

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`

2026/02/12, v0.15.1

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

Abstract

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

Contents

1	Implementation	3
2	General Support	3
2.1	Debugging Messages	4
2.2	Supporting Definitions	4
2.3	Dollardollar Abstraction	4
2.4	Look-Ahead in Alignment	5
2.5	Error Messages	7
2.6	<code>amsmath</code> Integration	8
2.7	PDF Tagging Support	8
2.8	Key-Value Processing	10
2.9	Scanning the Equation Body	12
3	Parameters and Registers	16
3.1	Parameters	17
3.2	Registers	20
3.3	Hooks	22
4	Features	22
4.1	Punctuation	22

4.2	Math Classes at Alignment	25
4.3	Framed Cells	26
4.4	Single-Line Composition	27
4.5	Alternative Content Description	29
5	Equation Numbering	29
5.1	Supporting Definitions	30
5.2	Schemes	31
5.3	Interface	34
5.4	Integration	39
5.5	Positioning	42
5.6	Component Display	45
5.7	Tag Composition	48
6	Subequation Numbering	51
6.1	Definitions	51
6.2	Environment	52
6.3	Subequation Scheme	54
7	Display Equations Support	54
7.1	Display Breaks	54
7.2	Explicit Vertical Space	56
7.3	Default Vertical Spacing	57
7.4	Entry and Exit	58
7.5	Stack	64
8	Multi-Line Support	65
8.1	Measure Support	65
8.2	Line Breaks	67
8.3	Intertext	69
8.4	Line Marks	71
9	Column Placement	72
9.1	Supporting Definitions	72
9.2	Shape Schemes	73
9.3	Width Data	76
9.4	Best Line Selection	80
9.5	Tag Margin	81
9.6	Single Column	81
9.7	Multiple Columns	83
10	Single Column Arrangement	88
10.1	Supporting Definitions	88
10.2	Arrangement Methods	89
10.3	Central Alignment	90
10.4	Left Alignment	92
10.5	Right Alignment	93
11	Equations Box Environment	94
11.1	Line Breaks	94
11.2	Column Breaks	96
11.3	Lines Mode	98
11.4	Columns Mode	99
11.5	Cases Mode	100
11.6	Main	102

11.7 Options Processing	104
11.8 Environment	105
12 Single-Line Equation	107
12.1 Native Mode	107
12.2 Print	108
13 Multi-Line with Single Column	109
13.1 Measure	109
13.2 Column Placement	111
13.3 Print	111
14 Multi-Line with Multiple Columns	112
14.1 Support	113
14.2 Column Breaks	113
14.3 Transpose	114
14.4 Measure	116
14.5 Columns Placement	119
14.6 Print	119
15 Interface	123
15.1 Options Processing	123
15.2 Single-Line Main	125
15.3 Multi-Line Main	126
15.4 Equations Environment	128
16 Options	131
16.1 Selection Tools	131
16.2 Options Declarations	132
16.3 Parameter Presets	147
16.4 Component Selection	149
16.5 Global and Package Options	160

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@number{starting line \the\eql@row@}
18 \def\eql@verbose@msg@startline@new{starting new line}

```

2.2 Supporting Definitions

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

```

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

```

TODO: for comparison within `\ifx`

```

21 \def\eql@relax{\relax}

```

TODO: describe

```

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

```

2.3 Dollardollar Abstraction

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

```

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

```

```

31 \def\eqldollardollar@end{$$}
32 \fi

```

2.4 Look-Ahead in Alignment

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

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

```

33 \let\eq@ifnextchar@loose\kernel@ifnextchar
34 \long\def\eq@ifnextchar@tight#1#2#3{%
35   \let\reserved@d= #1%
36   \def\reserved@a{#2}%
37   \def\reserved@b{#3}%
38   \futurelet\@let@token\eq@ifnch@tight
39 }
40 \def\eq@ifnch@tight{%
41   \ifx\@let@token\reserved@d
42     \let\reserved@b\reserved@a
43   \fi
44   \reserved@b
45 }

```

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

```

46 \let\eq@atxi=@
47 \begingroup
48   \makeatother
49   \let\tmp=@%
50   \makeatletter
51   \global\let\eq@atxii\tmp
52 \endgroup

```

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

```

53 \long\def\eq@ifnextgobble@loose#1#2{\eq@ifnextchar@loose#1{\@firstoftwo{#2}}}
54 \long\def\eq@ifnextgobble@tight#1#2{\eq@ifnextchar@tight#1{\@firstoftwo{#2}}}
55 \long\def\eq@ifstar@loose#1{\eq@ifnextchar@loose*{\@firstoftwo{#1}}}
56 \long\def\eq@ifstar@tight#1{\eq@ifnextchar@tight*{\@firstoftwo{#1}}}
57 \long\def\eq@ifat@loose#1#2{\eq@ifnextgobble@loose{#1}{%
58   \eq@ifnextgobble@loose\eq@atxii{#1}{#2}}}
59 \long\def\eq@ifat@tight#1#2{\eq@ifnextgobble@tight{#1}{%
60   \eq@ifnextgobble@tight\eq@atxii{#1}{#2}}}

```

```

61 \long\def\eqL@testopt@loose#1#2{\eqL@ifnextchar@loose[{#1}{#1[{#2}]}]}%
62 \long\def\eqL@testopt@tight#1#2{\eqL@ifnextchar@tight[{#1}{#1[{#2}]}]}%
63 \long\def\eqL@teststaropt@loose#1#2#3{%
64   \eqL@ifstar@loose{\eqL@testopt@loose{#1}{#3}}{\eqL@testopt@loose{#2}{#3}}
65 \long\def\eqL@teststaropt@tight#1#2#3{%
66   \eqL@ifstar@tight{\eqL@testopt@tight{#1}{#3}}{\eqL@testopt@tight{#2}{#3}}
67 \long\def\eqL@teststaroropt@loose#1#2#3{%
68   \eqL@ifstar@loose{#1}{\eqL@testopt@loose{#2}{#3}}
69 \long\def\eqL@teststaroropt@tight#1#2#3{%
70   \eqL@ifstar@tight{#1}{\eqL@testopt@tight{#2}{#3}}
71 \long\def\eqL@gobbleopt[#1]{}
72 \long\def\eqL@gobbleoptone[#1]#2{}

```

TODO: describe

```

73 \def\eqL@testopt@default{\eqL@testopt@default}

```

TODO: describe

```

74 \let\eqL@parseopt@warn@env\@empty
75 \let\eqL@parseopt@warn@cr\@empty

```

TODO: describe

```

76 \def\eqL@parseopt@env{%
77   \let\eqL@parseopt@warn\eqL@parseopt@warn@env\eqL@parseopt}
78 \def\eqL@parseopt@cr{%
79   \let\eqL@parseopt@warn\eqL@parseopt@warn@cr\eqL@parseopt}

```

TODO: describe

```

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

```

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

```

98 \def\eqL@spbgroup{\iffalse{\fi\ifnum0=‘}\fi}

```

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

```

99 \def\eql@speggroup{\ifnum0='{\fi\iffalse}\fi}
100 \def\eql@srbgroup{\relax\iffalse{\fi\ifnum0='{}\fi}
101 \def\eql@sregroup{\let\@let\@token\relax\ifnum0='{}\fi\iffalse}\fi}

```

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

```

102 \long\def\eql@ampprotect#1#2{\eql@srbgroup#1{\eql@sregroup#2}}
103 \long\def\eql@ampprotecttwo#1#2#3{%
104   \eql@srbgroup#1{\eql@sregroup#2}{\eql@sregroup#3}}

```

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

```

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

```

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

```

116 \let\eql@ifnextchar@org\@ifnextchar
117 \let\eql@ifstar@org\@ifstar
118 \let\eql@testopt@org\@testopt
119 \def\eql@amprevert{%
120   \let\@ifnextchar\eql@ifnextchar@org
121   \let\@testopt\eql@testopt@org
122   \let\@ifstar\eql@ifstar@org
123 }
124 \def\eql@amproof{%
125   \let\@ifnextchar\eql@ifnextchar@loose@ampsafe
126   \let\@testopt\eql@testopt@loose@ampsafe
127   \let\@ifstar\eql@ifstar@loose@ampsafe
128 }

```

2.5 Error Messages

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

```

129 \def\eql@error#1{\PackageError{eqnlines}{#1}{}}
130 \def\eql@warning{\PackageWarning{eqnlines}}

```

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

```

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

```

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

```

\eql@warn@label@multiple
\eql@warn@tag@unused
\eql@warn@tag@multiple
\eql@warn@name@unused
\eql@warn@name@multiple
\eql@warn@ref@unused
\eql@warn@ref@multiple

```

```

133 \def\eql@warn@tags@unused#1#2{\eql@warning{Unused equation #1:
134     #2 will be lost}}
135 \def\eql@warn@tags@multiple#1#2#3{\eql@warning{Multiple equation #1:
136     previous #2 will be lost#3}}
137 \def\eql@warn@label@unused{\eql@warn@tags@unused{\string\label}
138     {label '\eql@tags@label'}}
139 \def\eql@warn@label@multiple#1{\eql@warn@tags@multiple{\string\label's}
140     {label '\eql@tags@label'}{ and replaced by '#1'}}
141 \def\eql@warn@name@unused{\eql@warn@tags@unused{label name}
142     {name declaration}}
143 \def\eql@warn@name@multiple{\eql@warn@tags@multiple{label names}
144     {name declaration}{}}
145 \def\eql@warn@tag@unused{\eql@warn@tags@unused{\string\tag}
146     {tag declaration}}
147 \def\eql@warn@tag@multiple{\eql@warn@tags@multiple{\string\tag's}
148     {tag declaration will be lost}{}}
149 \def\eql@warn@ref@unused{\eql@warn@tags@unused{tag label}
150     {tag label declaration}}
151 \def\eql@warn@ref@multiple{\eql@warn@tags@multiple{tag labels}
152     {tag label declaration}{}}

153 \def\eql@warn@parseopt{\eql@warning{Unknown modifier token:
154     modifier parsing stopped}}
155 \def\eql@warn@parseopt@verbose{\eql@warning{Unknown modifier token:
156     '\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/`:

```

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

```

TODO: temporary fix for development stages

```

162 \@ifpackageloaded{amsmath}{}{
163     \DeclareHookRule{package/amsmath/after}
164     {eqnlines}{after}{latex-lab-testphase-math}}

```

2.7 PDF Tagging Support

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

```

165 \let\eql@tagging@on\eql@false
166 \IfFormatAtLeastTF{2025-06-01}{%
167     \csname tag_if_active:T\endcsname{\let\eql@tagging@on\eql@true}}{}

```

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


```

168 \ifdefined\eql@tagging@on
169   \def\eql@tagging@mathsave{%
170     \UseTaggingSocket{math/luamml/save/nNn}{\displaystyle{mtd}}}
171   \def\eql@tagging@mathaddlast{%
172     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}}
173   \def\eql@tagging@tagbegin{%
174     \UseTaggingSocket{math/display/tag/begin}}
175   \def\eql@tagging@tagend{%
176     \UseTaggingSocket{math/display/tag/end}}
177   \def\eql@tagging@tagsave{%
178     \UseTaggingSocket{math/luamml/mtable/tag/save}}
179   \def\eql@tagging@tagaddbox{%
180     \setbox\z@\copy\eql@tagbox%
181     \UseTaggingSocket{math/luamml/mtable/tag/set}}
182   \def\eql@tagging@tablesaverinner{%
183     \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}}
184   \def\eql@tagging@tableaddinner{%
185     \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
186   \def\eql@tagging@tablesavelines{%
187     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{gather}}
188   \def\eql@tagging@tablesavealign{%
189     \UseExpandableTaggingSocket{math/luamml/mtable/finalize}{align}}
190   \def\eql@tagging@alignleft{%
191     \UseTaggingSocket{math/luamml/mtable/aligncol}{left}}
192   \def\eql@tagging@aligncenter{%
193     \UseTaggingSocket{math/luamml/mtable/aligncol}{center}}
194   \def\eql@tagging@alignright{%
195     \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:

```

196   \let\eql@single@doscan\eql@true
197   \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:

```

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

```

When tagging is deactivated, provide empty definitions:

```

214 \else
215   \let\eql@tagging@mathsave\@empty
216   \let\eql@tagging@mathaddlast\@empty
217   \let\eql@tagging@tagbegin\@empty
218   \let\eql@tagging@tagend\@empty
219   \let\eql@tagging@tagsave\@empty
220   \let\eql@tagging@tagaddbox\@empty
221   \let\eql@tagging@tablesaveinner\@empty
222   \let\eql@tagging@tableaddinner\@empty
223   \let\eql@tagging@tablesavelines\@empty
224   \let\eql@tagging@tablesavealign\@empty
225   \let\eql@tagging@alignleft\@empty
226   \let\eql@tagging@aligncenter\@empty
227   \let\eql@tagging@alignright\@empty
228   \let\eql@tagging@start\@empty
229   \let\eql@tagging@end\@empty
230   \let\eql@tagging@register@luamml\@gobble
231   \let\eql@tagging@register@env\@gobble
232 \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:

```

233 \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:

```

234 \def\eql@decide@relax{\eql@tmpb:=\relax}
235 \def\eql@decide@select#1#2#3#4{%
236   \def\eql@tmpa{#3}%
237   \let\eql@tmpd\@undefined
238   \@for\eql@tmpc=#4\do{%
239     \ifdefined\eql@tmpd\else
240       \edef\eql@tmpb{\noexpand\eql@tmpb:=\expandafter\@firstoftwo\eql@tmpc}%
241       \ifx\eql@tmpb\eql@decide@relax
242         \let\eql@tmpa\eql@relax
243       \fi
244       \expandafter\@for\eql@tmpb\do{%
245         \ifx\eql@tmpa\eql@tmpb
246           \edef\eql@tmpd{\unexpanded\expandafter\expandafter\expandafter{%
247             \expandafter\@secondoftwo\eql@tmpc}}%
248         \fi

```

```

249     }%
250     \fi
251 }%
252 \ifdefined\eq1@tmpd
253     \eq1@tmpd
254 \else
255     \eq1@error{undefined value '#3' for option '#2' of '#1'}%
256 \fi
257 }

```

Decide between true and false or related pairs of values:

```

258 \def\eq1@decide@true{on,true,yes,enabled}
259 \def\eq1@decide@false{off,false,no,disabled}

```

`\eq1@decide@if`

```

260 \def\eq1@decide@if#1#2#3#4#5{%
261     \eq1@decide@select{#1}{#2}{#3}{%
262         {\eq1@decide@true{#4}},%
263         {\eq1@decide@false{#5}}}%

```

`\eq1@decide@bool` Store a boolean value into a conditional register:

```

264 \def\eq1@decide@bool#1#2#3#4{%
265     \eq1@decide@if{#1}{#2}{#3}{\let#4\eq1@true}{\let#4\eq1@false}}

```

Key Declaration.

`\eq1@define@key` For convenience, we define a wrapper for keyval's `\define@key` which accepts lists of categories and keys. We prepend the prefix `eq1@` to all our categories so that we can hide it from the user in error messages:

```

266 \def\eq1@define@key#1#2{%
267     \eq1@ifnextchar@loose[%
268         {\eq1@definekey@opt{#1}{#2}}%
269         {\eq1@definekey@noopt{#1}{#2}}}%
270 }
271 \def\eq1@definekey@noopt#1#2#3{\eq1@definekey@for{#1}{#2}{#3}}
272 \def\eq1@definekey@opt#1#2[#3]#4{\eq1@definekey@for{#1}{#2}{[#3]{#4}}}
273 \def\eq1@definekey@for#1#2#3{%
274     \def\eq1@for@fn##1##2##3{\define@key{eq1@##3}{##2}{#3}}%
275     \edef\eq1@for@vara{\noexpand\eq1@for@vara:=#1}%
276     \expandafter\@for\eq1@for@vara\do{%
277         \edef\eq1@for@varb{\noexpand\eq1@for@varb:=#2}%
278         \expandafter\@for\eq1@for@varb\do{%
279             \edef\eq1@for@call##1{%
280                 \noexpand\eq1@for@fn{##1}{\eq1@for@varb}{\eq1@for@vara}}%
281                 \eq1@for@call{##1}}%
282         }%
283     }%
284 }

```

`\eq1@setkeys` Our wrapper of keyval's `\setkeys` prepends the prefix `eq1@` to the category, and it expands the list argument once:

```

285 \def\eq1@setkeys#1#2{%
286     \def\eq1@tmp{\setkeys{eq1@#1}}%
287     \expandafter\eq1@tmp\expandafter{#2}%
288 }

```

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:

```
289 \let\eql@nextopt\@empty
290 \def\eql@nextopt@process#1{%
291   \eql@setkeys{#1}\eql@nextopt
292   \let\eql@tagging@opt\eql@nextopt
293   \global\let\eql@nextopt\@empty
294 }
```

`\eqnaddopt`

```
295 \newcommand{\eqnaddopt}[1]{%
296   \ifx\eql@nextopt\@empty
297     \eql@append\eql@nextopt{#1}%
298   \else
299     \eql@append\eql@nextopt{, #1}%
300   \fi
301 }
```

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

```
302 \newcommand{\eqnlineset}[1]{%
303   \eql@setkeys{setup}{#1}%
304   \ignorespaces
305 }
```

`\eql@control@default`

```
306 \protected\def\eql@control@default{%
307   \eql@warn@here\eqncontrol
308   \@gobble
309 }
310 \let\eqncontrol\eql@control@default
```

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

```
311 \newcommand{\eql@control}[1]{%
312   \relax
313   \eql@setkeys{control}{#1}%
314   \ignorespaces
315 }
```

2.9 Scanning the Equation Body

The multi-line equation 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.

```
316 \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: **TODO**: may skip `\expandafter` before `\scantokens`?

```

317 \def\eql@scan@body@dump{\the\eql@scan@reg@}
318 \def\eql@scan@body@rescan{%
319   \expandafter\scantokens\expandafter{\the\eql@scan@reg@}}
320 \let\eql@scan@body\eql@scan@body@dump

```

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

```

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

```

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

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

```

322 \def\eql@scan@env{%
323   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@env
324   \def\eql@scan@end{\expandafter\end\expandafter{\@currenvir}}}%
325   \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`:

```

326 \edef\eql@scan@iterate{\expandafter\noexpand\curname\@currenvir\endcurname}%
327 \expandafter\let\expandafter\eql@scan@env@org\eql@scan@iterate
328 \ifdefined\eql@scan@par
329   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate
330 \else
331   \expandafter\let\eql@scan@iterate\eql@scan@env@iterate@nopar
332 \fi
333 \eql@scan@iterate
334 }

```

`\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:

```

335 \long\def\eql@scan@env@iterate#1\end#2{%
336   \edef\eql@scan@stack{%
337     \eql@scan@env@count#1\begin\end\expandafter\@gobble\eql@scan@stack}%
338   \ifx\@empty\eql@scan@stack
339     \@checkend{#2}%
340     \eql@scan@addto{#1}%

```

```

341 \expandafter\let\eql@scan@iterate\eql@scan@env@org
342 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@env
343 \expandafter\eql@scan@call
344 \else
345 \eql@scan@addto{#1\end{#2}}}%
346 \expandafter\eql@scan@iterate
347 \fi
348 }

```

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

```

349 \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.

