routines of `LATEX` and `amsmath` carefully.

### Tools.

`\eql@provide@movecmd` We introduce a couple of tools to rename and undefine commands and environments:

```

\provide@undefinecmd 5664 \def\eql@provide@movecmd#1#2{%
\provide@undefineenv 5665 \eql@letcs{#1\expandafter}\csname#2\endcsname
@provide@undefineenv 5666 }
5667 \def\eql@provide@moveenv#1#2{%
5668 \eql@provide@movecmd{#1}{#2}%
5669 \ifcsname end#2\endcsname
5670 \eql@provide@movecmd{end#1}{end#2}%
5671 \fi
5672 }
5673 \def\eql@provide@undefinecmd#1{%
5674 \eql@letcs{#1}\@undefined
5675 }
5676 \def\eql@provide@undefineenv#1{%
5677 \eql@provide@undefinecmd{#1}%
5678 \eql@provide@undefinecmd{end#1}%
5679 }

```

**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:

```

5680 \def\eql@amsmath@endfix#1#2{%
5681 \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
5682 \expandafter\ifx\csname end#1\endcsname\eql@tmpa
5683 \eql@provide@movecmd{end#1}{end#2}%
5684 \fi
5685 }

```

`\eql@amsmath@fixmatrix` **TODO:** describe

```

\amsmath@fixmatrixend 5686 \def\eql@amsmath@fixmatrix#1{%
5687 \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
5688 \beginngroup

```

```

5689 \let\matrix@check\@gobble
5690 \def\env@matrix{\noexpand\env@matrix}%
5691 \def\env@cases{\noexpand\env@cases}%
5692 \global\edef\eql@tmp{\eql@tmp}%
5693 \endgroup
5694 \eql@letcs{#1}\eql@tmp
5695 }
5696 \def\eql@amsmath@fixmatrixend#1{%
5697 \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
5698 \begingroup
5699 \expandafter\def\expandafter\endmatrix\expandafter{%
5700 \expandafter\unexpanded\expandafter{\endmatrix}}%
5701 \global\long\edef\eql@tmp{\eql@tmp}%
5702 \endgroup
5703 \eql@letcs{end#1}\eql@tmp
5704 }

```

`\eql@amsmath@fixends` Perform the replacement for all amsmath environments whenever amsmath is loaded:

```

5705 \def\eql@amsmath@fixends{%
5706 \eql@amsmath@after{%
5707 \eql@amsmath@endfix{flalign}{align}%
5708 \eql@amsmath@endfix{alignat}{align}%
5709 \eql@amsmath@endfix{xalignat}{align}%
5710 \eql@amsmath@endfix{xxalignat}{align}%
5711 \eql@amsmath@endfix{gather*}{gather}%
5712 \eql@amsmath@endfix{multline*}{multline}%
5713 \eql@amsmath@endfix{align*}{align}%
5714 \eql@amsmath@endfix{flalign*}{align}%
5715 \eql@amsmath@endfix{alignat*}{align}%
5716 \eql@amsmath@endfix{xalignat*}{align}%
5717 \eql@amsmath@endfix{gathered}{aligned}%
5718 \eql@amsmath@endfix{alignedat}{aligned}%
5719 }
5720 }

```

`@amsmath@fixmatrices` Perform the replacement for all amsmath environments whenever amsmath is loaded:

```

5721 \def\eql@amsmath@fixmatrices{%
5722 \eql@amsmath@after{%
5723 \eql@amsmath@fixmatrix{cases}%
5724 \eql@amsmath@fixmatrix{matrix}%
5725 \eql@amsmath@fixmatrix{pmatrix}%
5726 \eql@amsmath@fixmatrixend{pmatrix}%
5727 \eql@amsmath@fixmatrixend{bmatrix}%
5728 \eql@amsmath@fixmatrixend{Bmatrix}%
5729 \eql@amsmath@fixmatrixend{vmatrix}%
5730 \eql@amsmath@fixmatrixend{Vmatrix}%
5731 }
5732 }

```

**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

```

5733 \def\eql@provide@backup@amsenv#1{%

```

```

5734 \eql@amsmath@after{%
5735   \eql@provide@moveenv{ams#1}{#1}%
5736   \eql@tagging@register@luamml{ams#1}%
5737   \eql@markline@amsthm@move{ams#1}{#1}%
5738 }%
5739 }

```

provide@backup@amsbox **TODO:** describe

```

5740 \def\eql@provide@backup@amsbox#1{%
5741   \eql@amsmath@after{%
5742     \eql@provide@moveenv{ams#1}{#1}%
5743   }%
5744 }

```

provide@backup@eqref Copy an eqref to amseqref whenever amsmath is loaded:

```

5745 \def\eql@provide@backup@eqref{%
5746   \eql@amsmath@after{%
5747     \eql@provide@movecmd{amseqref}{eqref}%
5748   }%
5749 }

```

vide@backup@multlined The environment multlined is supplied by mathtools. We copy it to amsmultlined anyway, but whenever mathtools is loaded:

```

5750 \def\eql@provide@backup@multlined{%
5751   \AddToHook{package/mathtools/after}{%
5752     \eql@provide@moveenv{amsmultlined}{multlined}}%
5753 }

```

vide@backup@equation The L<sup>A</sup>T<sub>E</sub>X environment equation is overwritten by several packages to implement their adjustments. Here we cater for adjustments through amsmath, hyperref and the PDF tagging mechanism. Copy equation and equation\* whenever amsmath is loaded. Whenever hyperref is loaded, and amsmath is not yet present, backup the original L<sup>A</sup>T<sub>E</sub>X and hyperref versions of equation. If neither hyperref nor amsmath are present, just backup the original L<sup>A</sup>T<sub>E</sub>X 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:

```

5754 \def\eql@provide@backup@equation{%
5755   \eql@amsmath@after{%
5756     \eql@provide@moveenv{amsequation}{equation}%
5757     \eql@provide@moveenv{amsequation*}{equation*}%
5758     \eql@tagging@register@env{amsequation}%
5759     \eql@tagging@register@env{amsequation*}%
5760     \eql@tagging@register@luamml{amsequation}%
5761     \eql@tagging@register@luamml{amsequation*}%
5762     \eql@markline@amsthm@move{amsequation}{equation}%
5763     \eql@markline@amsthm@move{amsequation*}{equation*}%
5764   }%
5765   \AddToHook{package/hyperref/after}{%
5766     \ifpackageloaded{amsmath}{%
5767       \eql@provide@moveenv{hyperrefequation}{equation}%
5768       \eql@tagging@register@env{hyperrefequation}%
5769       \eql@tagging@register@luamml{hyperrefequation}%
5770       \eql@markline@amsthm@move{hyperrefequation}{equation}%
5771     }%

```

```

5772 }%
5773 \@ifpackageloaded{amsmath}{\%
5774   \@ifpackageloaded{hyperref}{\%
5775     \let\latexequation\H@equation
5776     \let\endlatexequation\H@endequation
5777   }\eql@provide@moveenv{latexequation}{equation}}%
5778   \eql@tagging@register@env{latexequation}%
5779   \eql@tagging@register@luamml{latexequation}%
5780   \eql@markline@amsthm@move{latexequation}{equation}%
5781 }%
5782 }

```

e@backup@displaymath **TODO:** describe

```

5783 \def\eql@provide@backup@displaymath{%
5784   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
5785   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
5786 }

```

@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:

```

5787 \def\eql@provide@backup@subequations{%
5788   \eql@amsmath@after{%
5789     \eql@provide@moveenv{amssubequations}{subequations}%
5790   }%
5791   \AddToHook{package/hyperref/after}{%
5792     \AddToHook{cmd/amssubequations/before}{%
5793       {%
5794         \stepcounter{equation}%
5795         \protected@edef\theHparentequation{\theHequation}%
5796         \addtocounter{equation}{-1}%
5797       }%
5798       \AddToHook{cmd/amssubequations/after}{%
5799         {%
5800           \def\theHequation{\theHparentequation\alph{equation}}%
5801           \ignorespaces
5802         }%
5803       }%
5804 }