```

350 \long\def\eql@scan@env@count#1\begin#2{%
351 \ifx\end#2\else b\expandafter\eql@scan@env@count\fi
352 }

```

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

```

353 \def\eql@scan@env@cancel{%
354 \@namedef{end\@currenvir}{\ignorespacesafterend}%
355 \eql@scan@end
356 }

```

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 `\[`:

```

357 \def\eql@scan@sqr{%
358 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@sqr
359 \def\eql@scan@end{\}%
360 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
361 \let\eql@scan@sqr@org\[%\]
362 \ifdefined\eql@scan@par
363 \let\[\eql@scan@sqr@iterate%\]
364 \else
365 \let\[\eql@scan@sqr@iterate@nopar%\]
366 \fi
367 \[%\]
368 }

```

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

```

369 \long\def\eql@scan@sqr@iterate#1\[%\]
370 \edef\eql@scan@stack{%
371 \eql@scan@sqr@count#1\[\]\expandafter\@gobble\eql@scan@stack}%
372 \ifx\@empty\eql@scan@stack
373 \let\[\eql@scan@sqr@org%\]
374 \eql@scan@addto{#1}%
375 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@sqr
376 \expandafter\eql@scan@call
377 \else

```

```

378 \eql@scan@addto{#1\}}%
379 \expandafter\[%\]
380 \fi
381 }

```

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

```

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

```

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

```

383 \long\def\eql@scan@sqr@count#1\[#2{\% \]
384 \ifx\]#2\else b\expandafter\eql@scan@sqr@count\fi
385 }

```

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

```

386 \def\eql@scan@sqrang@cancel{\ignorespaces}

```

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

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

```

387 \def\eql@scan@ang{%
388 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@scan@ang
389 \def\eql@scan@end{\>}%
390 \eql@scan@reg@{\}\def\eql@scan@stack{b}%
391 \let\eql@scan@ang@org\<\>
392 \ifdefined\eql@scan@par
393 \let\<\eql@scan@ang@iterate%\>
394 \else
395 \let\<\eql@scan@ang@iterate@nopar%\>
396 \fi
397 \<\>
398 }

```

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

```

399 \long\def\eql@scan@ang@iterate#1\>{%
400 \edef\eql@scan@stack{%
401 \eql@scan@ang@count#1\<\>\expandafter\@gobble\eql@scan@stack}%
402 \ifx\@empty\eql@scan@stack
403 \let\<\eql@scan@ang@org%\>
404 \eql@scan@addto{#1}%
405 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@scan@ang
406 \expandafter\eql@scan@call
407 \else
408 \eql@scan@addto{#1\>}%
409 \expandafter\<\>
410 \fi
411 }

```

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

```

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

```

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

```

413 \long\def\eql@scan@ang@count#1\<#2{%\>
414   \ifx\>#2\else b\expandafter\eql@scan@ang@count\fi
415 }

```

Interface. TODO: describe

`\eql@sqr@open` If already in math mode, ignore the enclosed contents.

```

416 \protected\def\eql@sqr@open{%
417   \ifmmode
418     \expandafter\eql@sqr@cancel
419   \else
420     \expandafter\eql@equations@sqr@open
421   \fi
422 }

```

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

```

423 \protected\def\eql@sqr@close{%
424   \eql@error{'\string\]' may only close '\string\[']}%\]
425 }

```

`\eql@sqr@cancel`

```

426 \def\eql@sqr@cancel{%
427   \eql@error@mathmode{'\string\ [...\string\]]}%
428   \let\eql@scan@call\eql@scan@sqrang@cancel
429   \eql@scan@sqr
430 }

```

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

```

431 \protected\def\eql@ang@open{%
432   \ifmmode
433     \expandafter\eql@box@ang@open
434   \else
435     \expandafter\eql@equations@ang@open
436   \fi
437 }

```

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

```

438 \protected\def\eql@ang@close{%
439   \eql@error{'\string\>' may only close '\string\<']}%\>
440 }

```

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:

```
441 \let\eql@tagsleft\eql@false
```

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

```
442 \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`:

```
443 \def\eql@layoutleftmargin{\leftmargini}
444 \def\eql@layoutleftmarginmax{.5\maxdimen}
445 \def\eql@layoutleftmarginmin{\z@}
```

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

```
446 \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@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

```
447 \newdimen\eql@tagmargin@
448 \let\eql@tagmargin@val\@undefined
449 \newdimen\eql@tagmargin@ratio@
450 \eql@tagmargin@ratio@\p@
451 \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:

```
452 \newdimen\eql@indent@
453 \def\eql@indent@val{2em}
```

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

```
454 \newdimen\eql@paddingleft@
455 \newdimen\eql@paddingright@
456 \let\eql@paddingleft@val\@undefined
457 \let\eql@paddingright@val\@undefined
```

`\eqldisplay@linewidth` **TODO:** describe

`\eqldisplay@marginleft`

`\eqldisplay@marginright`

```

458 \let\eqldisplay@linewidth\@undefined
459 \let\eqldisplay@marginleft\@undefined
460 \let\eqldisplay@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`

```

461 \def\eql@box@colsep{2em}
462 \def\eql@box@shortsep{1em}
463 \def\eql@box@condsep{\eql@box@shortsep}

```

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

`\eql@break@line@shortsep`

`\eql@break@col@sep`

`\eql@break@col@shortsep`

```

464 \def\eql@break@line@sep{2em minus 1em}
465 \def\eql@break@line@shortsep{1em}
466 \def\eql@break@col@sep{2em minus 1em}
467 \def\eql@break@col@shortsep{1em}

```

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

```

468 \let\eql@spread@reset\eql@false
469 \def\eql@spread@val{\jot}
470 \newdimen\eql@spread@

```

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

```

471 \newdimen\eql@tagfuzz@
472 \eql@tagfuzz@16sp\relax

```

`\eqldisplay@height` An equation will appear to the surrounding text with a fixed apparent height and depth

`\eqldisplay@depth` specified by `\eqldisplay@height` and `\eqldisplay@depth`, respectively:

```

473 \def\eqldisplay@height\@undefined
474 \def\eqldisplay@depth\@undefined

```

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

```

475 \def\eql@skip@mode@short{2}

476 \def\eql@skip@mode@cont@above{2}
477 \def\eql@skip@mode@cont@below{0}

478 \def\eql@skip@mode@par@above{3}
479 \def\eql@skip@mode@par@below{0}

480 \def\eql@skip@mode@top@above{4}
481 \def\eql@skip@mode@top@below{0}

482 \newcount\eql@skip@mode@leave@
483 \let\eql@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) 484 \newcount\eq@skip@mode@above@
\mode@below@ (counter) 485 \newcount\eq@skip@mode@below@
486 \let\eq@skip@force@above@\undefined
487 \let\eq@skip@force@below@\undefined
488 \let\eq@skip@custom@above@\undefined
489 \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`:

```

490 \def\eq@skip@long@above{\abovedisplayskip}
491 \def\eq@skip@long@below{\belowdisplayskip}
492 \def\eq@skip@short@above{\abovedisplaysshortskip}
493 \def\eq@skip@short@below{\belowdisplaysshortskip}
494 \def\eq@skip@cont@above{\eq@skip@short@above}
495 \def\eq@skip@cont@below{\eq@skip@short@below}
496 \def\eq@skip@par@above{\eq@skip@long@above}
497 \def\eq@skip@par@below{\eq@skip@long@below}
498 \def\eq@skip@top@above{\eq@skip@long@above}
499 \def\eq@skip@top@below{\eq@skip@long@below}
500 \def\eq@skip@med@above{\abovedisplayskip/2}
501 \def\eq@skip@med@below{\belowdisplayskip/2}
502 \def\eq@skip@tag@above{\z@skip}
503 \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:

```

504 \newdimen\eq@colsepmin@
505 \def\eq@colsepmin@val{1em}
506 \def\eq@colsepmax@val{.5\maxdimen}

```

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

```

507 \newdimen\eq@tagwidthmin@
508 \eq@tagwidthmin@\z@

```

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

```

509 \newdimen\eq@tagsepmin@
510 \def\eq@tagsepmin@val{.5\fontdimen6\textfont\tw@}

```

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

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

```
\eql@equations@ang@opt
\eq1@box@ang@opt
511 \def\eq1@equations@sqr@opt{equation}
512 \def\eq1@equations@ang@opt{columns}
513 \def\eq1@box@ang@opt{columns}
```

Multi-Line Align Mode.

```
514 \let\eq1@columns@fulllength\eq1@false
```

3.2 Registers

TODO: describe

General. **TODO:** describe

```
515 \newcount\eq1@count@
516 \newdimen\eq1@dimen@
517 \newskip\eq1@skip@
```

TODO: describe

```
518 \let\eq1@display@container\@empty
```

`\eq1@cellbox@` (*box*) The box `\eq1@cellbox@` holds the present alignment component and `\eq1@tagbox@` the

`\eq1@tagbox@` (*box*) tag for the present line. The corresponding dimensions `\eq1@cellwidth@` and

`\eq1@cellwidth@` (*dimen*) `\eq1@tagwidth@` hold their widths. `\eq1@prevwidth@` holds the width of the previous

`\eq1@prevwidth@` (*dimen*) alignment component: **TODO:** adjust

`\eq1@tagwidth@` (*dimen*)

`\eq1@prevdepth@` (*dimen*)

`\eq1@prevgraf@` (*counter*)

```
519 \newbox\eq1@cellbox@
520 \newbox\eq1@tagbox@
521 \newdimen\eq1@cellwidth@
522 \newdimen\eq1@prevwidth@
523 \newdimen\eq1@tagwidth@
524 \newdimen\eq1@prevdepth@
525 \newcount\eq1@prevgraf@
```

`\eq1@totalwidth@` (*dimen*)

`\eq1@tagwidth@max@` (*dimen*)

`\eq1@totalheight@` (*dimen*)

```
526 \newdimen\eq1@totalwidth@
527 \newdimen\eq1@tagwidth@max@
528 \newdimen\eq1@totalheight@
529 \newdimen\eq1@topheight@
530 \newdimen\eq1@bottomdepth@
```

`\eq1@line@height@` (*dimen*) The dimension registers `\eq1@line@height@` and `\eq1@line@depth@` keep track of the

`\eq1@line@depth@` (*dimen*) height and depth of the present line in an alignment:

```
531 \newdimen\eq1@line@height@
532 \newdimen\eq1@line@depth@
```

`\eq1@line@width@` (*dimen*)

`\eq1@line@avail@` (*dimen*)

`\eq1@line@pos@` (*dimen*)

`\eq1@widthsep@` (*counter*)

`\eq1@availsep@` (*counter*)

`\eq1@line@possep@` (*counter*)

`\eq1@line@offset@` (*dimen*)

`\eq1@prevdepth@` (*dimen*)

`\eq1@interline@` (*dimen*)

```
533 \newdimen\eq1@line@width@
534 \newdimen\eq1@line@avail@
535 \newdimen\eq1@line@pos@
```

```

536 \newcount\eql@line@availsep@
537 \newcount\eql@line@widthsep@
538 \newcount\eql@line@possep@
539 \newdimen\eql@line@offset@
540 \newdimen\eql@line@prevdepth@
541 \newdimen\eql@line@interline@

```

Rows and Columns.

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

`\eql@tagrows@` (*counter*)

```

542 \newcount\eql@row@
543 \newcount\eql@totalrows@
544 \newcount\eql@tagrows@

```

`\eql@column@`

`\eql@totalcolumns@`

```

545 \newcount\eql@column@
546 \newcount\eql@totalcolumns@

```

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

```

547 \newdimen\eql@colsep@

```

`\eql@intercolumns@` (*counter*)

```

548 \newcount\eql@intercolumns@

```

Vertical Spacing Adjustments.

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

```

549 \newdimen\eql@display@firstavail@
550 \def\eql@display@firstavail@set#1{%
551   \ifnum\eql@row@=\@ne
552     \global\eql@appendexpand\eql@display@container{%
553       \eql@display@firstavail@\the#1\relax}%
554   \fi
555 }

```

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

```

556 \newdimen\eql@display@aboveextend@
557 \newdimen\eql@display@belowextend@

```

Shared Registers.

`\ifmeasuring@` (*bool*) All display environments get typeset twice – once during a “measuring” phase and then again during a “production” phase. We reuse the original `amsmath` definition

`\ifmeasuring@` to determine which case we're in, so we and other packages may take appropriate action. It does not hurt to define this conditional in any case. We should tell `hyperref` about measuring processes as we're not `amsmath` and not being catered for:

```
558 \ifdefined\measuring@true\else
559   \expandafter\newif\csname ifmeasuring@\endcsname
560 \fi
561 \AddToHook{package/hyperref/after}{
562   \ifdefined\Hy@ifnotmeasuring
563     \renewcommand\Hy@ifnotmeasuring[1]{\ifmeasuring@\else#1\fi}
564   \fi
565 }
```

`\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:

```
566 \ifdefined\@displaytrue\else
567   \expandafter\newif\csname if@display\endcsname
568   \everydisplay\expandafter{\the\everydisplay\@displaytrue}
569 \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 \LaTeX 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

```
570 \let\eql@hook@blockbefore\@empty
571 \let\eql@hook@blockafter\@empty
572 \let\eql@hook@blockin\@empty
573 \let\eql@hook@blockout\@empty
574 \let\eql@hook@linein\@empty
575 \let\eql@hook@lineout\@empty
576 \let\eql@hook@colin\@empty
577 \let\eql@hook@colout\@empty
578 \let\eql@hook@eqin\@empty
579 \let\eql@hook@eqout\@empty
580 \let\eql@hook@innerleft\@empty
581 \let\eql@hook@innerright\@empty
582 \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` 583 \let\eql@punct@col\@empty

```

584 \let\eqlopunct@line\@undefined
585 \let\eqlopunct@block\@undefined
586 \let\eqlopunct@next\@undefined
587 \let\eqlopunct@top\@undefined

```

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

```

588 \let\eqlopunct@class\@empty
589 \let\eqlopunct@sep\@empty

```

```

\eqlopunct@top@set TODO: describe
\eqlopunct@top@stop
\eqlopunct@top@reset 590 \def\eqlopunct@top@set{%
591   \let\eqlopunct@top\eqlopunct@block}
592 \def\eqlopunct@top@stop{%
593   \let\eqlopunct@top\relax}
594 \def\eqlopunct@top@reset{%
595   \let\eqlopunct@top\@undefined}

```

`\eqlopunct@set` **TODO:** describe

```

596 \def\eqlopunct@tilde{~}
597 \def\eqlopunct@set#1#2{%
598   \def#1{#2}%
599   \ifx#1\eqlopunct@relax
600     \let#1\@undefined
601   \fi
602   \ifx#1\eqlopunct@tilde
603     \let#1\@empty
604   \fi
605 }
606 \def\eqlopunct@clear{%
607   \let\eqlopunct@col\@empty
608   \let\eqlopunct@line\@empty
609   \let\eqlopunct@block\@empty
610 }
611 \def\eqlopunct@next@clear{\let\eqlopunct@next\@empty}

```

Set the punctuation 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

```

612 \def\eqlopunct@addopt{%
613   \eqlopunct@ifstar@tight\eqlopunct@addopt@relax\eqlopunct@addopt@set}
614 \def\eqlopunct@addopt@set#1{\eqnaddopt{punct={#1}}\ignorespaces}
615 \def\eqlopunct@addopt@relax{\eqnaddopt{punct*}\ignorespaces}

```

TODO: describe

```

616 \def\eqlopunct@setnext{%
617   \eqlopunct@ifstar@tight\eqlopunct@setnext@relax\eqlopunct@setnext@set}
618 \def\eqlopunct@setnext@set#1{%
619   \eqlopunct@set\eqlopunct@next{#1}%
620   \ifdefined\eqlopunct@next\else\let\eqlopunct@next\relax\fi
621   \ignorespaces}
622 \def\eqlopunct@setnext@relax{\let\eqlopunct@next\relax\ignorespaces}

```

`\eqnpunct` **TODO:** describe

623 \let\eqnpunct\eql@punct@adopt

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

```

624 \def\eql@punct@fill@next#1{%
625   \ifdefined\eql@punct@next
626     \ifx\eql@punct@next\relax
627       \let\eql@punct@next\@undefined
628     \fi
629   \else
630     \ifx\eql@punct@top\relax\else
631       \let\eql@punct@next#1%
632     \fi
633   \fi
634 }
```

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

```

635 \def\eql@punct@output@next{%
636   \ifx\eql@punct@next\@empty\else
637     \ifmmode\eql@punct@class\fi
638     \eql@punct@sep
639     \eql@punct@next
640   \fi
641   \let\eql@punct@next\@undefined
642 }
```

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

```

643 \def\eql@punct@print@next{%
644   \ifdefined\eql@punct@next
645     \eql@punct@output@next
646   \fi
647 }
```

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

```

648 \def\eql@punct@apply@next{%
649   \ifdefined\eql@punct@next
650     \eql@punct@output@next
651     \eql@punct@top@stop
652   \fi
653 }
```

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

```

654 \def\eql@punct@print@col{%
655   \eql@punct@fill@next\eql@punct@col
656   \eql@punct@print@next
657 }
```

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

```

658 \def\eql@punct@apply@col{%
659   \eql@punct@fill@next\eql@punct@col
660   \eql@punct@apply@next
661 }
```

Output the punctuation for the present line unless disabled:

`\eql@punct@print@line`

```
662 \def\eql@punct@print@line{%
663   \eql@punct@fill@next\eql@punct@line
664   \eql@punct@print@next
665 }
```

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

`\eql@punct@apply@line`

```
666 \def\eql@punct@apply@line{%
667   \eql@punct@fill@next\eql@punct@line
668   \eql@punct@apply@next
669 }
```

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

```
670 \def\eql@punct@apply@block{%
671   \eql@punct@fill@next\eql@punct@block
672   \eql@punct@apply@next
673 }
```

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

```
674 \def\eql@punct@apply@top{%
675   \eql@punct@fill@next\eql@punct@top
676   \eql@punct@print@next
677   \global\eql@punct@top@stop
678 }
```

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

```
679 \newcommand{\eqnpunctapply}{\ifmmode\else\unskip\fi\eql@punct@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 `\eql@class@innerlead` and `\eql@class@innerright` depending on whether the left part of the aligned column is empty:

```
680 \def\eql@class@innerright@sel@{%
681   \ifdim\eql@prevwidth@=\z@
682     \eql@class@innerlead
683   \else
684     \eql@class@innerright
685   \fi
686 }
```

`\@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`

```
687 \def\eql@class@innerleft@set#1{%
688   \def\eql@class@innerleft{#1}%
689 }
```

```

690 \def\eq@class@innerright@set#1{%
691   \def\eq@class@innerright{#1}%
692   \let\eq@class@innerright@sel\eq@class@innerright
693 }
694 \def\eq@class@innerlead@set#1{%
695   \def\eq@class@innerlead{#1}%
696   \let\eq@class@innerright@sel\eq@class@innerright@sel@
697 }

698 \def\eq@class@rel@symb{=}
699 \def\eq@class@rel@cont#1{\eq@class@rel@start{#1}\mathclose{}}
700 \def\eq@class@rel@start#1{\mathrel{\phantom{#1}}}
701 \def\eq@class@rel@amp{&}
702 \def\eq@class@rel@amprelax{&\relax}
703 \def\eq@class@rel@relaxamp{\relax&}
704 \def\eq@class@rel@ordamp{&}
705 \def\eq@class@rel@make#1{%
706   \ifdefined\eq@class@rel@after
707     \def\eq@tmp{&#1}%
708     \ifx\eq@tmp\eq@class@rel@amp
709       \def\eq@tmp{&\eq@class@rel@cont\eq@class@rel@symb}%
710     \else\ifx\eq@tmp\eq@class@rel@amprelax
711       \def\eq@tmp{&\eq@class@rel@start\eq@class@rel@symb}%
712     \fi\fi
713   \else
714     \def\eq@tmp{#1}%
715     \ifx\eq@tmp\eq@class@rel@relaxamp
716       \def\eq@tmp{&\mathclose{}\mathopen{}\mathrel{}}%
717     \fi
718     \ifx\eq@tmp\eq@class@rel@ordamp
719       \def\eq@tmp{&\mathclose{}}}%
720     \fi
721   \fi
722   \eq@tmp
723 }

```

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

```

724 \def\eq@class@ampeq{%
725   \eq@class@innerleft@set{}%
726   \eq@class@innerright@set{}}%
727   \let\eq@class@rel@after\eq@true
728 }
729 \def\eq@class@eqamp{%
730   \eq@class@innerleft@set{}%
731   \eq@class@innerright@set{\mathrel{}}%
732   \eq@class@innerlead@set{}}%
733   \let\eq@class@rel@after\eq@false
734 }
735 \eq@class@ampeq

```

4.3 Framed Cells

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

```

736 \let\eq@frame@cmd\@undefined
737 \newdimen\eq@frame@margin@

```

```

738 \def\eql@frame@set[#1]{%
739   \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}
740 \protected\def\framecell{\eql@testopt@tight@ampsafe\eql@frame@set\fbbox}
741 \def\eql@frame@measure{%
742   \setbox\z@\hbox{\eql@frame@cmd}}%
743   \eql@frame@margin0.5\wd\z@
744 }
745 \def\eql@frame@print{%
746   \setbox\eql@cellbox@\hbox{%
747     \eql@frame@cmd{\unhbox\eql@cellbox@}%
748   }%
749 }
750 \def\eql@frame@adjust{%
751   \setbox\eql@cellbox@\hbox{%
752     \eql@frame@cmd{%
753       \unhbox\eql@cellbox@
754       \unkern
755       \unskip
756     }%
757     \hfil
758     \kern\z@
759   }%
760 }