```

\eql@provide@backup Backup all amsmath environments:

```

5805 \def\eql@provide@backup{%
5806   \eql@provide@backup@eqref
5807   \eql@provide@backup@equation
5808   \eql@provide@backup@displaymath
5809   \eql@provide@backup@amsenv{gather}%
5810   \eql@provide@backup@amsenv{multline}%
5811   \eql@provide@backup@amsenv{align}%
5812   \eql@provide@backup@amsenv{flalign}%
5813   \eql@provide@backup@amsenv{alignat}%
5814   \eql@provide@backup@amsenv{xalignat}%
5815   \eql@provide@backup@amsenv{xxalignat}%
5816   \eql@provide@backup@amsenv{gather*}%
5817   \eql@provide@backup@amsenv{multline*}%
5818   \eql@provide@backup@amsenv{align*}%
5819   \eql@provide@backup@amsenv{flalign*}%

```

```

5820 \eql@provide@backup@amsenv{alignat*}%
5821 \eql@provide@backup@amsenv{xalignat*}%
5822 \eql@provide@backup@amsbox{gathered}%
5823 \eql@provide@backup@multlined
5824 \eql@provide@backup@amsbox{aligned}%
5825 \eql@provide@backup@amsbox{alignedat}%
5826 \eql@provide@backup@amsbox{cases}%
5827 \eql@provide@backup@amsbox{matrix}%
5828 \eql@provide@backup@amsbox{pmatrix}%
5829 \eql@provide@backup@amsbox{bmatrix}%
5830 \eql@provide@backup@amsbox{Bmatrix}%
5831 \eql@provide@backup@amsbox{vmatrix}%
5832 \eql@provide@backup@amsbox{Vmatrix}%
5833 \eql@provide@backup@subequations
5834 }

```

**Replacement amsmath Environments.** **TODO:** describe

```

5835 \def\eql@alignat@gobblecol#1{%
5836   \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.*)

```

5837 \newenvironment{eql@gathered}
5838   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
5839 \newenvironment{eql@multlined}
5840   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
5841 \newenvironment{eql@aligned}
5842   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
5843 \newenvironment{eql@alignedat}
5844   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
5845   {\endequationsbox}
5846 \newenvironment{eql@cases}
5847   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
5848 \newenvironment{eql@matrix}
5849   {\eqnaddopt{matrix=.}\equationsbox}{\endequationsbox}
5850 \newenvironment{eql@pmatrix}
5851   {\eqnaddopt{matrix=r}\equationsbox}{\endequationsbox}
5852 \newenvironment{eql@bmatrix}
5853   {\eqnaddopt{matrix=s}\equationsbox}{\endequationsbox}
5854 \newenvironment{eql@Bmatrix}
5855   {\eqnaddopt{matrix=c}\equationsbox}{\endequationsbox}
5856 \newenvironment{eql@vmatrix}
5857   {\eqnaddopt{matrix=v}\equationsbox}{\endequationsbox}
5858 \newenvironment{eql@Vmatrix}
5859   {\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.*)

```

5860 \newenvironment{eql@equation}
5861   {\eqnaddopt{equation}\equations}{\endequations}
5862 \newenvironment{eql@displaymath}
5863   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
5864 \newenvironment{eql@gather}
5865   {\eqnaddopt{lines}\equations}{\endequations}