```

4.4 Single-Line Composition

TODO: describe

```

\eql@break@line
\eql@break@col
761 \def\eql@break@line{%
762   \let\eql@break@sep\eql@break@line@sep
763   \let\eql@break@shortsep\eql@break@line@shortsep
764   \let\eql@break@print\eql@punct@print@line
765   \let\eql@punct@term\eql@false
766   \let\eql@class@rel@composed\@undefined
767   \eql@ampprotect\eql@break@test\eql@break@process}
768 \def\eql@break@col{%
769   \let\eql@break@sep\eql@break@col@sep
770   \let\eql@break@shortsep\eql@break@col@shortsep
771   \let\eql@break@print\eql@punct@print@col
772   \let\eql@punct@term\eql@false
773   \let\eql@class@rel@composed\@undefined
774   \eql@ampprotect\eql@break@test\eql@break@process}
775 \def\eql@break@cr{%
776   \let\eql@break@sep\eql@break@line@sep
777   \let\eql@break@shortsep\eql@break@line@shortsep
778   \let\eql@break@print\eql@punct@print@line
779   \let\eql@punct@term\eql@false
780   \let\eql@class@rel@composed\@undefined
781   \eql@ampprotect\eql@break@cr@test\eql@break@process}
782 \def\eql@break@amp{%
783   \eql@ampprotecttwo\eql@break@amp@testescape
784   \eql@amp@org\eql@break@process}
785 \def\eql@break@amp@testescape#1#2{%
786   \eql@ifnextgobble@tight/{#1}{\eql@break@amp@testnoescape{#2}}}
787 \def\eql@break@amp@testnoescape#1{%
788   \relax

```

```

789 \let\eql@break@sep\eql@break@col@sep
790 \let\eql@break@shortsep\eql@break@col@shortsep
791 \let\eql@break@print\eql@punct@print@col
792 \let\eql@punct@term\eql@false
793 \let\eql@class@rel@composed\@undefined
794 \eql@break@amp@test{#1}}

```

TODO: describe

```

795 \def\eql@break@test@setopt{\let\eql@break@test\eql@break@testopt}
796 \def\eql@break@test@setall{\let\eql@break@test\eql@break@testall}
797 \def\eql@break@cr@test@setopt{\let\eql@break@cr@test\eql@break@testopt}
798 \def\eql@break@cr@test@setall{\let\eql@break@cr@test\eql@break@testall}
799 \def\eql@break@amp@test@setopt{\let\eql@break@amp@test\eql@break@testopt}
800 \def\eql@break@amp@test@setall{\let\eql@break@amp@test\eql@break@testall}

```

`\eql@break@testopt` **TODO:** describe

```

801 \def\eql@break@testopt#1{\eql@ifstar@tight
802   {\let\eql@break@sep\eql@break@shortsep#1}{\eql@break@testopt@arg{#1}}}
803 \def\eql@break@testopt@arg#1{\eql@ifnextchar@tight [%
804   {\eql@break@testopt@set{#1}}{#1}}
805 \def\eql@break@testopt@set#1[#2]{\def\eql@break@sep{#2}#1}

```

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

`@break@testall@parse`

```

806 \def\eql@break@testall{\eql@parseopt@cr\eql@break@testall@parse}
807 \def\eql@break@testall@parse{%
808   \ifx\eql@parseopt@token[%
809     \let\eql@parseopt@next\eql@break@parse@val
810   \fi
811   \ifx\eql@parseopt@token*%
812     \let\eql@parseopt@next\eql@break@parse@star
813   \fi
814   \ifx\eql@parseopt@token.%
815     \let\eql@parseopt@next\eql@parseopt@punctpass
816   \fi
817   \ifx\eql@parseopt@token,%
818     \let\eql@parseopt@next\eql@parseopt@punctpass
819   \fi
820   \ifx\eql@parseopt@token~%
821     \let\eql@parseopt@next\eql@parseopt@punctpass
822   \fi
823   \ifx\eql@parseopt@token'%
824     \let\eql@parseopt@next\eql@parseopt@punctnext
825   \fi
826   \ifx\eql@parseopt@token!%
827     \let\eql@parseopt@next\eql@parseopt@punctterm
828   \fi
829   \ifx\eql@parseopt@token/%
830     \let\eql@parseopt@next\eql@parseopt@punctclear
831   \fi
832   \ifx\eql@parseopt@token=%
833     \let\eql@parseopt@next\eql@parseopt@relsyb
834   \fi
835   \ifx\eql@parseopt@token;%
836     \let\eql@parseopt@next\eql@parseopt@relcont
837   \fi
838   \ifx\eql@parseopt@token:%

```

```

839 \let\eql@parseopt@next\eql@parseopt@relstart
840 \fi
841 \ifx\eql@parseopt@token|
842 \let\eql@parseopt@next\eql@parseopt@relord
843 \fi
844 \ifx\eql@parseopt@token?
845 \let\eql@parseopt@next\eql@break@parse@rel
846 \fi
847 }
848 \def\eql@break@parse@val[#1]{%
849 \def\eql@break@sep{#1}\eql@parseopt@peek}
850 \def\eql@break@parse@star#1{%
851 \let\eql@break@sep\eql@break@shortsep\eql@parseopt@peek}
852 \def\eql@break@parse@rel#1#2{%
853 \def\eql@class@rel@composed{#2}\eql@parseopt@end}

```

`\eql@break@process`

```

854 \def\eql@break@process{%
855 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
856 \ifdefined\eql@class@rel@composed
857 \eql@class@rel@composed
858 \else
859 \eql@break@print
860 \hspace{\glueexpr\eql@break@sep\relax}%
861 \fi
862 }

```

`\eql@break@join`

```

863 \def\eql@break@join{\eql@srbgroup
864 \eql@ifstar@tight
865 {\eql@break@join@opt[\eql@break@col@shortsep]}%
866 {\eql@testopt@tight\eql@break@join@opt\eql@break@col@sep}}
867 \def\eql@break@join@opt[#1]#2{\eql@sregroup%
868 \hspace{\glueexpr#1\relax}#2\hspace{\glueexpr#1\relax}}

```

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

`\eqnbreak`

`\eqnjoin`

```

869 \newcommand{\eqnsep}{\eql@break@col}
870 \newcommand{\eqnbreak}{\eql@break@line}
871 \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

```

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

```

5 Equation Numbering

TODO: describe

5.1 Supporting Definitions

Parameters.

```
873 \let\eql@tags@autolabel\eql@false
874 \let\eql@tags@autotag\eql@true
875 \let\eql@tags@warn\eql@true

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

877 \let\eql@tagpos@doconvert\eql@false

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

Registers.

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

880 \let\eql@numbering@active\eql@false
881 \let\eql@numbering@multi\eql@true

882 \let\eql@tags@container\@undefined
883 \def\eql@tags@container@clear{%
884   \let\eql@tags@label\@undefined
885   \let\eql@tags@name\@undefined
886   \let\eql@tags@tag\@undefined
887   \let\eql@tags@ref\@undefined
888   \let\eql@tags@anchor\@empty
889   \eql@tagpos@shift@z@
890   \eql@tagpos@smashup@z@
891   \eql@tagpos@smashdown@z@
892   \let\eql@tagpos@reserve\eql@true
893 }

894 \let\eql@tags@label\@undefined
895 \let\eql@tags@name\@undefined
896 \let\eql@tags@tag\@undefined
897 \let\eql@tags@ref\@undefined
898 \let\eql@tags@frame@cmd\@firstofone
```

ags@glabel@ (*counter*)

```
899 \newcount\eql@tags@glabel@
900 \eql@tags@glabel@z@
901 \def\eql@tags@glabel{equation.eql-\the\eql@tags@glabel@}
902 \def\eql@tags@glabel@step{global\advance\eql@tags@glabel@\@ne}

903 \let\eql@tagpos@continuous\eql@false

904 \newcount\eql@tagpos@row@
905 \newcount\eql@tagpos@prevrow@
906 \newdimen\eql@tagpos@shift@
907 \newdimen\eql@tagpos@smashup@
908 \newdimen\eql@tagpos@smashdown@
909 \newdimen\eql@tagpos@current@
910 \newdimen\eql@tagpos@plain@
911 \newdimen\eql@tagpos@raised@
912 \newdimen\eql@tagpos@target@
913 \newdimen\eql@tagpos@headroom@
914 \newdimen\eql@tagpos@footroom@
```

5.2 Schemes

TODO: describe

Table.

```
915 \def\eql@numbering@tab@sub{sub}
916 \def\eql@numbering@tab@all{all}
917 \def\eql@numbering@tab@first{first}
918 \def\eql@numbering@tab@last{last}
919 \def\eql@numbering@tab@in{in}
920 \def\eql@numbering@tab@out{out}
921 \def\eql@numbering@tab@middle{middle}
922 \def\eql@numbering@tab@best{best}
923 \def\eql@numbering@tab@here{here}
924 \def\eql@numbering@tab@top{top}
925 \def\eql@numbering@tab@bottom{bottom}
926 \def\eql@numbering@tab@center{center}
927 \def\eql@numbering@tab@centerone{centerone}
928 \def\eql@numbering@tab@median{median}
929 \def\eql@numbering@tab@baseline{baseline}

930 \let\eql@numbering@mode\eql@numbering@tab@all
931 \let\eql@numbering@mode@multi\eql@numbering@tab@all
932 \let\eql@numbering@mode@single\eql@numbering@tab@out
```

TODO: describe

```
933 \let\eql@numbering@tab@subeq\eql@numbering@tab@sub
934 \let\eql@numbering@tab@subequation\eql@numbering@tab@sub
935 \let\eql@numbering@tab@subequations\eql@numbering@tab@sub
936 \let\eql@numbering@tab@mid\eql@numbering@tab@middle
937 \let\eql@numbering@tab@outside\eql@numbering@tab@out
938 \let\eql@numbering@tab@inside\eql@numbering@tab@in
939 \let\eql@numbering@tab@within\eql@numbering@tab@in
940 \let\eql@numbering@tab@opt\eql@numbering@tab@best
941 \let\eql@numbering@tab@optimal\eql@numbering@tab@best
942 \let\eql@numbering@tab@pick\eql@numbering@tab@here
943 \let\eql@numbering@tab@med\eql@numbering@tab@median
944 \eql@letcs{eql@numbering@tab@center*}\eql@numbering@tab@baseline
945 \eql@letcs{eql@numbering@tab@center!}\eql@numbering@tab@centerone
```

TODO: describe

```
946 \let\eql@numbering@tab@a\eql@numbering@tab@all
947 \let\eql@numbering@tab@s\eql@numbering@tab@sub
948 \let\eql@numbering@tab@f\eql@numbering@tab@first
949 \let\eql@numbering@tab@l\eql@numbering@tab@last
950 \let\eql@numbering@tab@o\eql@numbering@tab@out
951 \let\eql@numbering@tab@i\eql@numbering@tab@in
952 \let\eql@numbering@tab@h\eql@numbering@tab@here
953 \let\eql@numbering@tab@t\eql@numbering@tab@top
954 \let\eql@numbering@tab@b\eql@numbering@tab@bottom
955 \let\eql@numbering@tab@c\eql@numbering@tab@center
956 \let\eql@numbering@tab@m\eql@numbering@tab@median
957 \eql@letcs{eql@numbering@tab@+}\eql@numbering@tab@best
958 \eql@letcs{eql@numbering@tab@m*}\eql@numbering@tab@middle
959 \eql@letcs{eql@numbering@tab@c*}\eql@numbering@tab@baseline
960 \eql@letcs{eql@numbering@tab@c!}\eql@numbering@tab@centerone
```

Implementations. **TODO:** describe

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

TODO: describe

```
962 \def\eql@numbering@init@sub{%
963   \let\eql@numbering@multi\eql@true
964   \ifdefined\eql@subequations@active
965     \let\eql@numbering@mode\eql@numbering@tab@all
966   \else
967     \let\eql@numbering@subeq@use\eql@true
968   \fi
969 }
```

```
970 \def\eql@numbering@init@first{\eql@tagpos@row@{\@ne}
971 \def\eql@numbering@init@last{\eql@tagpos@row@{\@MM}
972 \def\eql@numbering@init@here{\eql@tagpos@row@{\m@ne}
```

TODO: describe

```
973 \def\eql@numbering@init@in{%
974   \ifdefined\eql@tagsleft
975     \eql@numbering@init@last
976   \else
977     \eql@numbering@init@first
978   \fi
979 }
```

TODO: describe

```
980 \def\eql@numbering@init@out{%
981   \ifdefined\eql@tagsleft
982     \eql@numbering@init@first
983   \else
984     \eql@numbering@init@last
985   \fi
986 }
```

TODO: describe

```
987 \def\eql@tagpos@eval@middle{%
988   \ifnum\eql@tagpos@row@=\z@
989     \eql@tagpos@row@\numexpr(\eql@totalrows@
990       +\ifdefined\eql@tagsleft\z@\else\@ne\fi)/\tw@\relax
991   \fi
992 }
```

TODO: describe

```
993 \def\eql@tagpos@eval@best{%
994   \ifnum\eql@tagpos@row@=\z@
995     \let\eql@numbering@best@use\eql@true
996     \eql@numbering@init@out
997   \fi
998 }
```

TODO: describe

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

TODO: describe


```

1000 \let\eql@numbering@init@top\eql@numbering@init@continuous
1001 \def\eql@tagpos@eval@top{%
1002   \eql@tagpos@current@\z@
1003 }

```

TODO: describe

```

1004 \let\eql@numbering@init@bottom\eql@numbering@init@continuous
1005 \def\eql@tagpos@eval@bottom{%
1006   \eql@tagpos@current@\dimexpr\eql@totalheight@
1007     -\eql@tagheight@block@-\eql@tagdepth@block@\relax
1008 }

```

TODO: describe

```

1009 \let\eql@numbering@init@center\eql@numbering@init@continuous
1010 \def\eql@tagpos@eval@center{%
1011   \ifnum\eql@totalrows@=\@ne
1012     \eql@tagpos@row@\@ne
1013   \fi
1014   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1015     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1016 }

```

TODO: describe

```

1017 \let\eql@numbering@init@centerone\eql@numbering@init@continuous
1018 \def\eql@tagpos@eval@centerone{%
1019   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1020     -\eql@tagheight@block@-\eql@tagdepth@block@)/\tw@\relax
1021 }

```

TODO: describe

```

1022 \let\eql@numbering@init@baseline\eql@numbering@init@continuous
1023 \def\eql@tagpos@eval@baseline{%
1024   \eql@tagpos@current@\dimexpr(\eql@totalheight@
1025     +\eql@topheight@-\eql@bottomdepth@)/\tw@-\eql@tagheight@block@\relax
1026 }

```

TODO: describe

```

1027 \let\eql@numbering@init@median\eql@numbering@init@continuous
1028 \def\eql@tagpos@eval@median{%
1029   \ifnum\eql@tagpos@row@=\z@
1030     \ifodd\eql@totalrows@
1031       \eql@tagpos@row@\numexpr(\eql@totalrows@+\@ne)/\tw@\relax
1032     \else
1033       \eql@tagpos@row@\numexpr(\eql@totalrows@+\tw@)/\tw@\relax
1034       \eql@dimensions@get\eql@tagpos@row@
1035       \advance\eql@tagpos@shift@\dimexpr\eql@line@height@
1036         +(\eql@line@interline@-\eql@tagheight@block@
1037         +\eql@tagdepth@block@)/\tw@\relax
1038     \fi
1039     \ifnum\eql@totalrows@=\@ne
1040       \eql@tagpos@row@\@ne
1041     \else
1042       \eql@tagpos@adjust@eval@convert
1043       \eql@tagpos@row@\z@
1044     \fi
1045   \fi
1046 }

```

Selection.

```
1047 \def\eql@numbering@set#1{%
1048   \ifcsname eql@numbering@tab@#1\endcsname
1049     \expandafter\let\expandafter\eql@numbering@mode
1050       \csname eql@numbering@tab@#1\endcsname
1051   \ifx\eql@numbering@mode\eql@numbering@tab@all
1052     \let\eql@numbering@mode@multi\eql@numbering@mode
1053   \else\ifx\eql@numbering@mode\eql@numbering@tab@sub
1054     \let\eql@numbering@mode@multi\eql@numbering@mode
1055   \else
1056     \let\eql@numbering@mode@single\eql@numbering@mode
1057   \fi\fi
1058 \else
1059   \eql@error{numbering mode '#1' unknown: setting mode to 'all'}%
1060   \let\eql@numbering@mode\eql@numbering@tab@all
1061 \fi
1062 }
```

TODO: describe

```
1063 \def\eql@numbering@init{%
1064   \let\eql@numbering@multi\eql@false
1065   \let\eql@tagpos@continuous\eql@false
1066   \let\eql@numbering@subeq@use\eql@false
1067   \let\eql@numbering@best@use\eql@false
1068   \eql@tagpos@row@z@
1069   \csname eql@numbering@init@\eql@numbering@mode\endcsname
1070   \ifdefined\eql@numbering@active
1071     \let\eql@numbering@eqnswinit\@eqnswtrue
1072   \else
1073     \let\eql@numbering@eqnswinit\@eqnswfalse
1074   \fi
1075   \let\eql@numbering@active\eql@false
1076 }
```

5.3 Interface

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

```
1077 \eql@amsmath@after{
1078   \let\eql@print@eqnum@default\print@eqnum
1079   \let\eql@incr@eqnum@default\incr@eqnum
1080 }
```

TODO: describe

```
1081 \protected\def\donumber{%
1082   \if@eqnsw\else
1083     \global\@eqnswtrue
1084     \ifx\print@eqn\@empty
1085       \global\let\print@eqn\eql@print@eqnum@default
1086     \fi
1087     \ifx\incr@eqn\@empty
1088       \global\let\incr@eqn\eql@incr@eqnum@default
1089     \fi
1090   \fi
1091 }
```

TODO: reconsider operation

`\numberhere`

```
1092 \protected\def\eq1@numberhere{%
1093   \ifdefined\eq1@numbering@multi
1094     \global\@eqnswtrue
1095   \else
1096     \global\eq1@tagpos@row@\eq1@row@
1097   \fi
1098 }
```

TODO: describe

`\numbernext`

```
1099 \protected\def\eq1@numbernext{%
1100   \ifdefined\eq1@numbering@multi
1101     \global\@eqnswfalse
1102   \else
1103     \ifdefined\eq1@tagpos@continuous\else
1104       \ifnum\eq1@tagpos@row@=\eq1@row@
1105         \global\advance\eq1@tagpos@row@\@ne
1106       \fi
1107     \fi
1108   \fi
1109 }
```

Activation Trigger.