```

```

5866 \newenvironment{eql@multline}
5867   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out}\equations}
5868   {\endequations}
5869 \newenvironment{eql@align}
5870   {\eqnaddopt{columns}\equations}{\endequations}
5871 \newenvironment{eql@flalign}
5872   {\eqnaddopt{fulllength}\eql@align}{\endequations}
5873 \newenvironment{eql@alignat}
5874   {\eqnaddopt{colsep=off}\eql@xalignat}{\endequations}
5875 \newenvironment{eql@xalignat}
5876   {\eql@alignat@gobblecol\eql@align}{\endequations}
5877 \newenvironment{eql@xxalignat}
5878   {\eqnaddopt{fulllength}\eql@xalignat}{\endequations}
5879 \newenvironment{eql@equation*}
5880   {\eqnaddopt{nonumber}\eql@equation}{\endequations}
5881 \newenvironment{eql@gather*}
5882   {\eqnaddopt{nonumber}\eql@gather}{\endequations}
5883 \newenvironment{eql@multline*}
5884   {\eqnaddopt{nonumber}\eql@multline}{\endequations}
5885 \newenvironment{eql@align*}
5886   {\eqnaddopt{nonumber}\eql@align}{\endequations}
5887 \newenvironment{eql@flalign*}
5888   {\eqnaddopt{nonumber}\eql@flalign}{\endequations}
5889 \newenvironment{eql@alignat*}
5890   {\eqnaddopt{nonumber}\eql@alignat}{\endequations}
5891 \newenvironment{eql@xalignat*}
5892   {\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):

```

5893 \def\eql@provide@onlyonce#1#2{%
5894   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
5895   \ifx\eql@tmp\eql@tmpa
5896     \let\eql@tmp\empty
5897   \fi
5898   \ifx\eql@tmp\empty
5899     \let\eql@tmp\@undefined
5900     \ifx\@nodocument\relax
5901       \def\eql@tmp{#1}%
5902     \fi
5903     \ifcsname eql@provided@#1\endcsname
5904       \def\eql@tmp{#1}%
5905     \fi
5906     \eql@letcs{eql@provided@#1}\eql@true
5907   \else
5908     \def\eql@tmpa{*}%
5909     \ifx\eql@tmp\eql@tmpa
5910       \def\eql@tmp{#1}%
5911     \fi
5912   \fi
5913 }

```

`\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:

```

5914 \def\eql@provide@eqref#1{%
5915   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
5916   \ifx\eql@tmp\eql@tmpa
5917     \let\eql@tmp\empty
5918   \fi
5919   \ifx\eql@tmp\empty
5920     \eql@provide@onlyonce{eqref}{}%
5921   \else
5922     \def\eql@tmpa{*}%
5923     \ifx\eql@tmp\eql@tmpa
5924       \def\eql@tmp{eqref}%
5925     \else
5926       \edef\eql@tmp{\expandafter\@gobble\string#1}%
5927     \fi
5928   \fi
5929   \ifdefined\eql@tmp
5930     \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eql@eqref}%
5931   \else
5932     \eql@amsmath@after{%
5933       \eql@provide@movecmd{eqref}{eql@eqref}%
5934     }%
5935     \AddToHook{package/mathtools/after}{%
5936       \eql@provide@movecmd{eqref}{eql@eqref}%
5937     }%
5938     \eql@provide@movecmd{eqref}{eql@eqref}%
5939     \eql@amsmath@undefine\eqref
5940   \fi
5941 }

```

`\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:

```

5942 \def\eql@provide@amsenv#1#2{%
5943   \eql@provide@onlyonce{#1}{#2}%
5944   \ifdefined\eql@tmp
5945     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5946     \eql@tagging@register@luamml{\eql@tmp}%
5947     \eql@markline@amsthm@register{\eql@tmp}%
5948   \else
5949     \eql@amsmath@after{%
5950       \eql@provide@moveenv{#1}{eql@#1}%
5951       \eql@markline@amsthm@register{#1}%
5952     }%
5953     \AddToHook{package/mathtools/after}{%
5954       \eql@provide@moveenv{#1}{eql@#1}%
5955       \eql@markline@amsthm@register{#1}%
5956     }%
5957     \eql@provide@moveenv{#1}{eql@#1}%
5958     \eql@markline@amsthm@register{#1}%
5959     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5960   \fi
5961 }

```

`\eql@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:

```

5962 \def\eql@provide@amsbox#1#2{%
5963   \eql@provide@onlyonce{#1}{#2}%
5964   \ifdefined\eql@tmp
5965     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5966   \else
5967     \eql@amsmath@after{%
5968       \eql@provide@moveenv{#1}{eql@#1}}%
5969     \AddToHook{package/mathtools/after}{%
5970       \eql@provide@moveenv{#1}{eql@#1}}%
5971     \eql@provide@moveenv{#1}{eql@#1}%
5972     \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
5973   \fi
5974 }
```

`\eql@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:

```

5975 \def\eql@provide@multlined#1{%
5976   \eql@provide@onlyonce{multlined}{#1}%
5977   \ifdefined\eql@tmp
5978     \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
5979   \else
5980     \AddToHook{package/mathtools/after}{%
5981       \eql@provide@moveenv{multlined}{eql@multlined}}%
5982     \eql@provide@moveenv{multlined}{eql@multlined}%
5983     \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
5984       \eql@provide@undefineenv{multlined}}}%
5985   \fi
5986 }
```

`\eql@provide@matrix` Provide the cases and matrix environments. Copy into place, and copy again when amsmath is loaded:

```

5987 \def\eql@provide@matrix#1#2#3{%
5988   \eql@provide@onlyonce{#1}{#3}%
5989   \ifdefined\eql@tmp
5990     \eql@provide@moveenv{\eql@tmp}{eql@#1}%
5991     \eql@tagging@register@luamml{\eql@tmp}%
5992   \else
5993     \eql@amsmath@after{%
5994       \eql@provide@moveenv{#1}{eql@#1}%
5995     }%
5996     \eql@provide@moveenv{#1}{eql@#1}%
5997     \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
5998   \fi
5999 }
```

`\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:

```

6000 \def\eql@provide@equation#1{%
6001   \eql@provide@onlyonce{equation}{#1}%
6002   \ifdefined\eql@tmp
6003     \eql@provide@moveenv{\eql@tmp}{eql@equation}%

```



```

6004 \eql@tagging@register@luamml{\eql@tmp}%
6005 \eql@markline@amsthm@register{\eql@tmp}%
6006 \else
6007 \eql@amsmath@after{%
6008 \eql@provide@moveenv{equation}{eql@equation}%
6009 \eql@markline@amsthm@register{equation}%
6010 }%
6011 \AddToHook{package/hyperref/after}{%
6012 \ifpackageloaded{amsmath}{}{%
6013 \eql@provide@moveenv{equation}{eql@equation}%
6014 \eql@markline@amsthm@register{equation}%
6015 }%
6016 }%
6017 \eql@provide@moveenv{equation}{eql@equation}%
6018 \eql@markline@amsthm@register{equation}%
6019 \ifdefined\eql@tagging@on
6020 \AddToHook{begindocument/end}{%
6021 \eql@provide@moveenv{equation}{eql@equation}%
6022 \eql@markline@amsthm@register{equation}%
6023 }%
6024 \fi
6025 \fi
6026 }

```

**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:

```

6027 \def\eql@provide@equationstar#1{%
6028 \eql@provide@onlyonce{equation*}{#1}%
6029 \ifdefined\eql@tmp
6030 \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
6031 \eql@tagging@register@luamml{\eql@tmp}%
6032 \eql@markline@amsthm@register{\eql@tmp}%
6033 \else
6034 \eql@amsmath@after{%
6035 \eql@provide@moveenv{equation*}{eql@equation*}%
6036 \eql@markline@amsthm@register{equation*}%
6037 }%
6038 \eql@provide@moveenv{equation*}{eql@equation*}%
6039 \eql@markline@amsthm@register{equation*}%
6040 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
6041 \ifdefined\eql@tagging@on
6042 \AddToHook{begindocument/end}{%
6043 \eql@provide@moveenv{equation*}{eql@equation*}%
6044 \eql@markline@amsthm@register{equation*}%
6045 }%
6046 \fi
6047 \fi
6048 }

```

**@provide@displaymath** **TODO:** describe