```
1110 \def\eq1@tags@autoenable{%
1111   \global\@eqnswtrue
1112   \ifnum\eq1@tagpos@row@=\m@ne
1113     \numberhere
1114   \fi
1115 }
```

Labels. **TODO:** describe

`\eq1@label@org`

```
1116 \let\eq1@label@org\label
```

TODO: describe

```
1117 \def\eq1@label@gobble{\eq1@ampprotect\eq1@testopt@tight\eq1@gobbleoptone{}}
```

TODO: describe

```
1118 \protected\def\eq1@label{%
1119   \eq1@ampprotect\eq1@testopt@tight\eq1@tags@add@labelname\eq1@testopt@default
1120 }
```

TODO: describe

```
1121 \def\eq1@tags@add@labelname[#1]#2{%
1122   \def\eq1@tmp{#1}%
1123   \ifx\eq1@tmp\eq1@testopt@default\else
1124     \eq1@tags@add@name{#1}%
1125   \fi
1126   \eq1@tags@add@label{#2}%
1127 }
```

TODO: describe

```
1128 \def\eql@tags@set@label#1{%
1129   \ifdefined\eql@tags@warn
1130     \ifdefined\eql@tags@label
1131       \eql@warn@label@multiple{#1}%
1132     \fi
1133   \fi
1134   \def\eql@tags@label{#1}%
1135 }
```

TODO: describe

```
1136 \def\eql@tags@set@name#1{%
1137   \ifdefined\eql@tags@warn
1138     \ifdefined\eql@tags@name
1139       \eql@warn@name@multiple
1140     \fi
1141   \fi
1142   \def\eql@tags@name{#1}%
1143 }
```

TODO: describe

```
1144 \def\eql@tags@add@label#1{%
1145   \ifdefined\eql@tags@autolabel
1146     \eql@tags@autoenable
1147   \fi
1148   \global\eql@appendexpand\eql@tags@container{%
1149     \noexpand\eql@tags@set@label{#1}}%
1150 }
```

TODO: describe

```
1151 \def\eql@tags@add@name#1{%
1152   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1153   \global\eql@appendmacro\eql@tags@container\eql@tmp
1154 }
```

TODO: describe

```
1155 \def\eql@tags@addblock@label#1{%
1156   \eql@appendexpand\eql@tags@container@block{%
1157     \noexpand\eql@tags@set@label{#1}}%
1158 }
```

TODO: describe

```
1159 \def\eql@tags@addblock@name#1{%
1160   \protected@edef\eql@tmp{\noexpand\eql@tags@set@name{#1}}%
1161   \eql@appendmacro\eql@tags@container@block\eql@tmp
1162 }
```

Tags. **TODO:** describe

`\eql@tag@default`

```
1163 \protected\def\eql@tag@default{%
1164   \eql@warn@here\tag
1165   \eql@tag@gobble
1166 }
1167 \let\tag\eql@tag@default
```

\eql@tag@gobble

```
1168 \def\eql@tag@gobble{%
1169   \eql@ampprotecttwo\eql@teststaropt@tight\eql@gobbleoptone\eql@gobbleoptone{}}
```

TODO: describe

```
1170 \protected\def\eql@tag{%
1171   \eql@ampprotecttwo\eql@teststaropt@tight
1172   {\eql@tags@add@tagform@off\eql@tags@add@tagref}{\eql@tags@add@tagref}
1173   \eql@testopt@default
1174 }
```

\eql@tags@add@tagref

```
1175 \def\eql@tags@add@tagref[#1]#2{%
1176   \def\eql@tmp{#1}%
1177   \ifx\eql@tmp\eql@testopt@default\else
1178     \eql@tags@add@ref{#1}%
1179   \fi
1180   \eql@tags@add@tag{#2}%
1181 }
```

TODO: describe

```
1182 \def\eql@tags@set@tag#1{%
1183   \ifdefined\eql@tags@warn
1184     \ifdefined\eql@tags@tag
1185       \eql@warn@tag@multiple
1186     \fi
1187   \fi
1188   \def\eql@tags@tag{#1}%
1189 }
```

TODO: describe

```
1190 \def\eql@tags@set@ref#1{%
1191   \ifdefined\eql@tags@warn
1192     \ifdefined\eql@tags@ref
1193       \eql@warn@ref@multiple
1194     \fi
1195   \fi
1196   \def\eql@tags@ref{#1}%
1197 }
```

TODO: describe

```
1198 \def\eql@tags@add@tag#1{%
1199   \ifdefined\eql@tags@autotag
1200     \eql@tags@autoenable
1201   \fi
1202   \protected\edef\eql@tmp{\noexpand\eql@tags@set@tag{#1}}%
1203   \global\eql@appendmacro\eql@tags@container\eql@tmp
1204 }
```

TODO: describe

```
1205 \def\eql@tags@add@ref#1{%
1206   \protected\edef\eql@tmp{\noexpand\eql@tags@set@ref{#1}}%
1207   \global\eql@appendmacro\eql@tags@container\eql@tmp
1208 }
```

tags@add@tagform@off

```
1209 \def\eq@tags@add@tagform@off{%
1210   \global\eq@append\eq@tags@container{\let\eq@tags@tagform\@firstofone}%
1211 }
```

TODO: describe

```
1212 \def\eq@tags@addblock@tag#1{%
1213   \protected@edef\eq@tmp{\noexpand\eq@tags@set@tag{#1}}%
1214   \eq@appendmacro\eq@tags@container@block\eq@tmp
1215 }
```

TODO: describe

```
1216 \def\eq@tags@addblock@ref#1{%
1217   \protected@edef\eq@tmp{\noexpand\eq@tags@set@ref{#1}}%
1218   \eq@appendmacro\eq@tags@container@block\eq@tmp
1219 }
```

TODO: describe

```
1220 \def\eq@tags@addblock@tagform@off{%
1221   \eq@append\eq@tags@container@block{\let\eq@tags@tagform\@firstofone}%
1222 }
```

Raise Tags.

\raisetag

```
1223 \def\eq@raisetag@default{%
1224   \eq@warn@here\raisetag
1225   \eq@raisetag@gobble
1226 }

1227 \def\eq@raisetag@gobble{%
1228   \eq@ampprotecttwo\eq@ifstar@tight\@gobble\@gobble
1229 }
```

TODO: describe

```
1230 \eq@amsmath@let\raisetag\eq@raisetag@default

1231 \def\eq@raisetag{%
1232   \eq@ampprotecttwo\eq@ifstar@tight\eq@tags@add@raiseshift\eq@raisetag@test
1233 }

1234 \def\eq@raisetag@test#1{%
1235   \def\eq@tmpa{#1}\def\eq@tmpb{!}%
1236   \ifx\eq@tmpa\eq@tmpb
1237     \eq@tags@add@forceraise
1238   \else
1239     \eq@tags@add@raisesmash{#1}%
1240   \fi
1241 }

1242 \def\eq@tags@add@raiseshift#1{%
1243   \global\eq@appendexpand\eq@tags@container{%
1244     \advance\eq@tagpos@shift@the\glueexpr#1\relax\relax}%
1245 }
```

```

1246 \def\eql@tags@add@raisesmash#1{%
1247   \dimen@{\glueexpr#1\relax
1248   \ifdim\dimen@<\z@
1249     \global\eql@appendexpand\eql@tags@container{%
1250       \advance\eql@tagpos@smashdown@-\the\dimen@\relax}%
1251   \else
1252     \global\eql@appendexpand\eql@tags@container{%
1253       \advance\eql@tagpos@smashup@\the\dimen@\relax}%
1254   \fi
1255 }

1256 \def\eql@tags@add@forceraise{%
1257   \global\eql@append\eql@tags@container{\let\eql@tagpos@reserve\eql@false}%
1258 }

```

5.4 Integration

TODO: describe

Support. **TODO:** describe

```

1259 \def\eql@numbering@settools{%
1260   \let\label\eql@label
1261   \let\tag\eql@tag
1262   \let\raisetag\eql@raisetag
1263   \let\numberhere\eql@numberhere
1264   \let\numbernext\eql@numbernext
1265 }

```

TODO: not necessary anymore

```

1266 \def\eql@numbering@settools@gobble{%
1267   \let\label\eql@label@gobble
1268   \let\tag\eql@tag@gobble
1269   \let\raisetag\eql@raisetag@gobble
1270   \let\numberhere\relax
1271   \let\numbernext\relax
1272 }

```

```

1273 \def\eql@numbering@autoblock{%
1274   \begingroup
1275     \let\eql@tags@warn\eql@false
1276     \eql@tags@container@block
1277     \ifdefined\eql@tags@autolabel
1278       \ifdefined\eql@tags@label
1279         \global\@eqnswtrue
1280       \fi
1281     \fi
1282     \ifdefined\eql@tags@autotag
1283       \ifdefined\eql@tags@tag
1284         \global\@eqnswtrue
1285       \fi
1286     \fi
1287   \endgroup
1288 }

```

```

1289 \def\eql@numbering@warnunused{%
1290   \ifdefined\eql@tags@label

```

```

1291     \eql@warn@label@unused
1292 \fi
1293 \ifdefined\eql@tags@name
1294     \eql@warn@name@unused
1295 \fi
1296 \ifdefined\eql@tags@tag
1297     \eql@warn@tag@unused
1298 \fi
1299 \ifdefined\eql@tags@erf
1300     \eql@warn@ref@unused
1301 \fi
1302 }

```

Single Line. **TODO:** describe

```

1303 \def\eql@numbering@single@init{%
1304     \let\eql@numbering@multi\eql@false
1305     \eql@numbering@settools
1306     \eql@numbering@eqnswinit
1307     \eql@numbering@autoblock
1308     \global\let\eql@tags@container\eql@tags@container@block
1309     \let\eql@tags@warn\eql@true
1310 }

1311 \def\eql@numbering@single@eval{%
1312     \ifnum\eql@tagpos@row@=\m@ne
1313         \@eqnswfalse
1314     \fi
1315 }

```

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

```

1316 \def\eql@numbering@measure@init{%
1317     \eql@numbering@settools
1318     \ifdefined\eql@numbering@multi\else
1319         \eql@numbering@eqnswinit
1320         \eql@numbering@autoblock
1321     \fi
1322     \global\let\eql@tags@container\eql@tags@container@block
1323     \let\eql@tags@warn\eql@true
1324 }

```

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

```

1325 \def\eql@numbering@measure@line@begin{%
1326     \ifdefined\eql@numbering@multi
1327         \global\eql@numbering@eqnswinit
1328     \fi
1329 }

```

TODO: describe

```

1330 \def\eql@numbering@measure@blocktag{%
1331     \ifdefined\eql@numbering@multi
1332         \@eqnswfalse
1333     \else
1334         \ifnum\eql@tagpos@row@=\m@ne
1335             \@eqnswfalse
1336         \fi

```



```

1337 \ifnum\eql@totalrows@=\z@
1338 \eqnswfalse
1339 \fi
1340 \fi
1341 }

```

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!

```

1342 \def\eql@numbering@print@init{%
1343 \let\eql@tags@warn\eql@false
1344 \ifdefined\eql@numbering@multi
1345 \eql@numbering@settools
1346 \global\let\eql@tags@container\eql@tags@container@block
1347 \else
1348 \let\eql@tags@container@block\eql@tags@container
1349 \eql@numbering@settools@gobble
1350 \fi
1351 }

```

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

```

1352 \def\eql@numbering@print@block@begin{%
1353 \ifdefined\eql@numbering@multi\else
1354 \ifnum\eql@tagpos@row@>\z@
1355 \eql@tags@makeblockanchor
1356 \global\eql@appendexpand\eql@tags@container@block{%
1357 \def\noexpand\eql@tags@anchor{%
1358 \unexpanded\expandafter{\eql@tags@anchor}}}%
1359 \fi
1360 \fi
1361 \ifdefined\eql@numbering@subeq@use
1362 \eql@tags@printsubeqlabel
1363 \fi
1364 }

```

TODO: describe

```

1365 \def\eql@numbering@print@line@begin{%
1366 \ifdefined\eql@numbering@multi
1367 \global\eql@numbering@eqnswinit
1368 \fi
1369 }

```

TODO: describe

```

1370 \def\eql@numbering@print@line@eval{%
1371 \ifdefined\eql@numbering@multi
1372 \ifeqnsw
1373 \eql@tags@container
1374 \fi
1375 \else
1376 \ifnum\eql@tagpos@row@=\eql@row@
1377 \eqnswtrue
1378 \eql@tags@container@block
1379 \else
1380 \eqnswfalse
1381 \fi

```

```

1382 \fi
1383 }

```

5.5 Positioning

TODO: describe

```

1384 \def\eql@tagpos@single@eval{%
1385   \if@eqnsw
1386     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1387     \ifnum\eql@tagpos@row@>\@ne
1388       \eql@tagpos@row@\@ne
1389     \fi
1390     \ifdefined\eql@tagpos@doconvert
1391       \let\eql@tagpos@continuous\eql@true
1392     \fi
1393     \ifdefined\eql@tagpos@continuous
1394       \eql@tagpos@single@eval@continuous
1395     \fi
1396   \else
1397     \eql@tagpos@row@\z@
1398   \fi
1399   \eql@tagpos@prevrow@\z@
1400   \eql@tagpos@headroom@\z@
1401   \eql@tagpos@footroom@\z@
1402 }

```

TODO: describe

```

1403 \def\eql@tagpos@single@eval@continuous{%
1404   \ifnum\eql@tagpos@row@>\z@
1405     \eql@tagpos@target@\eql@tagpos@shift@
1406   \else
1407     \eql@tagpos@target@\dimexpr\eql@line@height@
1408       -\eql@tagpos@current@+\eql@tagpos@shift@-\eql@tagheight@block@\relax
1409   \fi
1410   \eql@tagpos@row@\@ne
1411   \ifdim\ifdim\eql@tagpos@target@<\z@-\fi
1412     \eql@tagpos@target@<\glueexpr\eql@tagpos@snap\relax
1413     \eql@tagpos@target@\z@
1414   \fi
1415 }

```

TODO: describe

```

1416 \def\eql@tagpos@adjust@eval{%
1417   \if@eqnsw
1418     \csname eql@tagpos@eval@\eql@numbering@mode\endcsname
1419     \ifnum\eql@tagpos@row@>\eql@totalrows@
1420       \eql@tagpos@row@\eql@totalrows@
1421     \fi
1422     \ifdefined\eql@tagpos@doconvert
1423       \let\eql@tagpos@continuous\eql@true
1424     \fi
1425     \ifdefined\eql@tagpos@continuous
1426       \ifnum\eql@tagpos@row@>\z@
1427         \eql@tagpos@adjust@eval@convert
1428       \fi
1429       \eql@tagpos@adjust@eval@continuous

```

```

1430 \fi
1431 \else
1432 \eql@tagpos@row@z@
1433 \eql@tagpos@prevrow@z@
1434 \fi
1435 }

```

TODO: describe

```

1436 \def\eql@tagpos@adjust@eval@convert{%
1437 \eql@tagpos@current@z@
1438 \eql@dimensions@for{%
1439 \ifnum\eql@row@<\eql@tagpos@row@
1440 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1441 +\eql@line@height@+\eql@line@depth@relax
1442 \fi
1443 \ifnum\eql@row@=\eql@tagpos@row@
1444 \advance\eql@tagpos@current@dimexpr\eql@line@interline@
1445 +\eql@line@height@-\eql@tagheight@block@relax
1446 \fi
1447 }%
1448 }

```

TODO: describe

```

1449 \def\eql@tagpos@adjust@eval@continuous{%
1450 \dimen@dimexpr\eql@tagpos@current@-\eql@tagpos@shift@relax
1451 \eql@tagpos@row@\eql@totalrows@
1452 \eql@tagpos@prevrow@z@
1453 \eql@tagpos@headroom@z@
1454 \eql@tagpos@footroom@z@
1455 \eql@dimensions@for{%
1456 \ifnum\eql@tagpos@row@=\eql@totalrows@
1457 \eql@tagpos@headroom@\eql@line@interline@
1458 \eql@tagpos@target@dimexpr\eql@line@interline@
1459 +\eql@line@height@-\dimen@-\eql@tagheight@block@relax
1460 \ifdim\ifdim\eql@tagpos@target@<z@-\fi
1461 \eql@tagpos@target@<\glueexpr\eql@tagpos@snaprelax
1462 \advance\dimen@\eql@tagpos@target@
1463 \eql@tagpos@target@z@
1464 \fi
1465 \ifdim\eql@tagpos@target@>%
1466 \ifdefined\eql@tagsleft-1sprelax\elsez@\fi
1467 \eql@tagpos@row@\eql@row@
1468 \eql@tagpos@prevrow@\numexpr\eql@row@-\@ne\relax
1469 \fi
1470 \advance\dimen@-\dimexpr\eql@line@interline@
1471 +\eql@line@depth@+\eql@line@height@relax
1472 \fi
1473 \ifnum\eql@row@=\numexpr\eql@tagpos@row@+\@ne\relax
1474 \eql@tagpos@footroom@\eql@line@interline@
1475 \fi
1476 }%
1477 }

```

TODO: describe

```

1478 \def\eql@tagpos@print@line@eval{%
1479 \ifdefined\eql@tagpos@continuous
1480 \eql@tagpos@print@line@eval@continuous
1481 \else

```

```

1482 \eq\tagpos@print@line@eval@discrete
1483 \fi
1484 }

```

TODO: describe

```

1485 \def\eq\tagpos@print@line@eval@continuous{%
1486 \if@eqnsw
1487 \ht\eq\tagbox@\dimexpr\ht\eq\tagbox@-\eq\tagpos@smashup@\relax
1488 \dp\eq\tagbox@\dimexpr\dp\eq\tagbox@-\eq\tagpos@smashdown@\relax
1489 \eq\tagpos@plain@\eq\tagpos@target@
1490 \@tempdima\dimexpr\eq\line@height@+\eq\tagpos@headroom@
1491 -\ht\eq\tagbox@\relax
1492 \@tempdimb\dimexpr-\eq\line@depth@-\eq\tagpos@footroom@
1493 +\dp\eq\tagbox@\relax
1494 \ifnum\eq\row@=\@ne
1495 \@tempdima.5\maxdimen
1496 \fi
1497 \ifnum\eq\row@=\eq\totalrows@
1498 \@tempdimb-.5\maxdimen
1499 \fi
1500 \ifdim\eq\tagpos@plain@>\@tempdima
1501 \ifdim\eq\tagpos@plain@>\@tempdimb
1502 \ifdim\@tempdima>\@tempdimb
1503 \eq\tagpos@plain@\@tempdima
1504 \else
1505 \eq\tagpos@plain@\@tempdimb
1506 \fi
1507 \fi
1508 \else
1509 \ifdim\eq\tagpos@plain@<\@tempdimb
1510 \ifdim\@tempdima>\@tempdimb
1511 \eq\tagpos@plain@\@tempdimb
1512 \else
1513 \eq\tagpos@plain@\@tempdima
1514 \fi
1515 \fi
1516 \fi
1517 \ifnum\eq\tagpos@prevrow@>\z@
1518 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1519 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1520 \eq\tagpos@raised@\eq\tagpos@plain@
1521 \let\eq\tagpos@reserve\eq>false
1522 \fi
1523 \else
1524 \ifdim\eq\tagpos@target@>%
1525 \ifdefined\eq\tagleft-isp\relax\else\z@\fi
1526 \eq\tagpos@raised@\dimexpr\eq\line@height@+\dp\eq\tagbox@\relax
1527 \ifdim\eq\tagpos@raised@>\eq\tagpos@plain@\else
1528 \eq\tagpos@raised@\eq\tagpos@plain@
1529 \let\eq\tagpos@reserve\eq>false
1530 \fi
1531 \else
1532 \eq\tagpos@raised@\dimexpr-\eq\line@depth@
1533 -\ht\eq\tagbox@\relax
1534 \ifdim\eq\tagpos@raised@<\eq\tagpos@plain@\else
1535 \eq\tagpos@raised@\eq\tagpos@plain@
1536 \let\eq\tagpos@reserve\eq>false
1537 \fi

```

```

1538     \fi
1539   \fi
1540 \else
1541   \ifnum\eql@tagpos@prevrow@=\eql@row@
1542     \eql@tagwidth@\eql@tagwidth@block@
1543   \else
1544     \let\eql@tagpos@reserve\eql@false
1545   \fi
1546 \fi
1547 }

```

TODO: describe

```

1548 \def\eql@tagpos@print@line@eval@discrete{%
1549   \if@eqnsw
1550     \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@smashup@\relax
1551     \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@smashdown@\relax
1552     \eql@tagpos@plain@\eql@tagpos@shift@
1553     \eql@tagpos@headroom@z@
1554     \eql@tagpos@footroom@z@
1555     \ifdim\eql@tagpos@shift@>%
1556       \ifdefined\eql@tagsleft-1sp\relax\else z@\fi
1557     \eql@tagpos@raised@\dimexpr\eql@line@height@+\dp\eql@tagbox@\relax
1558   \else
1559     \eql@tagpos@raised@\dimexpr-\eql@line@depth@-\ht\eql@tagbox@\relax
1560   \fi
1561 \else
1562   \let\eql@tagpos@reserve\eql@false
1563 \fi
1564 }

```

TODO: describe

```

1565 \def\eql@tagpos@print@line@end{%
1566   \ifdefined\eql@tagpos@continuous
1567     \ifnum\eql@tagpos@prevrow@=\eql@row@
1568       \ifdefined\eql@tagpos@reserve
1569         \global\eql@appendexpand\eql@tags@container@block{%
1570           \advance\eql@tagpos@headroom@the\dimexpr\eql@line@height@
1571             +\eql@line@depth@\relax\relax}%
1572         \eql@displaybreak@star\@M
1573       \fi
1574     \fi
1575   \fi
1576 }

```

5.6 Component Display

Showkeys Integration. **TODO:** describe

```

1577 \let\eql@SK@loaded\eql@false
1578 \let\eql@SK@label\@gobble
1579 \let\eql@SK@clearlabel\@empty
1580 \let\eql@SK@setlabel\@gobble
1581 \let\eql@SK@printlabel@right\@empty
1582 \let\eql@SK@printlabel@left\@empty
1583 \let\eql@SK@printlabel@line\@empty
1584 \def\eql@label@clean{\eql@label@org}
1585 \AddToHook{package/showkeys/after}{

```

```

1586 \let\eql@SK@loaded\eql@true
1587 \def\eql@SK@label#1{\SK@\SK@@label#1}
1588 \def\eql@SK@clearlabel{\let\eql@SK@lab\relax}
1589 \eql@SK@clearlabel
1590 \def\eql@SK@@label#1>#2\SK@{%
1591   \def\eql@SK@lab{\smash{\SK@labelcolor\showkeyslabelformat{#2}}}%
1592 }
1593 \def\eql@SK@setlabel#1{\SK@\eql@SK@@label#1}
1594 \def\eql@SK@printlabel@right{%
1595   \ifx\eql@SK@lab\relax\else
1596     \rlap{\kern\marginparsep\eql@SK@lab}%
1597     \eql@SK@clearlabel
1598   \fi
1599 }
1600 \def\eql@SK@printlabel@left{%
1601   \ifx\eql@SK@lab\relax\else
1602     \llap{\eql@SK@lab\kern\marginparsep}%
1603     \eql@SK@clearlabel
1604   \fi
1605 }
1606 \def\eql@SK@printlabel@line{%
1607   \ifx\eql@SK@lab\relax\else
1608     \dimen@ \prevdepth
1609     \nointerlineskip
1610     \ifdefined\eql@tagsleft
1611       \llap{%
1612         \eql@SK@lab
1613         \kern\marginparsep
1614       }%
1615       \eql@SK@clearlabel
1616     \else
1617       \rlap{%
1618         \dimen@ \displaywidth
1619         \advance\dimen@ \marginparsep
1620         \kern\dimen@
1621         \eql@SK@lab
1622       }%
1623       \fi
1624       \eql@SK@clearlabel
1625       \prevdepth\dimen@
1626     \fi
1627   }
1628 \let\eql@label@org\label
1629 \def\eql@label@clean{\let\SK@\gobbletwo\eql@label@org}
1630 }

```

Labels.

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

```

1631 \def\eql@composetag@label{%
1632   \eql@SK@clearlabel
1633   \ifdefined\eql@tags@label
1634     \eql@SK@setlabel\eql@tags@label
1635     \ifdefined\eql@tags@name
1636       \let\@currentlabelname\eql@tags@name
1637     \else
1638       \let\@currentlabelname\eql@tags@name@generic

```

```

1639 \fi
1640 \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1641 \fi
1642 }

```

TODO: describe

```

1643 \def\eq1@tags@printsubeqlabel{%
1644 \eq1@tags@container@parent
1645 \ifdefined\eq1@tags@label
1646 \eq1@tags@makeblockanchor
1647 \eq1@SK@setlabel\eq1@tags@label
1648 \begingroup
1649 \def\@currentcounter{equation}%
1650 \eq1@tags@anchor
1651 \let\@currentlabelname\eq1@tags@name@generic
1652 \protected@edef\@currentlabel{\p@equation\theparentequation}%
1653 \expandafter\eq1@label@clean\expandafter{\eq1@tags@label}%
1654 \endgroup
1655 \eq1@SK@printlabel@line
1656 \fi
1657 }

```

Hyperref Anchors. **TODO:** describe

```

1658 \let\eq1@Hy@anchor\@gobble
1659 \AddToHook{package/hyperref/after}{
1660 \def\eq1@Hy@anchor#1{%
1661 \Hy@raisedlink{\hyper@anchor{#1}}%
1662 }%
1663 }

```

TODO: describe

```

1664 \def\eq1@tags@makeblockanchor{%
1665 \eq1@tags@glabel@step
1666 \eq1@Hy@anchor\eq1@tags@glabel
1667 \edef\eq1@tags@anchor{%
1668 \def\noexpand\thepage{\thepage}%
1669 \def\noexpand\@currentHref{\eq1@tags@glabel}%
1670 }%
1671 }

```

TODO: describe

ql@composetag@anchor

```

1672 \def\eq1@composetag@anchor{%
1673 \ifdefined\eq1@tags@tag
1674 \def\@currentcounter{equation}%
1675 \ifdefined\eq1@tags@ref
1676 \let\@currentlabel\eq1@tags@ref
1677 \else
1678 \protected@edef\@currentlabel{\p@equation\eq1@tags@tag}%
1679 \fi
1680 \eq1@tags@glabel@step
1681 \edef\@currentHref{\eq1@tags@glabel}%
1682 \eq1@Hy@anchor\@currentHref
1683 \else

```

```

1684 \refstepcounter{equation}%
1685 \protected@edef\eql@tags@tag{\theequation}%
1686 \fi
1687 \eql@tags@anchor
1688 }

```

Tag Layout. **TODO:** describe

```

1689 \def\eql@tags@taglayout@set@direct#1{%
1690 \def\eql@tags@taglayout##1{#1}%
1691 }
1692 \def\eql@tags@taglayout@set#1{%
1693 \def\eql@tags@taglayout##1{\hbox{\m@th\normalfont#1}}%
1694 }

```

TODO: describe

```

1695 \def\eql@tags@tagform@set@direct#1{%
1696 \def\eql@tags@tagform##1{#1}%
1697 }
1698 \def\eql@tags@tagform@set#1#2#3{%
1699 \def\eql@tags@tagform##1{#1\ignorespaces#2\unskip\@@italiccorr#3}%
1700 }

1701 \eql@tags@taglayout@set{#1}
1702 \eql@tags@tagform@set({#1})
1703 \def\eql@tags@tagcompose#1{\eql@tags@taglayout{\eql@tags@tagform{#1}}}

1704 \protected\def\tagform{\eql@tags@tagform}
1705 \protected\def\tagbox{\eql@tags@taglayout}
1706 \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:

```

1707 \protected\def\eql@eqref#1{\textup{\eql@tags@tagcompose{\ref{#1}}}}

```

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

```

1708 \def\eql@composetag@tag{%
1709 \eql@tagging@tagbegin
1710 \eql@tags@frame@cmd{%
1711 \eql@tags@taglayout{%
1712 \eql@tags@tagform\eql@tags@tag
1713 \eql@tagging@tagsave
1714 }%
1715 }%
1716 \eql@tagging@tagend
1717 }

```

5.7 Tag Composition

TODO: describe

```

1718 \def\eql@composetag@measure{%
1719 \ifdefined\eql@tags@tag\else
1720 \stepcounter{equation}%
1721 \let\eql@tags@tag\theequation

```



```

1722 \fi
1723 \eql@tags@frame@cmd{\eql@tags@taglayout{\eql@tags@tagform\eql@tags@tag}}%
1724 \ifdefined\eql@numbering@multi
1725   \global\let\eql@tags@container\eql@tags@container@clear
1726 \fi
1727 }

```

TODO: describe

```

1728 \def\eql@composetag@print{%
1729   \eql@composetag@anchor
1730   \eql@composetag@label
1731   \ifdefined\eql@tags@left
1732     \eql@SK@printlabel@left
1733     \eql@composetag@tag
1734   \else
1735     \eql@composetag@tag
1736     \eql@SK@printlabel@right
1737   \fi
1738   \global\let\eql@tags@container\eql@tags@container@clear
1739 }

```

TODO: describe

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

```

1740 \def\eql@tagbox@make#1{%
1741   \setbox\eql@tagbox@\hbox{\eql@strut@tag\@lign#1}%
1742   \eql@tagwidth@\wd\eql@tagbox@
1743   \ifdim\eql@tagwidth@<\eql@tagwidthmin@
1744     \eql@tagwidth@\eql@tagwidthmin@
1745   \fi
1746   \advance\eql@tagwidth@\eql@tagsepmin@
1747 }

```

TODO: describe

```

1748 \def\eql@tagbox@print@adjustheadroom{%
1749   \dimen@\dimexpr\ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@\relax
1750   \ifdim\dimen@>\z@
1751     \ifdim\dimen@>\eql@tagpos@headroom@
1752       \ht\eql@tagbox@\dimexpr\ht\eql@tagbox@-\eql@tagpos@headroom@\relax
1753     \else
1754       \ht\eql@tagbox@\dimexpr\eql@line@height@-\eql@tagpos@current@\relax
1755     \fi
1756   \fi
1757 }

```

TODO: describe

```

1758 \def\eql@tagbox@print@adjustfootroom{%
1759   \dimen@\dimexpr\dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@\relax
1760   \ifdim\dimen@>\z@
1761     \ifdim\dimen@>\eql@tagpos@footroom@
1762       \dp\eql@tagbox@\dimexpr\dp\eql@tagbox@-\eql@tagpos@footroom@\relax
1763     \else
1764       \dp\eql@tagbox@\dimexpr\eql@line@depth@+\eql@tagpos@current@\relax
1765     \fi
1766   \fi
1767 }

```

TODO: describe

```

1768 \def\eql@tagbox@print@extendabove{%
1769   \dimen@ \dimexpr \ht\eql@tagbox@+\eql@tagpos@current@-\eql@line@height@ \relax
1770   \ifdim \dimen@>\z@
1771     \global\eql@appendexpand\eql@display@container{%
1772       \eql@display@aboveextend@ \the\dimen@ \relax}%
1773   \fi
1774 }

```

TODO: describe

```

1775 \def\eql@tagbox@print@extendbelow{%
1776   \dimen@ \dimexpr \dp\eql@tagbox@-\eql@tagpos@current@-\eql@line@depth@ \relax
1777   \ifdim \dimen@>\z@
1778     \global\eql@appendexpand\eql@display@container{%
1779       \eql@display@belowextend@ \the\dimen@ \relax}%
1780   \fi
1781 }

```

TODO: describe

```

1782 \def\eql@tagbox@print@prepare{%
1783   \ifdefined\eql@tagpos@reserve
1784     \eql@tagpos@current@ \eql@tagpos@plain@
1785   \else
1786     \eql@tagpos@current@ \eql@tagpos@raised@
1787   \fi
1788   \ifdim \eql@tagpos@headroom@>\z@
1789     \eql@tagbox@print@adjustheadroom
1790   \fi
1791   \ifdim \eql@tagpos@footroom@>\z@
1792     \eql@tagbox@print@adjustfootroom
1793   \fi
1794   \ifnum \eql@row@=\@ne
1795     \eql@tagbox@print@extendabove
1796   \fi
1797   \ifnum \eql@row@=\eql@totalrows@
1798     \eql@tagbox@print@extendbelow
1799   \fi
1800 }

```

TODO: describe

```

1801 \def\eql@tagbox@print@tagsright{%
1802   \eql@tagbox@print@prepare
1803   \kern-\wd\eql@tagbox@
1804   \raise\eql@tagpos@current@ \box\eql@tagbox@
1805 }

```

TODO: describe

```

1806 \def\eql@tagbox@print@tagsleft{%
1807   \eql@display@firstavail@set\z@
1808   \eql@tagbox@print@prepare
1809   \wd\eql@tagbox@ \z@
1810   \raise\eql@tagpos@current@ \box\eql@tagbox@
1811 }

```

$\text{eql@tagbox@print@cell}$

```

1812 \def\eql@tagbox@print@cell{%
1813   \eql@tagging@tagaddbox

```

```

1814 \ifdefined\eq\@tagsleft
1815   \ifdefined\eq\@tagpos@reserve
1816     \ifdim\eq\@tagwidth@>\dimexpr\eq\@line@avail@+\eq\@tagfuzz@\relax
1817       \let\eq\@tagpos@reserve\eq\@false
1818     \fi
1819   \fi
1820   \if@eqnsw
1821     \eq\@tagbox@print@tagsleft
1822   \fi
1823   \kern\displaywidth
1824 \else
1825   \kern\displaywidth
1826   \ifdefined\eq\@tagpos@reserve
1827     \ifdim\eq\@tagwidth@>%
1828       \dimexpr\displaywidth-\eq\@line@width@+\eq\@tagfuzz@\relax
1829     \let\eq\@tagpos@reserve\eq\@false
1830   \fi
1831 \fi
1832 \if@eqnsw
1833   \eq\@tagbox@print@tagsright
1834 \fi
1835 \fi
1836 }

```

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:

```

1837 \eq\@amsmath@undefine\c@parentequation
1838 \eq\@amsmath@undefine\theparentequation
1839 \ifdefined\c@parentequation\else
1840 \newcounter{parentequation}
1841 \fi

```

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

```

1842 \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:

```

1843 \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`:

```

1844 \def\eq@subequations@init{%
1845   \edef\eq@subequations@restorecounter{%
1846     \global\c@equation\the\c@equation\relax}%
1847   \eq@tags@container@block
1848   \ifdefined\eq@tags@tag
1849     \eq@tags@glabel@step
1850     \protected@edef\theHparentequation{\eq@tags@glabel}%
1851     \protected@edef\theparentequation{\eq@tags@tag}%
1852   \else
1853     \advance\c@equation\@ne
1854     \protected@edef\theparentequation{\theequation}%
1855     \ifdefined\theHequation
1856       \protected@edef\theHparentequation{\theHequation}%
1857     \fi
1858   \fi
1859   \global\c@parentequation\c@equation
1860   \global\c@equation\z@
1861   \let\theequation\eq@subequations@template
1862   \def\theHequation{\theHparentequation.\arabic{equation}}%
1863 }
```

`\close@subequations` 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:

```

1864 \def\eq@subequations@close{%
1865   \ifnum\c@equation=\z@
1866     \eq@subequations@restorecounter
1867   \else
1868     \global\c@equation\c@parentequation
1869   \fi
1870 }
```

6.2 Environment

`\start@subequations` 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`

```

1871 \def\eq@subequations@start{%
1872   \let\eq@tags@container@block\eq@tags@container@clear
1873   \eq@nextopt@process{subequations}%
1874   \eq@subequations@init
1875   \eq@tags@glabel@step
1876   \edef\eq@subequations@currentHref{\eq@tags@glabel}%
1877   \eq@Hy@anchor\eq@subequations@currentHref
1878   \edef\eq@subequations@thepage{\thepage}%
1879   \def\@currentcounter{equation}%
1880   \let\@currentHref\eq@subequations@currentHref
1881   \protected@edef\@currentlabel{\p@equation\theparentequation}%
1882   \eq@tags@container@block
1883   \ifdefined\eq@tags@name
1884     \let\@currentlabelname\eq@tags@name
1885   \else
```

```

1886 \let\@currentlabelname\eq\@tags\@name\@generic
1887 \fi
1888 \let\eq\@subequations\@active\eq\@true
1889 \ifdefined\eq\@tags\@label
1890 \eq\@SK\@label\eq\@tags\@label
1891 \fi
1892 \ignorespaces
1893 }

```

`\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:

```

1894 \def\eq\@subequations\@end{%
1895 \ifnum\c\@equation>\z@
1896 \eq\@tags\@container\@block
1897 \ifdefined\eq\@tags\@label
1898 \begingroup
1899 \def\@currentcounter{equation}%
1900 \let\thepage\eq\@subequations\thepage
1901 \let\@currentHref\eq\@subequations\@currentHref
1902 % \TODO how about tag* ?! also within equations!
1903 \protected\edef\@currentlabel{\p\@equation\theparentequation}%
1904 \ifdefined\eq\@tags\@name
1905 \let\@currentlabelname\eq\@tags\@name
1906 \else
1907 \let\@currentlabelname\eq\@tags\@name\@generic
1908 \fi
1909 \expandafter\eq\@label\@clean\expandafter{\eq\@tags\@label}%
1910 \endgroup
1911 \fi
1912 \fi
1913 \eq\@subequations\@close
1914 }

```

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

```

1915 \newenvironment{eq\@subequations}{%
1916 \eq\@verbose\@info\eq\@verbose\@msg\@enterenv
1917 \eq\@subequations\@testall\eq\@subequations\@start%
1918 }{%
1919 \eq\@subequations\@end
1920 \ignorespacesafterend
1921 \eq\@verbose\@info\eq\@verbose\@msg\@leaveenv
1922 }

```

TODO: describe

```

1923 \def\eq\@subequations\@testall{\eq\@parseopt\@env\eq\@subequations\@testall\@parse}
1924 \def\eq\@subequations\@testall\@parse{%
1925 \ifx\eq\@parseopt\@token[%]
1926 \let\eq\@parseopt\@next\eq\@parseopt\@opt
1927 \fi
1928 \ifx\eq\@parseopt\@token\eq\@atxi
1929 \let\eq\@parseopt\@next\eq\@parseopt\@label
1930 \fi
1931 \ifx\eq\@parseopt\@token\eq\@atxii
1932 \let\eq\@parseopt\@next\eq\@parseopt\@label
1933 \fi

```

```

1934 \ifx\eql@parseopt@token\label
1935 \let\eql@parseopt@next\eql@parseopt@end
1936 \fi
1937 }

```

6.3 Subequation Scheme

TODO: describe

```

1938 \def\eql@numbering@subeq@init{%
1939 \let\eql@save@theequation\theequation
1940 \let\eql@save@theHequation\theHequation
1941 \eql@subequations@init
1942 \let\eql@tags@container@parent\eql@tags@container@block
1943 \let\eql@tags@container@block\eql@tags@container@clear
1944 }

```

TODO: describe

```

1945 \def\eql@numbering@subeq@test{%
1946 \ifnum\eql@tagrows@<\tw@
1947 \let\eql@tags@container@block\eql@tags@container@parent
1948 \let\eql@numbering@subeq@use\eql@false
1949 \let\theequation\eql@save@theequation
1950 \let\theHequation\eql@save@theHequation
1951 \eql@subequations@restorecounter
1952 \fi
1953 }

```

TODO: describe

```

1954 % \TODO note must not use setcounter here (when calc is loaded)
1955 \def\eql@numbering@subeq@close{%
1956 \eql@subequations@close
1957 }

```

7 Display Equations Support

TODO: describe

```

1958 \let\eql@display@injectbefore\@undefined
1959 \let\eql@display@injectafter\@undefined
1960 \let\eql@interline@container\@undefined
1961 \def\eql@interline@container@clear{%
1962 \eql@displaybreak@open@\@MM
1963 \eql@vspaceskip@\z@skip
1964 }

```

7.1 Display Breaks

TODO: describe

erdisplaylinepenalty

```

1965 \interdisplaylinepenalty\@M

```

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

```
1966 \def\eqldisplaybreak@open#1{%
1967   \ifcase #1\@M \or 9999 \or 6999 \or 2999 \or \z@\fi
1968 }
```

TODO: allow a displaybreak before equations

```
1969 \protected\def\eqldisplaybreak@default{%
1970   \eqldisplaybreak@warning{Invalid use of \string\displaybreak}{}%
1971   \eqldisplaybreak@teststaroropt@loose\@gobble\eqldisplaybreak@opt}%
1972 \eqldisplaybreak@after{\let\eqldisplaybreak@default\displaybreak}
1973 \eqldisplaybreak@let\displaybreak\eqldisplaybreak@default
```

```
1974 \newcount\eqldisplaybreak@pen@
1975 \newcount\eqldisplaybreak@prepen@
1976 \newcount\eqldisplaybreak@postpen@
```

TODO: describe

```
1977 \protected\def\eqldisplaybreak{%
1978   \relax
1979   \eqldisplaybreak@protecttwo\eqldisplaybreak@teststaroropt@tight
1980   \eqldisplaybreak@star\eqldisplaybreak@level{4}%
1981 }
```

```
1982 \def\eqldisplaybreak@star#1{%
1983   \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1984     \eqldisplaybreak@open@the\numexpr#1\relax\relax}%
1985 }
```

```
1986 \def\eqldisplaybreak@level[#1]{%
1987   \ifnum#1<\z@
1988     \global\eqldisplaybreak@append\eqldisplaybreak@interline@container{\eqldisplaybreak@open@\@MM}%
1989   \else
1990     \global\eqldisplaybreak@appendexpand\eqldisplaybreak@interline@container{%
1991       \eqldisplaybreak@open@-\@getpen{#1}\relax}%
1992   \fi
1993 }
```

TODO: describe

```
1994 \def\eqldisplaybreak@pre#1{%
1995   \ifnum#1<\z@
1996     \eqldisplaybreak@prepen@\@MM
1997   \else
1998     \eqldisplaybreak@prepen@-\@getpen{#1}\relax
1999   \fi
2000 }
```

TODO: describe

```
2001 \def\eqldisplaybreak@post#1{%
2002   \ifnum#1<\z@
2003     \eqldisplaybreak@postpen@\@MM
2004   \else
2005     \eqldisplaybreak@postpen@-\@getpen{#1}\relax
2006   \fi
2007 }
```