```

6049 \def\eql@provide@displaymath#1{%
6050 \eql@provide@onlyonce{displaymath}{#1}%
6051 \ifdefined\eql@tmp
6052 \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
6053 \eql@markline@amsthm@register{\eql@tmp}%

```

```

6054 \eql@tagging@register@luamml{\eql@tmp}%
6055 \else
6056 \eql@provide@moveenv{displaymath}{eql@displaymath}%
6057 \eql@markline@amsthm@register{displaymath}%
6058 \ifdefined\eql@tagging@on
6059 \AddToHook{begindocument/end}{%
6060 \eql@provide@moveenv{displaymath}{eql@displaymath}}%
6061 \fi
6062 \fi
6063 }

```

**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:

```

6064 \def\eql@provide@subequations#1{%
6065 \eql@provide@onlyonce{subequations}{#1}%
6066 \ifdefined\eql@tmp
6067 \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
6068 \else
6069 \eql@amsmath@after{%
6070 \eql@provide@moveenv{subequations}{eql@subequations}%
6071 }%
6072 \AddToHook{package/hyperref/after}{%
6073 \AddToHook{cmd/subequations/before}[hyperref]{}%
6074 \AddToHook{cmd/subequations/after}[hyperref]{}%
6075 \RemoveFromHook{cmd/subequations/before}[hyperref]%
6076 \RemoveFromHook{cmd/subequations/after}[hyperref]%
6077 \AddToHook{begindocument/end}{%
6078 \eql@provide@moveenv{subequations}{eql@subequations}}%
6079 }%
6080 \eql@provide@moveenv{subequations}{eql@subequations}%
6081 \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6082 \fi
6083 }

```

**\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:

```

6084 \def\eql@provide@sqr{%
6085 \let\[ \eql@sqr@open
6086 \let\] \eql@sqr@close
6087 \eql@amsmath@after{%
6088 \let\[ \eql@sqr@open
6089 \let\] \eql@sqr@close
6090 }%
6091 \ifdefined\eql@tagging@on
6092 \AddToHook{begindocument/end}{%
6093 \let\[ \eql@sqr@open
6094 \let\] \eql@sqr@close
6095 }%
6096 \fi
6097 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```
6098 \def\eql@provide@ang{%
6099   \let\<\eql@ang@open
6100   \let\>\eql@ang@close
6101 }
```

## Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```
6102 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
6103 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
6104 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
6105 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
6106 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
6107 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
6108 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
6109 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
6110 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
6111 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
6112 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
6113 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
6114 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
6115 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}
6116 \eql@define@key{provide}{alignat*}[]{\eql@provide@amsenv{alignat*}{#1}}
6117 \eql@define@key{provide}{xalignat*}[]{\eql@provide@amsenv{xalignat*}{#1}}
6118 \eql@define@key{provide}{gathered}[]{\eql@provide@amsbox{gathered}{#1}}
6119 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
6120 \eql@define@key{provide}{aligned}[]{\eql@provide@amsbox{aligned}{#1}}
6121 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsbox{alignedat}{#1}}
6122 \eql@define@key{provide}{cases}[]{\eql@provide@matrix{cases}\eql@false{#1}}
6123 \eql@define@key{provide}{matrix}[]{\eql@provide@matrix{matrix}\eql@false{#1}}
6124 \eql@define@key{provide}{pmatrix}[]{\eql@provide@matrix{pmatrix}\eql@false{#1}}
6125 \eql@define@key{provide}{bmatrix}[]{\eql@provide@matrix{bmatrix}\eql@true{#1}}
6126 \eql@define@key{provide}{Bmatrix}[]{\eql@provide@matrix{Bmatrix}\eql@true{#1}}
6127 \eql@define@key{provide}{vmatrix}[]{\eql@provide@matrix{vmatrix}\eql@true{#1}}
6128 \eql@define@key{provide}{Vmatrix}[]{\eql@provide@matrix{Vmatrix}\eql@true{#1}}
6129 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
6130 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
6131 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
6132 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
6133 \eql@define@key{provide}{tagform}[]{%
6134   \def\tagform@##1{\maketag@@@{\eql@tags@tagform{#1}}}%
6135 \eql@define@key{provide}{maketag}[]{%
6136   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}
```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```
6137 \newcommand{\eqnlinesprovide}[1]{%
6138   \eql@setkeys{provide}{#1}%
6139   \ignorespaces
6140 }
```

## 16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6141 \let\eql@error@packageoption\@gobble
```