TODO: describe

```

2008 \def\eqldisplaybreak@inter#1{%
2009   \ifnum#1<\z@
2010     \interdisplaylinepenalty\@M
2011   \else
2012     \interdisplaylinepenalty\eqldgetdsp@pen{#1}\relax
2013   \fi
2014 }

```

7.2 Explicit Vertical Space

TODO: describe

`\eqlvspaceskip@` (*skip*)

```

2015 \newskip\eqlvspaceskip@

2016 \let\eqlvspace@org\vspace
2017 \def\eqlvspace{%
2018   \ifvmode
2019     \expandafter\eqlvspace@immediate
2020   \else
2021     \expandafter\eqlvspace@line
2022   \fi
2023 }

```

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

```

2024 \def\eqlvspace@line{%
2025   \eq@ifstar@loose\eqlvspace@addfixedbefore\eqlvspace@add
2026 }
2027 \def\eqlvspace@add#1{%
2028   \global\eql@appendexpand\eql@interline@container{%
2029     \advance\eqlvspaceskip@\the\glueexpr#1\relax\relax}}
2030 \def\eqlvspace@addfixedbefore#1{%
2031   \global\eql@appendexpand\eql@interline@container{%
2032     \noexpand\eql@append\noexpand\eqldisplay@injectbefore{%
2033       \skip@\the\glueexpr#1\relax\relax
2034       \penalty\@M
2035       \vskip\skip@
2036       \global\advance\eql@line@interline@\skip@
2037     }%
2038   }%
2039 }
2040 \def\eqlvspace@addfixedafter#1{%
2041   \global\eql@appendexpand\eql@interline@container{%
2042     \noexpand\eql@append\noexpand\eqldisplay@injectafter{%
2043       \dimen@\prevdepth
2044       \hrule\@height\z@
2045       \skip@\the\glueexpr#1\relax\relax
2046       \penalty\@M
2047       \vskip\skip@
2048       \global\advance\eql@line@interline@\skip@
2049       \prevdepth\dimen@
2050     }%
2051   }%
2052 }

```


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

```

2053 \def\eqlvspace@immediate{%
2054   \noalign\bgroup
2055     \eq@ifstar@loose\eqlvspace@fixed\eqlvspace@discardable
2056 }
2057 \def\eqlvspace@fixed#1{%
2058   \skip@\glueexpr#1\relax
2059   \ifnum\eql@row@=\@ne
2060     \global\eql@appendexpand\eqldisplay@container{%
2061       \advance\eql@abovespace@\the\skip@\relax}%
2062   \else\ifnum\eql@row@>\eql@totalrows@
2063     \global\eql@appendexpand\eqldisplay@container{%
2064       \advance\eql@belowspace@\the\skip@\relax}%
2065   \else
2066     \dimen@\prevdepth
2067     \hrule\@height\z@
2068     \penalty\@M
2069     \vskip\skip@
2070     \global\advance\eql@line@interline@\skip@
2071     \prevdepth\dimen@
2072   \fi\fi
2073 \egroup
2074 }
2075 \def\eqlvspace@discardable#1{%
2076   \skip@\glueexpr#1\relax
2077   \ifnum\eql@row@=\@ne
2078     \global\eql@appendexpand\eqldisplay@container{%
2079       \advance\eql@abovespace@\the\skip@\relax}%
2080   \else\ifnum\eql@row@>\eql@totalrows@
2081     \global\eql@appendexpand\eqldisplay@container{%
2082       \advance\eql@belowspace@\the\skip@\relax}%
2083   \else
2084     \vskip\skip@
2085     \global\advance\eql@line@interline@\skip@
2086   \fi\fi
2087 \egroup
2088 }

```

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.

```

2089 \newbox\eql@strutbox@
2090 \def\eql@strut@depth{.3}
2091 \def\eql@strut{\copy\eql@strutbox@}
2092 \let\eql@strut@cell\eql@strut
2093 \let\eql@strut@tag\eql@strut
2094 \def\eql@strut@make{%
2095   \setbox\eql@strutbox@\hbox{%
2096     \@tempdimb\dimexpr
2097       \eql@strut@depth\normalbaselineskip+.5\normallineskiplimit\relax
2098     \@tempdima\dimexpr
2099       \normalbaselineskip-\normallineskiplimit-\@tempdimb\relax
2100     \vrule\@height\@tempdima\@depth\@tempdimb\@width\z@

```

```

2101 }
2102 }
2103 \AtBeginDocument{\eql@strut@make}

```

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

```

2104 \def\eql@spread@set{%
2105   \ifdefined\eql@spread@reset
2106     \lineskip\normallineskip
2107     \lineskiplimit\normallineskiplimit
2108     \baselineskip\normalbaselineskip
2109   \fi
2110   \eql@spread@\dimexpr\glueexpr\eql@spread@val\relax
2111     +\normalbaselineskip-\baselineskip\relax
2112   \ifdim\eql@spread@>\z@
2113     \openup\eql@spread@
2114     \ifdefined\spread@equation
2115       \let\spread@equation\@empty
2116     \fi
2117   \fi
2118 }

```

7.4 Entry and Exit

TODO: describe

```

2119 \let\eql@beamerbasecolor@fix\@empty
2120 \AddToHook{package/beamerbasecolor/after}{%
2121   \def\eql@beamerbasecolor@fix{%
2122     \donotcolorouterdisplaymaths
2123     \donotcoloroutermaths
2124     \beamer@setdisplaymathcolor
2125   }%
2126 }

```

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

```

2127 \newskip\eql@abovespace@
2128 \newskip\eql@belowspace@

```

`\eql@display@enter`

```

2129 \def\eql@display@enter{%
2130   \if@noskipsec\leavevmode\par\fi
2131   \ifvmode
2132     \eql@prevdepth@\prevdepth
2133     \nointerlineskip
2134     \noindent
2135   \else
2136     \eql@prevdepth@\maxdimen
2137   \fi
2138   \eql@beamerbasecolor@fix
2139 }

```

`\eql@display@adjust`

```

2140 \def\eql@display@adjust{%
2141   \ifdefined\eql@display@linewidth

```

```

2142 \displaywidth\glueexpr\eqldisplay@linewidth\relax
2143 \advance\displaywidth-\displayindent
2144 \fi
2145 \ifdefined\eqldisplay@marginleft
2146 \advance\displaywidth\displayindent
2147 \displayindent\glueexpr\eqldisplay@marginleft\relax
2148 \advance\displaywidth-\displayindent
2149 \fi
2150 \ifdefined\eqldisplay@marginright
2151 \advance\displaywidth-\glueexpr\eqldisplay@marginright\relax
2152 \fi
2153 \ifdim\displaywidth<\z@
2154 \displaywidth\z@
2155 \fi
2156 }

```

\eqldisplay@init

```

2157 \def\eqldisplay@init{%
2158 \let\eqldisplay@restore\eqldisplay@restore@active
2159 \let\displaybreak\eqldisplaybreak
2160 \let\eqlvspace@org\vspace
2161 \let\vspace\eqlvspace
2162 \let\eqncontrol\eql@control
2163 \let\eqldisplay@injectbefore\@empty
2164 \let\eqldisplay@injectafter\@empty
2165 \let\eqnpunct\eql@punct@setnext
2166 \eql@spread@set
2167 \eql@strut@make
2168 \let\eql@frame@cmd\@undefined
2169 }

```

\eqldisplay@print

```

2170 \def\eqldisplay@print{%
2171 \eql@punct@top@set
2172 \let\eqldisplay@container\@empty
2173 \eqldisplay@firstavail\z@
2174 \eqldisplay@aboveextend\z@
2175 \eqldisplay@belowextend\z@
2176 \global\let\eql@interline@container\eql@interline@container@clear
2177 }

```

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

```

2178 \def\eqldisplay@halign@init#1{%
2179 \eql@row\z@
2180 \eql@prevgraf\prevgraf
2181 \everycr{\noalign{%
2182 \global\advance\eql@row\@one
2183 \prevgraf\numexpr\prevgraf+\@one\relax
2184 #1%
2185 }}%
2186 }

```

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

```

2187 \def\eqldisplay@halign@start{%
2188 \prevgraf\numexpr\eql@prevgraf+\@one\relax

```

```

2189 \ifdim\eql@prevdepth@=\maxdimen\else
2190   \prevdepth\eql@prevdepth@
2191 \fi
2192 \ifdim\prevdepth=-\@m\p@\else
2193   \ifdefined\eql@display@height
2194     \skip@\baselineskip
2195     \advance\skip@-\glueexpr\eql@display@height\relax
2196     \advance\skip@-\prevdepth\relax
2197     \ifdim\skip@<\lineskiplimit
2198       \skip@\lineskip
2199     \fi
2200     \advance\skip@-\eql@spread@\relax
2201     \vskip\skip@
2202     \nointerlineskip
2203   \else
2204     \vskip-\eql@spread@\relax
2205   \fi
2206 \fi
2207 }

```

TODO: describe

```

2208 \def\eql@display@vspace{%
2209   \advance\abovedisplayskip\eql@abovespace@
2210   \advance\belowdisplayskip\eql@belowspace@
2211 }

```

TODO: describe

```

2212 \def\eql@display@vspace@native{%
2213   \advance\abovedisplayskip\eql@abovespace@
2214   \advance\belowdisplayskip\eql@belowspace@
2215   \advance\abovedisplayshortskip\eql@abovespace@
2216   \advance\belowdisplayshortskip\eql@belowspace@
2217 }

```

TODO: describe

```

2218 \def\eql@display@penalty{%
2219   \ifnum\eql@displaybreak@postpen@=\@MM\else
2220     \postdisplaypenalty\eql@displaybreak@postpen@
2221   \fi
2222   \ifnum\eql@displaybreak@open@=\@MM\else
2223     \postdisplaypenalty\eql@displaybreak@open@
2224   \fi
2225   \ifnum\eql@displaybreak@prepen@=\@MM\else
2226     \predisplaypenalty\eql@displaybreak@prepen@
2227   \fi
2228 }

```

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

```

2229 \def\eql@display@halign@end{%
2230   \eql@interline@container
2231   \eql@display@injectbefore
2232   \global\eql@prevgraf@\numexpr\prevgraf+\@ne\relax
2233   \ifdefined\eql@display@depth
2234     \prevdepth\glueexpr\eql@display@depth\relax
2235   \fi
2236 }

```

`\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?

```

2237 \def\eql@display@close{%
2238   \eql@display@container
2239   \ifdim\eql@display@firstavail@<\z@
2240     \eql@display@firstavail@>\z@
2241   \fi
2242   \eql@skip@mode@leave@>\z@
2243   \ifdim\eql@prevdepth@=\maxdimen
2244     \ifdim\predisplaysize=-\maxdimen
2245       \eql@skip@mode@above@\eql@skip@mode@cont@above\relax
2246       \eql@skip@mode@below@\eql@skip@mode@cont@below\relax
2247     \else
2248       \eql@skip@mode@above@>\z@
2249       \eql@skip@mode@below@>\z@
2250       \advance\eql@display@firstavail@>\displayindent
2251       \ifdim\eql@display@firstavail@>\predisplaysize
2252         \ifcase\eql@skip@mode@short\relax
2253         \or
2254           \eql@skip@mode@above@>\@ne
2255         \or
2256           \eql@skip@mode@above@>\@ne
2257           \ifnum\eql@totalrows@=\@ne
2258             \eql@skip@mode@below@>\@ne
2259           \fi
2260         \or
2261           \eql@skip@mode@above@>\@ne
2262           \eql@skip@mode@below@>\@ne
2263         \fi
2264       \fi
2265     \fi
2266   \else
2267     \ifdim\eql@prevdepth@=-\@m\p@
2268       \eql@skip@mode@above@\eql@skip@mode@top@above\relax
2269       \eql@skip@mode@below@\eql@skip@mode@top@below\relax
2270     \else
2271       \eql@skip@mode@above@\eql@skip@mode@par@above\relax
2272       \eql@skip@mode@below@\eql@skip@mode@par@below\relax
2273     \fi
2274   \fi
2275   \ifcase\eql@skip@mode@above@
2276   \or\or\or
2277     \predisplaypenalty\z@
2278   \or
2279     \predisplaypenalty\z@
2280   \fi
2281   \ifcase\eql@skip@mode@below@
2282   \or\or\or
2283     \eql@skip@mode@leave@>\@ne
2284   \or
2285     \eql@skip@mode@leave@>\tw@
2286   \fi
2287   \ifdefined\eql@skip@force@above
2288     \eql@skip@mode@above@\eql@skip@force@above\relax
2289   \fi
2290   \ifdefined\eql@skip@force@below
2291     \eql@skip@mode@below@\eql@skip@force@below\relax

```

```

2292 \fi
2293 \ifdefined\eq@skip@force@leave
2294   \eq@skip@mode@leave@\eq@skip@force@leave\relax
2295 \fi
2296 \ifnum\eq@skip@mode@leave@>\z@
2297   \postdisplaypenalty\z@
2298 \fi
2299 \ifcase\eq@skip@mode@above@
2300   \abovedisplayskip\glueexpr\eq@skip@long@above\relax
2301 \or
2302   \abovedisplayskip\glueexpr\eq@skip@short@above\relax
2303 \or
2304   \abovedisplayskip\glueexpr\eq@skip@cont@above\relax
2305 \or
2306   \abovedisplayskip\glueexpr\eq@skip@par@above\relax
2307 \or
2308   \abovedisplayskip\glueexpr\eq@skip@top@above\relax
2309 \or
2310   \abovedisplayskip\z@skip
2311 \or
2312   \abovedisplayskip\glueexpr\eq@skip@med@above\relax
2313 \or
2314   \abovedisplayskip\glueexpr\eq@skip@custom@above\relax
2315 \fi
2316 \ifcase\eq@skip@mode@below@
2317   \belowdisplayskip\glueexpr\eq@skip@long@below\relax
2318 \or
2319   \belowdisplayskip\glueexpr\eq@skip@short@below\relax
2320 \or
2321   \belowdisplayskip\glueexpr\eq@skip@cont@below\relax
2322 \or
2323   \belowdisplayskip\glueexpr\eq@skip@par@below\relax
2324 \or
2325   \belowdisplayskip\glueexpr\eq@skip@top@below\relax
2326 \or
2327   \belowdisplayskip\z@skip
2328 \or
2329   \belowdisplayskip\glueexpr\eq@skip@med@below\relax
2330 \or
2331   \belowdisplayskip\glueexpr\eq@skip@custom@below\relax
2332 \fi
2333 \global\eq@skip@mode@leave@\eq@skip@mode@leave@
2334 \eq@interline@container
2335 \advance\eq@belowspace@\eq@vspaceskip@
2336 \eq@display@penalty
2337 \eq@display@vspace
2338 \skip@\glueexpr\eq@skip@tag@above\relax
2339 \ifdim\skip@>\abovedisplayskip
2340   \skip@\abovedisplayskip
2341 \fi
2342 \advance\abovedisplayskip-\eq@display@aboveextend@\relax
2343 \ifdim\abovedisplayskip<\skip@
2344   \abovedisplayskip\skip@
2345 \fi
2346 \skip@\glueexpr\eq@skip@tag@below\relax
2347 \ifdim\skip@>\belowdisplayskip
2348   \skip@\belowdisplayskip
2349 \fi

```

```

2350 \ifdim\eqldisplay@belowextend@>\z@
2351 \advance\belowdisplayskip-\eqldisplay@belowextend@relax
2352 \ifdim\belowdisplayskip<\skip@
2353 \belowdisplayskip\skip@
2354 \fi
2355 \fi
2356 }

```

TODO: describe

```

2357 \def\eqldisplay@leave{%
2358 \prevgraf\eql@prevgraf@
2359 \ifcase\eql@skip@mode@leave@
2360 \or
2361 \endgraf
2362 \or
2363 \endgraf
2364 \prevdepth-\@m\p@
2365 \fi
2366 }

```

TODO: describe

```

2367 \def\eqldisplay@nest{%
2368 \let\displaybreak\eqldisplaybreak@default
2369 \let\intertext\eql@intertext@default
2370 \let\vspace\eql@vspace@org
2371 }

```

TODO: describe **TODO:** box version?! (but also consider nesting)

```

2372 \def\eqldisplay@restore@active{%
2373 \let\label\eql@label@org
2374 \let\tag\eql@tag@default
2375 \let\raisetag\eql@raisetag@default
2376 \let\displaybreak\eqldisplaybreak@default
2377 \let\intertext\eql@intertext@default
2378 \let\vspace\eql@vspace@org
2379 \ifdefined\eql@amp@mode
2380 \let\&\eql@amp@org
2381 \fi
2382 \let\eqnpunct\eql@punct@adopt
2383 \let\eql@punct@block\@undefined
2384 \let\eqldisplay@restore\@empty
2385 }

```

TODO: describe

```

2386 \let\eqldisplay@restore\@empty
2387 \eql@append\@arrayparboxrestore{%
2388 \eqldisplay@restore
2389 \ifdefined\eql@ampproof@active
2390 \eql@amp@prevert
2391 \fi
2392 \@displayfalse
2393 }

```

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)

```
2394 \def\eql@stack@enable{%
2395   \let\eql@stack@save@equations\eql@stack@save@equations@
2396   \let\eql@stack@save@box\eql@stack@save@box@
2397 }
```

TODO: describe

```
2398 \let\eql@stack@save@equations\eql@stack@enable
2399 \let\eql@stack@save@box\eql@stack@enable
2400 \let\eql@stack@restore\@empty
```

TODO: describe

```
2401 \def\eql@stack@save@reg#1{\global#1\the#1\relax}
2402 \def\eql@stack@save@let#1#2{\global\let\noexpand#2\noexpand#1}
```

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

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

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

```
2403 \def\eql@stack@save@equations@{%
2404   \let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2405   \let\eql@stack@cell@container\eql@cell@container
2406   \let\eql@stack@tags@container\eql@tags@container
2407   \let\eql@stack@interline@container\eql@interline@container
2408   \let\eql@stack@dimensions@tab\eql@dimensions@tab
2409   \let\eql@stack@block@container\eql@display@container
2410   \let\eql@stack@punct@top\eql@punct@top
2411   \edef\eql@stack@restore{%
2412     \global\if@eqnsw\noexpand\@eqnswtrue\else\noexpand\@eqnswfalse\fi
2413     \eql@stack@save@let\eql@stack@numbering@eqnswinit\eql@numbering@eqnswinit
2414     \eql@stack@save@let\eql@stack@cell@container\eql@cell@container
2415     \eql@stack@save@let\eql@stack@tags@container\eql@tags@container
2416     \eql@stack@save@let\eql@stack@interline@container\eql@interline@container
2417     \eql@stack@save@let\eql@stack@dimensions@tab\eql@dimensions@tab
2418     \eql@stack@save@let\eql@stack@block@container\eql@display@container
2419     \eql@stack@save@let\eql@stack@punct@top\eql@punct@top
2420     \eql@stack@save@reg\eql@column@
2421     \eql@stack@save@reg\eql@totalcolumns@
2422     \eql@stack@save@reg\eql@line@avail@
2423     \eql@stack@save@reg\eql@line@pos@
2424     \eql@stack@save@reg\eql@line@width@
2425     \eql@stack@save@reg\eql@line@depth@
2426     \eql@stack@save@reg\eql@line@height@
2427     \eql@stack@save@reg\eql@line@prevdepth@
2428     \eql@stack@save@reg\eql@line@interline@
2429     \eql@stack@save@reg\eql@totalheight@
2430     \eql@stack@save@reg\eql@tagwidth@max@
2431     \eql@stack@save@reg\eql@tagpos@row@
2432     \eql@stack@save@reg\eql@row@
2433     \eql@stack@save@reg\eql@tagrows@
2434   }%
2435 }
```


TODO: describe

```
2436 \def\eql@stack@save@box@{%
2437   \let\eql@stack@cell@container\eql@cell@container
2438   \edef\eql@stack@restore{%
2439     \eql@stack@save\let\eql@stack@cell@container\eql@cell@container
2440     \eql@stack@save@reg\eql@row@
2441   }%
2442 }
```

8 Multi-Line Support

TODO: describe

8.1 Measure Support

TODO: describe

```
2443 \def\eql@measure@init#1#2{%
2444   \eql@dimensions@reset
2445   \let\eql@display@container\@empty
2446   \eql@numbering@measure@init
2447   \eql@row@\z@
2448   \eql@totalheight@\z@
2449   \eql@totalrows@\@M
2450   \eql@line@prevdepth@-\@m\p@
2451   \eql@line@interline@\z@
2452   \tabskip\z@skip
2453   \everycr{\noalign{%
2454     \global\advance\eql@row@\@ne
2455     #1%
2456   }}%
2457   \eql@punct@top@set
2458   \global\let\eql@interline@container\eql@interline@container@clear
2459   \eql@measure@savestate
2460   \eql@multi@cr@let{#2}%
2461 }
```

TODO: describe

```
2462 \def\eql@measure@tag{%
2463   \eql@tagwidth@\z@
2464   \ifdefined\eql@numbering@multi
2465     \if@eqnsw
2466       \eql@tags@container
2467       \eql@tagbox@make\eql@composetag@measure
2468       \ifdefined\eql@tagpos@reserve\else
2469         \eql@tagwidth@\z@
2470       \fi
2471     \fi
2472   \fi
2473 }
```

TODO: describe

```
2474 \def\eql@measure@endrow{%
2475   \ifdim\eql@line@prevdepth@=-\@m\p@\else
```

```

2476 \dimen@ \dimexpr \baselineskip - \eqlline@height@ - \eqlline@prevdepth@ \relax
2477 \ifdim \dimen@ < \lineskiplimit
2478 \dimen@ \lineskip
2479 \fi
2480 \advance \eqlline@interline@ \dimen@
2481 \fi
2482 \eqldimensions@endrow
2483 \ifdim \eqlltagwidth@ > \eqlltagwidth@max@
2484 \global \eqlltagwidth@max@ \eqlltagwidth@
2485 \fi
2486 \ifdim \eqlltagwidth@ > \z@
2487 \global \advance \eqlltagrows@ \@ne
2488 \fi
2489 \global \advance \eqlltotalheight@ \dimexpr
2490 \eqllline@interline@ + \eqllline@height@ + \eqllline@depth@
2491 \global \eqllline@interline@ \z@
2492 \global \eqllline@prevdepth@ \eqllline@depth@
2493 }

```

TODO: describe

```

2494 \def \eqllmeasure@close{%
2495 \advance \eqllrow@ - \tw@
2496 \eqlltotalrows@ \eqllrow@
2497 \ifnum \eqlltotalrows@ > \z@
2498 \eqldimensions@get \@ne
2499 \eqlltopheight@ \dimexpr \eqllline@height@ + \eqllline@interline@ \relax
2500 \eqldimensions@get \eqlltotalrows@
2501 \eqllbottomdepth@ \eqllline@depth@
2502 \fi
2503 \eqllnumbering@measure@blocktag
2504 \begingroup
2505 \eqlltags@container
2506 \if@eqnsw
2507 \eqlltagbox@make \eqllcomposetag@measure
2508 \ifdefined \eqlltagpos@reserve \else
2509 \eqlltagwidth@ \z@
2510 \fi
2511 \eqldimensions@saveblocktag
2512 \else
2513 \eqldimensions@savenoblocktag
2514 \eqllnumbering@warnunused
2515 \fi
2516 \endgroup
2517 \eqldimensions@get \z@
2518 \eqllmeasure@restorestate
2519 }

```

measure@restorestate

eqllmeasure@savestate

```

2520 \let \eqllmeasure@restorestate \empty
2521 \def \eqllmeasure@savestate{%
2522 \begingroup
2523 \def \elt##1{%
2524 \global \csname c@##1 \endcsname \the \csname c@##1 \endcsname}%
2525 \global \edef \gtempa { \cl@ckpt}%
2526 \endgroup
2527 \let \eqllmeasure@restorestate \gtempa
2528 }

```

8.2 Line Breaks

TODO: describe

`\eq@multi@cr`

```
2529 \def\eq@multi@cr{%
2530   \let\eq@punct@term\eq@false
2531   \let\eq@class@rel@composed\@empty
2532   \eq@ampprotect\eq@multi@cr@test\eq@multi@cr@process}
```

TODO: describe

```
2533 \def\eq@multi@cr@test@setopt{%
2534   \let\eq@multi@cr@test\eq@multi@cr@testopt}
2535 \def\eq@multi@cr@test@setall{%
2536   \let\eq@multi@cr@test\eq@multi@cr@testall}
```

`\eq@multi@cr@testopt` **TODO:** describe

```
2537 \def\eq@multi@cr@testopt#1{\eq@teststaropt@tight
2538   {\eq@displaybreak@star\@M\eq@multi@cr@testopt@set{#1}}
2539   {\eq@multi@cr@testopt@set{#1}}{Opt}}
2540 \def\eq@multi@cr@testopt@set#1[#2]{\eq@vspace@add{#2}#1}
```

`\eq@multi@cr@testall` **TODO:** describe

`\lti@cr@testall@parse`

```
2541 \def\eq@multi@cr@testall{\eq@parseopt@cr\eq@multi@cr@testall@parse}
2542 \def\eq@multi@cr@testall@parse{%
2543   \ifx\eq@parseopt@token%
2544     \let\eq@parseopt@next\eq@multi@cr@parse@vspace
2545   \fi
2546   \ifx\eq@parseopt@token*%
2547     \let\eq@parseopt@next\eq@multi@cr@parse@star
2548   \fi
2549   \ifx\eq@parseopt@token.%
2550     \let\eq@parseopt@next\eq@parseopt@punctpass
2551   \fi
2552   \ifx\eq@parseopt@token,%
2553     \let\eq@parseopt@next\eq@parseopt@punctpass
2554   \fi
2555   \ifx\eq@parseopt@token~%
2556     \let\eq@parseopt@next\eq@parseopt@punctpass
2557   \fi
2558   \ifx\eq@parseopt@token'%
2559     \let\eq@parseopt@next\eq@parseopt@punctnext
2560   \fi
2561   \ifx\eq@parseopt@token!%
2562     \let\eq@parseopt@next\eq@parseopt@punctterm
2563   \fi
2564   \ifx\eq@parseopt@token/%
2565     \let\eq@parseopt@next\eq@multi@cr@parse@break
2566   \fi
2567   \ifx\eq@parseopt@token=%
2568     \let\eq@parseopt@next\eq@parseopt@relsyb
2569   \fi
2570   \ifx\eq@parseopt@token;%
2571     \let\eq@parseopt@next\eq@parseopt@relcont
2572   \fi
```

```

2573 \ifx\eql@parseopt@token:%
2574   \let\eql@parseopt@next\eql@parseopt@relstart
2575 \fi
2576 \ifx\eql@parseopt@token|%
2577   \let\eql@parseopt@next\eql@parseopt@relord
2578 \fi
2579 \ifx\eql@parseopt@token?%
2580   \let\eql@parseopt@next\eql@multi@cr@parse@rel
2581 \fi
2582 \ifx\eql@parseopt@token&%
2583   \let\eql@parseopt@next\eql@parseopt@end
2584 \fi
2585 }
2586 \def\eql@multi@cr@parse@vspace[#1]{\eql@vspace@add{#1}\eql@parseopt@peek}
2587 \def\eql@multi@cr@parse@star#1{\eql@displaybreak@star\@M\eql@parseopt@peek}
2588 \def\eql@multi@cr@parse@break{\numbernext\eql@parseopt@punctclear}
2589 \def\eql@multi@cr@parse@rel#1#2{%
2590   \def\eql@tmp{#2}%
2591   \ifx\eql@tmp\eql@relax\else
2592     \eql@punct@next@clear
2593     \ifdefined\eql@multi@cr@relnext\numbernext\fi
2594   \fi
2595   \ifdefined\eql@multi@linesmode
2596     \ifx\eql@tmp\@empty
2597       \def\eql@class@rel@composed{\eql@shape@cont}%
2598     \else
2599       \def\eql@class@rel@composed{\eql@shape@rel#2}%
2600     \fi
2601   \else
2602     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
2603   \fi
2604   \eql@parseopt@end}

```

eql@multi@cr@process

```

2605 \def\eql@multi@cr@process{%
2606   \ifdefined\eql@punct@term\eql@punct@apply@top\fi
2607   \edef\eql@tmp{%
2608     \unexpanded{%
2609       \eql@multi@endline
2610       \cr
2611       \eql@multi@cr@interline
2612     }%
2613     \unexpanded\expandafter{\eql@class@rel@composed}%
2614   }%
2615   \eql@tmp
2616 }

2617 \def\eql@multi@cr@interline{%
2618   \noalign{%
2619     \eql@interline@container
2620     \eql@display@injectbefore
2621     \ifnum\eql@displaybreak@pen@=\@MM
2622       \penalty\interdisplaylinepenalty
2623     \else
2624       \penalty\eql@displaybreak@pen@
2625     \fi
2626     \vskip\eql@vspaceskip@
2627     \global\advance\eql@line@interline@\eql@vspaceskip@

```

```

2628 \eqldisplay@injectafter
2629 \global\let\eql@interline@container\eql@interline@container@clear
2630 }%
2631 }

```

\eql@multi@cr@let

```

2632 \def\eql@multi@cr@let#1{%
2633 \let\\eql@multi@cr
2634 \let\eql@multi@endline#1%
2635 }

```

8.3 Intertext

TODO: describe

TODO: revert in everymath?

```

2636 \def\eql@intertext@default{\eql@error{Invalid use of \string\intertext}}
2637 \eql@amsmath@let\intertext\eql@intertext@default

```

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

```

2638 \def\eql@intertext@process{%
2639 \eql@multi@endline
2640 \cr
2641 \ifmeasuring@
2642 \expandafter\@gobble
2643 \else
2644 \expandafter\eql@intertext@print
2645 \fi
2646 }

```

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

```

2647 \def\eql@intertext@print#1{%
2648 \noalign{%
2649 \eqldisplay@halign@end
2650 \let\eql@skip@force@below\z@
2651 \let\eql@skip@force@above\z@
2652 \eql@setkeys{intertext}\eql@intertext@opt
2653 \openup-\eql@spread@
2654 \penalty\postdisplaypenalty
2655 \ifcase\eql@skip@force@below\relax
2656 \advance\eql@vspaceskip@\glueexpr\eql@skip@long@below\relax
2657 \or
2658 \advance\eql@vspaceskip@\glueexpr\eql@skip@short@below\relax
2659 \or
2660 \advance\eql@vspaceskip@\glueexpr\eql@skip@cont@below\relax
2661 \or
2662 \advance\eql@vspaceskip@\glueexpr\eql@skip@par@below\relax
2663 \or
2664 \advance\eql@vspaceskip@\glueexpr\eql@skip@top@below\relax
2665 \or
2666 \advance\eql@vspaceskip@\z@skip
2667 \or
2668 \advance\eql@vspaceskip@\glueexpr\eql@skip@med@below\relax
2669 \or
2670 \advance\eql@vspaceskip@\glueexpr\eql@skip@custom@below\relax

```

```

2671 \fi
2672 \vskip\eq\vspaceskip@
2673 \global\let\eq\interline@container\eq\interline@container@clear
2674 \vbox{%
2675 \@parboxrestore
2676 \ifdim
2677 \ifdim\@totalleftmargin=\z@\linewidth\else-\maxdimen\fi=\columnwidth
2678 \else
2679 \parshape\@ne
2680 \@totalleftmargin\linewidth
2681 \fi
2682 \noindent
2683 \prevgraf\eq\prevgraf@
2684 \ignorespaces
2685 #1%
2686 \par
2687 \global\eq\prevgraf@\prevgraf
2688 }%
2689 \penalty\predisplaypenalty
2690 \ifcase\eq\skip@force@above\relax
2691 \vskip\glueexpr\eq\skip@long@above\relax
2692 \or
2693 \vskip\glueexpr\eq\skip@short@above\relax
2694 \or
2695 \vskip\glueexpr\eq\skip@cont@above\relax
2696 \or
2697 \vskip\glueexpr\eq\skip@par@above\relax
2698 \or
2699 \vskip\glueexpr\eq\skip@top@above\relax
2700 \or
2701 \vskip\z@skip
2702 \or
2703 \vskip\glueexpr\eq\skip@med@above\relax
2704 \or
2705 \vskip\glueexpr\eq\skip@custom@above\relax
2706 \fi
2707 % \eq\prevdepth@\maxdimen
2708 \eq\prevdepth@\z@
2709 \eq\display@halign@start
2710 }
2711 }

```

TODO: describe

```

2712 \newenvironment{eq\intertext}{%
2713 \eq\testopt@tight\eq\intertext@}%
2714 }{%
2715 \aftergroup\eq\intertext@after
2716 \ignorespacesafterend
2717 }

```

TODO: describe

```

2718 \def\eq\intertext@env{intertext}
2719 \def\eq\intertext@[#1]{%
2720 \global\def\eq\intertext@opt{#1}%
2721 \ifx\@currenvir\eq\intertext@env
2722 \def\eq\scan@call{\eq\intertext@inject\eq\scan@end}%
2723 \expandafter\eq\scan@env
2724 \else

```

```

2725 \expandafter\eql@intertext@process
2726 \fi
2727 }

```

TODO: describe

```

2728 \def\eql@intertext@inject{%
2729 \global\edef\eql@intertext@after{%
2730 \noexpand\eql@intertext@process{%
2731 \ifx\eql@scan@body\eql@scan@body@dump
2732 \eql@scan@body@dump
2733 \else
2734 \noexpand\scantokens{\eql@scan@body@dump}%
2735 \fi
2736 }%
2737 }%
2738 }

```

8.4 Line Marks

TODO: describe

```

2739 \def\eql@markline@pos@below{below}
2740 \def\eql@markline@pos@bottom{bottom}
2741 \def\eql@markline@pos@baseline{baseline}
2742 \let\eql@markline@pos\eql@markline@pos@baseline
2743 \let\eql@markline@shift\z@
2744 \def\eql@markline@qed{\ifdefined\qedsymbol\qedsymbol\else QED\fi}
2745 \def\eql@markline@symbol{}

```

TODO: describe

```

2746 \def\eql@markline@select#1{%
2747 \let\eql@markline@shift\z@
2748 \eql@setkeys{markline}{#1}%
2749 \eql@markline@print
2750 }

```

TODO: describe

```

2751 \def\eql@markline@print{%
2752 \dimen@=\dimexpr\eql@markline@shift\relax
2753 \ifx\eql@markline@pos\eql@markline@pos@below
2754 \ifdim\dimen@=\z@\else
2755 \penalty\@M
2756 \vskip-\dimen@
2757 \fi
2758 \nointerlineskip
2759 \penalty\@M
2760 \vbox{\hfill\hbox{\eql@markline@symbol}}}%
2761 \else
2762 \ifx\eql@markline@pos\eql@markline@pos@baseline
2763 \advance\dimen@\prevdepth
2764 \fi
2765 \setbox\z@\hbox{\raise\dimen@\hbox{\eql@markline@symbol}}}%
2766 \dimen@\prevdepth
2767 \ht\z@\z@
2768 \dp\z@\z@
2769 \nointerlineskip

```

```

2770 \penalty\@M
2771 \vbox{\hfill\box\z@}%
2772 \prevdepth\dimen@
2773 \fi
2774 }

```

TODO: describe

```

2775 \def\eql@markline@inject#1{%
2776 \let\eql@markline@push\eql@false
2777 \ifx\eql@markline@pos\eql@markline@pos@below\else
2778 \ifdefined\eql@tagsleft\else
2779 \ifx\eql@equations@main\eql@multi@main
2780 \ifdefined\eql@numbering@multi
2781 \if@eqnsw
2782 \let\eql@markline@push\eql@true
2783 \fi
2784 \else
2785 \ifnum\eql@row@=\eql@tagpos@row@
2786 \let\eql@markline@push\eql@true
2787 \fi
2788 \fi
2789 \else
2790 \if@eqnsw
2791 \let\eql@markline@push\eql@true
2792 \fi
2793 \fi
2794 \fi
2795 \fi
2796 \ifdefined\eql@markline@push
2797 \global\eql@append\eql@interline@container{%
2798 \eql@append\eql@display@injectbefore{\eql@markline@select{push,#1}}}%
2799 \else
2800 \global\eql@append\eql@interline@container{%
2801 \eql@append\eql@display@injectbefore{\eql@markline@select{#1}}}%
2802 \fi
2803 }

```

TODO: describe

```

2804 \def\eql@markline@amsthm@opt[#1]{\eql@markline@inject{qed,#1}}
2805 \def\eql@markline@amsthm@staropt[#1]{\eql@markline@inject{qed,push,#1}}
2806 \def\eql@markline@amsthm@qed{\eql@teststaropt@tight
2807 \eql@markline@amsthm@staropt\eql@markline@amsthm@opt{}}
2808 \def\eql@markline@amsthm@register#1{\eql@letcs{#1@qed}\eql@markline@amsthm@qed}
2809 \def\eql@markline@amsthm@move#1#2{%
2810 \AddToHook{package/amsthm/after}{%
2811 \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:

```
2812 \newcount\eq\shape@pos@
2813 \newdimen\eq\shape@amount@
2814 \let\eq\shape@lastrow\eq\false
```

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

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

```
2815 \newdimen\eq\marginleft@
2816 \newdimen\eq\marginright@
2817 \newdimen\eq\marginleft@min@
2818 \newdimen\eq\centeroffset@
```

9.2 Shape Schemes

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

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

```
2819 \def\eq\shape@tab@default{default}
2820 \def\eq\shape@tab@left{left}
2821 \def\eq\shape@tab@center{center}
2822 \def\eq\shape@tab@right{right}
2823 \def\eq\shape@tab@first{first}
2824 \def\eq\shape@tab@hanging{hanging}
2825 \def\eq\shape@tab@steps{steps}
```

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

```
2826 \let\eq\shape@tab@def\eq\shape@tab@default
2827 \let\eq\shape@tab@\eq\shape@tab@default
2828 \let\eq\shape@tab@l\eq\shape@tab@left
2829 \let\eq\shape@tab@c\eq\shape@tab@center
2830 \let\eq\shape@tab@r\eq\shape@tab@right
2831 \let\eq\shape@tab@rc\eq\shape@tab@first
2832 \let\eq\shape@tab@indent\eq\shape@tab@first
2833 \let\eq\shape@tab@hang\eq\shape@tab@hanging
2834 \let\eq\shape@tab@lcr\eq\shape@tab@hanging
2835 \let\eq\shape@tab@outdent\eq\shape@tab@hanging
2836 \let\eq\shape@tab@lcr\eq\shape@tab@steps
```

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

```
2837 \let\eq\shape@mode\eq\shape@tab@default
```

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

```
2838 \def\eq\shape@set#1{%
2839   \ifcsname eq\shape@tab@#1\endcsname
2840     \expandafter\let\expandafter\eq\shape@mode
2841       \csname eq\shape@tab@#1\endcsname
2842   \else
2843     \eq\error{shape '#1' unknown: setting to default}%
2844     \let\eq\shape@mode\eq\shape@tab@default
2845   \fi
2846 }
```

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

```
2847 \def\eq@shape@layoutcenter@left{\eq@shape@pos@z@eq@shape@amount@z@}
2848 \def\eq@shape@layoutcenter@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2849 \def\eq@shape@layoutcenter@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2850 \let\eq@shape@layoutcenter@default\eq@shape@layoutcenter@center
2851 \def\eq@shape@layoutleft@left{\eq@shape@pos@z@eq@shape@amount@z@}
2852 \def\eq@shape@layoutleft@center{\eq@shape@pos@ne@eq@shape@amount@z@}
2853 \def\eq@shape@layoutleft@right{\eq@shape@pos@tw@eq@shape@amount@z@}
2854 \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):

```
2855 \def\eq@shape@layoutcenter@first{%
2856   \eq@shape@pos@z@
2857   \eq@shape@amount@z@
2858   \ifnum\eq@totalrows@>\@ne
2859     \ifnum\eq@row@=\@ne
2860       \eq@shape@amount@eq@indent@
2861     \fi
2862   \fi
2863 }
2864 \def\eq@shape@layoutleft@first{%
2865   \eq@shape@pos@z@
2866   \eq@shape@amount@z@
2867   \ifnum\eq@totalrows@>\@ne
2868     \ifnum\eq@row@=\@ne
2869       \eq@shape@amount@eq@indent@
2870     \fi
2871   \fi
2872 }
```

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:

```
2873 \def\eq@shape@layoutcenter@hanging{%
2874   \eq@shape@pos@z@
2875   \eq@shape@amount@eq@indent@
2876   \ifnum\eq@totalrows@>\@ne
2877     \ifnum\eq@row@=\@ne
2878       \eq@shape@amount@z@
2879     \fi
2880   \fi
2881 }
2882 \def\eq@shape@layoutleft@hanging{%
2883   \eq@shape@pos@z@
2884   \eq@shape@amount@z@
2885   \ifnum\eq@totalrows@>\@ne
2886     \ifnum\eq@row@=\@ne
2887       \eq@shape@amount@-eq@indent@
2888     \fi
2889   \fi
2890 }
```