Declare math layout options `leqno` and `fleqn` for common L<sup>A</sup>T<sub>E</sub>X classes:

```
6142 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
```

```
6143 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
6144 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
6145 \eql@defaults@eqnlines
```

```
6146 \eql@mode@columns
```

```
6147 \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<sub>E</sub>X conditional parsing mechanism:

```
6148 \ifdefined\tagsleft@true\else
```

```
6149   \expandafter\newif\csname iftagsleft@\endcsname
```

```
6150 \fi
```

```
6151 \ifdefined\@fleqntrue\else
```

```
6152   \expandafter\newif\csname if@fleqn\endcsname
```

```
6153 \fi
```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
6154 \eql@amsmath@after{%
```

```
6155   \ifnum\eql@provide@opt@env=\tw@
```

```
6156     \iftagsleft@
```

```
6157       \eqnlineset{tags=left}%
```

```
6158     \else
```

```
6159       \eqnlineset{tags=right}%
```

```
6160     \fi
```

```
6161     \if@fleqn
```

```
6162       \eqnlineset{layout=left}%
```

```
6163     \else
```

```
6164       \eqnlineset{layout=center}%
```

```
6165     \fi
```

```
6166   \fi
```

```
6167 }
```

Process package options:

```
6168 \ProcessOptions
```

`\error@packageoption` Enable error message for exclusive package options:

```
6169 \def\eql@error@packageoption#1{%
```

```
6170   \eql@error{may only use '#1' as a package option}%
```

```
6171 }
```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```

6172 \ifnum\eql@provide@opt@env=\tw@
6173 \ifdefined\eql@provide@opt@matrix
6174   \let\eql@provide@opt@amsmathpatch\eql@false
6175 \fi\fi
6176 \ifdefined\eql@provide@opt@backup
6177   \let\eql@provide@opt@amsmathpatch\eql@true
6178 \fi
6179 \ifdefined\eql@provide@opt@amsmathpatch
6180   \eql@amsmath@fixends
6181   \eql@amsmath@fixmatrices
6182 \fi

```

Backup all amsmath environments that may be overwritten to `ams...`. This will happen before any replacements:

```

6183 \ifdefined\eql@provide@opt@backup\eql@provide@backup\fi

```

Provide native  $\LaTeX$  environment `equation` and symbolic shortcut `\[...\]` if desired:

```

6184 \ifnum\eql@provide@opt@env>\z@
6185   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6186 \fi

```

Provide amsmath equation environments if desired:

```

6187 \ifnum\eql@provide@opt@env=\tw@
6188   \eqnlinesprovide{%
6189     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6190     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6191     multlined,gathered,aligned,alignedat,%
6192     subequations}
6193 \fi

```

Provide symbolic shortcut `\<...\>` if desired:

```

6194 \ifdefined\eql@provide@opt@ang\eqnlinesprovide{ang}\fi

```

Provide equation reference `\eqref` if desired:

```

6195 \ifdefined\eql@provide@opt@eqref\eqnlinesprovide{eqref}\fi

```

Provide `cases` and `matrix` environments if desired:

```

6196 \ifdefined\eql@provide@opt@matrix
6197   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6198 \fi

```