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:

```

2891 \def\eql@shape@layoutcenter@steps{%
2892   \eql@shape@amount@z@
2893   \eql@shape@pos@\@ne
2894   \ifnum\eql@totalrows@>\@ne
2895     \ifnum\eql@row@=\@ne
2896       \eql@shape@pos@z@
2897     \fi
2898     \ifnum\eql@row@=\eql@totalrows@
2899       \eql@shape@pos@tw@
2900     \fi
2901   \fi
2902 }
2903 \def\eql@shape@layoutleft@steps{%
2904   \eql@shape@pos@z@
2905   \eql@shape@amount@z@
2906   \ifnum\eql@totalrows@>\@ne
2907     \ifnum\eql@row@=\@ne
2908       \eql@shape@amount@-\eql@indent@
2909     \fi
2910     \ifnum\eql@row@=\eql@totalrows@
2911       \eql@shape@amount@\eql@indent@
2912     \fi
2913   \fi
2914 }

```

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

```

2915 \let\eql@shape@eval\undefined
2916 \def\eql@shape@select{%
2917   \expandafter\let\expandafter\eql@shape@eval
2918     \csname eql@shape%
2919     @\ifdefined\eql@layoutleft layoutleft\else layoutcenter\fi
2920     @\eql@shape@mode\endcsname
2921 }

```

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

```

2922 \protected\def\eql@shape@alignleft{%
2923   \global\eql@append\eql@cell@container{\eql@shape@pos@z@}%
2924   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2925 \protected\def\eql@shape@aligncenter{%
2926   \global\eql@append\eql@cell@container{\eql@shape@pos@\@ne}%
2927   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2928 \protected\def\eql@shape@alignright{%
2929   \global\eql@append\eql@cell@container{\eql@shape@pos@tw@}%
2930   \eql@ampprotect\eql@shape@align@testpar\eql@shape@alignamount@opt}
2931 \def\eql@shape@align@testpar#1{%
2932   \eql@ifstar@tight{#1[\eql@indent@]}%
2933   {\eql@ifnextgobble@tight{!}{#1[-\eql@indent@]}%
2934   {\eql@testopt@tight{#1}\z@}}%
2935 \def\eql@shape@alignamount@opt[#1]{\eql@shape@alignamount@set{#1}}

```

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

```

2936 \protected\def\eql@shape@alignamount{%

```

```

2937 \eql@ampprotecttwo\eql@ifstar@tight
2938 \eql@shape@alignamount@set\eql@shape@alignamount@add}
2939 \def\eql@shape@alignamount@add#1{%
2940 \global\eql@appendexpand\eql@cell@container{%
2941 \advance\eql@shape@amount@the\glueexpr#1\relax\relax}}
2942 \def\eql@shape@alignamount@set#1{%
2943 \global\eql@appendexpand\eql@cell@container{%
2944 \eql@shape@amount@the\glueexpr#1\relax\relax}}
2945 \def\eql@shape@align@enable{%
2946 \let\shoveleft\eql@shape@alignleft
2947 \let\shovecenter\eql@shape@aligncenter
2948 \let\shoveright\eql@shape@alignright
2949 \let\shoveby\eql@shape@alignamount
2950 }

```

TODO: describe

```

2951 \protected\def\eql@shape@align@default{%
2952 \eql@warn@here{\shove...}%
2953 \eql@ampprotect\eql@shape@align@testpar\eql@gobbleopt}
2954 \protected\def\eql@shape@alignamount@default{%
2955 \eql@warn@here{\shove...}%
2956 \eql@ampprotecttwo\eql@ifstar@tight\@gobble\@gobble}
2957 \def\eql@shape@align@disable{%
2958 \let\shoveleft\eql@shape@align@default
2959 \let\shovecenter\eql@shape@align@default
2960 \let\shoveright\eql@shape@align@default
2961 \let\shoveby\eql@shape@alignamount@default
2962 }

```

9.3 Width Data

width@block@ (*dimen*)

```

2963 \newdimen\eql@tagwidth@block@
2964 \newdimen\eql@tagheight@block@
2965 \newdimen\eql@tagdepth@block@

```

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

```

2966 \let\eql@dimensions@tab\@empty

```

$\text{eql@dimensions@reset}$

```

2967 \def\eql@dimensions@reset{%
2968 \let\eql@dimensions@tab\@empty
2969 \eql@tagwidth@max@z@
2970 \eql@tagrows@z@
2971 }

```

$\text{eql@dimensions@add}$

```

2972 \def\eql@dimensions@add#1{%
2973 \global\eql@appendexpand\eql@dimensions@tab{#1}%
2974 }

```

$\text{eql@dimensions@addreg}$

```

2975 \def\eql@dimensions@addreg#1{#1\the#1\relax}

```

@dimensions@startrow

```
2976 \def\eqldimensions@startrow{%
2977   \eqldimensions@add{\eqldimensions@addreg\eqldrow@}%
2978 }
```

@dimensions@savecell

```
2979 \def\eqldimensions@savecell{%
2980   \eqldimensions@add{%
2981     \eqldimensions@addreg\eqldshape@pos@
2982     \eqldimensions@addreg\eqldcellwidth@
2983     \eqldimensions@addreg\eqldshape@amount@
2984     \noexpand\eqldimensions@cellcall
2985   }%
2986 }
```

@dimensions@savesep

```
2987 \def\eqldimensions@savesep{%
2988   \eqldimensions@add{\noexpand\eqldimensions@sepcall}%
2989 }
```

@dimensions@endrow

```
2990 \def\eqldimensions@endrow{%
2991   \eqldimensions@add{,%
2992     \eqldimensions@addreg\eqldtagwidth@
2993     \eqldimensions@addreg\eqldline@height@
2994     \eqldimensions@addreg\eqldline@depth@
2995     \eqldimensions@addreg\eqldline@interline@
2996   ;}%
2997 }
```

@dimensions@saveblocktag

```
2998 \def\eqldimensions@saveblocktag{%
2999   \eqldimensions@add{\eqldrow@0\relax,%
3000     \eqldtagwidth@block@\the\eqldtagwidth@\relax
3001     \eqldtagheight@block@\the\ht\eqldtagbox@\relax
3002     \eqldtagdepth@block@\the\dp\eqldtagbox@\relax
3003     \eqldimensions@addreg\eqldtagpos@shift@
3004     \let\noexpand\eqldtagpos@reserve\ifdefined\eqldtagpos@reserve
3005     \noexpand\eqldtrue\else\noexpand\eqldfalse\fi
3006   ;}%
3007   \global\eqldtagwidth@max@\eqldtagwidth@
3008   \global\eqldtaggrows@\@ne
3009 }
```

@dimensions@savenoblocktag

```
3010 \def\eqldimensions@savenoblocktag{%
3011   \eqldimensions@add{\eqldrow@0\relax,;%
3012 }
```

@eqldimensions@for

```
3013 \def\eqldimensions@for#1{%
3014   \def\eqldimensions@forcall{#1}%
3015   \expandafter\eqldimensions@forstep\eqldimensions@tab
3016 }
```

l@dimensions@forstep

```

3017 \def\eql@dimensions@forstep\eql@row@#1\relax#2,##3;%
3018   \eql@row@#1\relax
3019   \ifnum\eql@row@=\z@\else
3020     #3%
3021     \def\eql@dimensions@cells{##2}%
3022     \eql@dimensions@forall
3023     \expandafter\eql@dimensions@forstep
3024   \fi
3025 }

```

\eql@dimensions@get

```

3026 \def\eql@dimensions@get#1{%
3027   \eql@row@#1\relax
3028   \expandafter\eql@dimensions@getdef\expandafter{\the\eql@row@}%
3029   \expandafter\eql@dimensions@getparse\eql@dimensions@tab\@nil
3030 }

```

ql@dimensions@getdef

```

3031 \def\eql@dimensions@getdef#1{%
3032   \def\eql@dimensions@getparse
3033     ##1\eql@row@#1\relax##2,##3;##4\@nil{%
3034     ##3%
3035     \def\eql@dimensions@cells{##2}%
3036   }%
3037 }

```

\eql@colwidth@tab

```

3038 \let\eql@colwidth@tab\@empty

```

\eql@colwidth@get

```

3039 \def\eql@colwidth@get#1{%
3040   \ifcase\expandafter#1\eql@colwidth@tab\else\z@\fi
3041 }

```

\eql@colwidth@save

```

3042 \def\eql@colwidth@save#1{%
3043   \edef\eql@colwidth@tab{%
3044     \noexpand\or\the#1%
3045     \unexpanded\expandafter{\eql@colwidth@tab}%
3046   }%
3047 }

```

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

```

3048 \def\eql@dimensions@calc{%
3049   \eql@column@\z@
3050   \eql@line@pos@\z@

```

```

3051 \eql@line@possep@\z@
3052 \eql@line@avail@\eql@totalwidth@
3053 \eql@line@availsep@\eql@intercolumns@
3054 \eql@line@width@\z@
3055 \eql@line@widthsep@\z@
3056 \let\eql@dimensions@cellcall\eql@dimensions@calc@call
3057 \let\eql@dimensions@sepcall\eql@dimensions@calc@callsep
3058 \eql@dimensions@cells
3059 }

```

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

```

3060 \def\eql@dimensions@calc@callsep{%
3061   \advance\eql@line@possep@\@ne
3062 }%

```

`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?!)

```

3063 \def\eql@dimensions@calc@call{%
3064   \advance\eql@column@\@ne
3065   \ifnum\eql@totalcolumns@=\@ne
3066     \dimen@\eql@totalwidth@
3067   \else
3068     \dimen@\eql@colwidth@get\eql@column@\relax
3069   \fi
3070   \ifdim\eql@cellwidth@>\z@
3071     \ifdim\eql@line@width@=\z@
3072       \eql@line@avail@\eql@line@pos@
3073       \eql@line@availsep@\eql@line@possep@
3074       \ifcase\eql@shape@pos@
3075       \or
3076         \advance\eql@line@avail@\dimexpr
3077           (\dimen@-\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3078       \or
3079         \advance\eql@line@avail@\dimexpr\dimen@-\eql@cellwidth@\relax
3080       \fi
3081       \advance\eql@line@avail@\eql@shape@amount@
3082     \fi
3083     \eql@line@width@\eql@line@pos@
3084     \eql@line@widthsep@\eql@line@possep@
3085     \ifcase\eql@shape@pos@
3086     \advance\eql@line@width@\eql@cellwidth@
3087     \or
3088     \advance\eql@line@width@\dimexpr
3089       (\dimen@+\eql@cellwidth@+\eql@centeroffset@)/\tw@\relax
3090     \or
3091     \advance\eql@line@width@\dimen@
3092     \fi
3093     \advance\eql@line@width@\eql@shape@amount@
3094   \fi
3095   \advance\eql@line@pos@\dimen@
3096 }

```

9.4 Best Line Selection

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

```
3097 \let\eql@numbering@best@auto\eql@false
```

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

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

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

```
3098 \newcount\eql@numbering@best@row@
```

```
3099 \newdimen\eql@numbering@best@space@
```

```
3100 \let\eql@numbering@best@use\eql@false
```

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

```
3101 \def\eql@numbering@best@find{%
3102   \eql@numbering@best@row@ \z@
3103   \eql@numbering@best@space@ \z@
3104   \eql@dimensions@for{%
3105     \eql@dimensions@calc
3106     \ifdefined\eql@tagsleft
3107       \dimen@ \eql@line@avail@
3108     \else
3109       \dimen@ \dimexpr \eql@totalwidth@ - \eql@line@width@ \relax
3110     \fi
3111     \ifdim \dimen@ > \eql@numbering@best@space@
3112       \eql@numbering@best@row@ \eql@row@
3113       \eql@numbering@best@space@ \dimen@
3114     \fi
3115   }%
3116   \ifnum \eql@numbering@best@row@ > \z@
3117     \eql@tagpos@row@ \eql@numbering@best@row@
3118     \let\eql@tagpos@continuous\eql@false
3119     \eql@tagpos@prevrow@ \z@
3120   \fi
3121 }
```

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

```
3122 \def\eql@numbering@best@test#1{%
3123   \eql@dimensions@get#1%
3124   \eql@dimensions@calc
3125   \ifdefined\eql@tagsleft
3126     \dimen@ \dimexpr \eql@line@avail@
3127       + \eql@marginleft@ + \eql@line@availsep@ \eql@colsep@ \relax
3128   \else
3129     \dimen@ \dimexpr \displaywidth@ - \eql@line@width@
3130       - \eql@marginleft@ - \eql@line@widthsep@ \eql@colsep@ \relax
3131   \fi
3132   \ifdim \dimen@ < \eql@tagwidth@block@
3133     \let\eql@numbering@best@use\eql@true
3134   \fi
3135 }
```

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

```
3136 \def\eql@numbering@best@eval{%
3137   \ifdefined\eql@numbering@best@auto
3138     \ifdefined\eql@numbering@best@use\else
3139       \ifdefined\eql@numbering@multi\else
```



```

3140         \ifnum\eql@tagpos@row@>\z@
3141             \eql@numbering@best@test\eql@tagpos@row@
3142         \fi
3143         \ifnum\eql@tagpos@prevrow@>\z@
3144             \eql@numbering@best@test\eql@tagpos@prevrow@
3145         \fi
3146     \fi
3147 \fi
3148 \fi
3149 \ifdefined\eql@numbering@best@use
3150     \eql@numbering@best@find
3151 \fi
3152 }

```

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

```

3153 \def\eql@adjust@calc@tagmargin{%
3154     \ifdefined\eql@tagmargin@val
3155         \eql@tagmargin@\glueexpr\eql@tagmargin@val\relax
3156     \else
3157         \eql@tagmargin@\eql@tagwidth@max@
3158         \ifdim\eql@tagmargin@>\z@
3159             \advance\eql@tagmargin@-\eql@tagsepmin@
3160         \fi
3161     \fi

3162     \dimen@\eql@tagrows@\p@
3163     \ifnum\eql@totalrows@=\@ne
3164         \ifnum\eql@tagrows@=\@ne
3165             \advance\dimen@1sp\relax
3166         \fi
3167     \fi
3168     \ifdim\dimen@>\eql@totalrows@\eql@tagmargin@ratio@\else
3169         \eql@tagmargin@\z@
3170     \fi

3171     \@tempdima\dimexpr\displaywidth
3172         -\eql@totalwidth@-\eql@intercolumns@\eql@colsepmin@\relax
3173     \@tempdimb\dimexpr\@tempdima-\tw@\eql@tagmargin@\relax
3174     \ifdim\@tempdimb>\z@
3175         \ifdim\eql@tagmargin@threshold\@tempdima<\@tempdimb
3176             \eql@tagmargin@\z@
3177         \fi
3178     \fi
3179 }

```

9.6 Single Column

ql@adjust@calc@lines

```

3180 \def\eql@adjust@calc@lines{%
3181     \eql@totalcolumns@\@ne

```

```

3182 \eq@intercolumns@\z@
3183 \eq@colsep@\z@
3184 \ifdefined\eq@layoutleft
3185 \eq@marginleft@\glueexpr\eq@layoutleftmargin\relax
3186 \eq@marginleft@min@\glueexpr\eq@layoutleftmarginmin\relax
3187 \ifdim\eq@marginleft@<\eq@marginleft@min@
3188 \eq@marginleft@\eq@marginleft@min@
3189 \fi
3190 \dimen@\glueexpr\eq@layoutleftmarginmax\relax
3191 \ifdim\eq@marginleft@>\dimen@
3192 \eq@marginleft@\dimen@
3193 \fi
3194 \eq@marginright@\z@
3195 \eq@centeroffset@\z@
3196 \else
3197 \eq@adjust@calc@tagmargin
3198 \ifdefined\eq@paddingleft@val
3199 \eq@marginleft@\dimexpr
3200 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3201 -\glueexpr\eq@paddingleft@val\relax\relax
3202 \ifdim\eq@marginleft@<\z@
3203 \eq@marginleft@\z@
3204 \fi
3205 \else
3206 \eq@marginleft@\z@
3207 \fi
3208 \ifdefined\eq@paddingright@val
3209 \eq@marginright@\dimexpr
3210 (\displaywidth-\eq@totalwidth@-\eq@tagmargin@)/\tw@
3211 -\glueexpr\eq@paddingright@val\relax\relax
3212 \ifdim\eq@marginright@<\z@
3213 \eq@marginright@\z@
3214 \fi
3215 \else
3216 \eq@marginright@\z@
3217 \fi
3218 \ifdim\eq@tagmargin@>\z@
3219 \ifdefined\eq@tagsleft
3220 \ifdim\eq@marginleft@<\eq@tagsepmin@
3221 \eq@marginleft@\eq@tagsepmin@
3222 \fi
3223 \advance\eq@marginleft@\eq@tagmargin@
3224 \advance\eq@centeroffset@\eq@tagmargin@
3225 \else
3226 \ifdim\eq@marginright@<\eq@tagsepmin@
3227 \eq@marginright@\eq@tagsepmin@
3228 \fi
3229 \advance\eq@marginright@\eq@tagmargin@
3230 \advance\eq@centeroffset@-\eq@tagmargin@
3231 \fi
3232 \fi
3233 \eq@marginleft@min@\z@
3234 \eq@centeroffset@\dimexpr\eq@marginright@-\eq@marginleft@
3235 \ifdefined\eq@tagsleft+ \else -\fi \eq@tagmargin@ \relax
3236 \fi

3237 \eq@totalwidth@\dimexpr\displaywidth
3238 -\eq@marginleft@-\eq@marginright@\relax
3239 }

```

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:

```
3240 \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:

```
3241 \ifnum\eql@totalcolumns@<\thr@@
3242   \eql@totalcolumns@=\tw@
3243   \let\eql@columns@fulllength\eql@false
3244 \fi
```

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

```
3245 \eql@intercolumns@=\numexpr(\eql@totalcolumns@-\tw@)/\tw@\relax
```

Evaluate the minimum intercolumn space which we will need often:

```
3246 \eql@colsepmin@\glueexpr\eql@colsepmin@val\relax
```

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

```
3247 \ifdefined\eql@layoutleft
3248   \eql@marginleft@\glueexpr\eql@layoutleftmargin\relax
3249   \eql@marginleft@min@\glueexpr\eql@layoutleftmarginmin\relax
3250   \ifdim\eql@marginleft@<\eql@marginleft@min@
3251     \eql@marginleft@\eql@marginleft@min@
3252   \fi
3253 \else
```

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

```
3254   \eql@adjust@calc@tagmargin
3255   \ifdefined\eql@columns@fulllength
3256     \ifdim\eql@tagmargin@>\z@
3257       \advance\eql@tagmargin@\eql@tagsepmin@
3258     \fi
3259   \fi
3260   \ifdim\eql@tagmargin@>\dimexpr\displaywidth-\eql@totalwidth@
3261     -\eql@intercolumns@\eql@colsepmin@\relax
3262     \eql@tagmargin@\z@
3263   \fi
3264   \eql@marginleft@min@\z@
3265 \fi
```

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

```
3266 \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:

```

3267 \eql@colsep@\dimexpr\displaywidth-\eql@totalwidth@\relax
3268 \ifdefined\eql@layoutleft
3269 \advance\eql@colsep@-\eql@marginleft@
3270 \else
3271 \advance\eql@colsep@-\eql@tagmargin@
3272 \fi
3273 \count@\eql@intercolumns@
3274 \ifdefined\eql@columns@fulllength\else
3275 \ifdefined\eql@layoutleft
3276 \advance\count@\@ne
3277 \else
3278 \advance\count@\tw@
3279 \fi
3280 \fi
3281 \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:

```

3282 \ifdim\eql@colsep@<\eql@colsepmin@
3283 \eql@colsep@\eql@colsepmin@
3284 \else
3285 \ifdefined\eql@columns@fulllength\else
3286 \dimen@\glueexpr\eql@colsepmax@val\relax
3287 \ifdim\eql@colsep@>\dimen@
3288 \eql@colsep@\dimen@
3289 \fi
3290 \fi
3291 \fi
3292 \else

```

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

```

3293 \eql@colsep@\eql@colsepmin@
3294 \fi

```

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

```

3295 \ifdefined\eql@layoutleft

```

Set the default value:

```

3296 \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:

```

3297 \dimen@\dimexpr\displaywidth-\eql@totalwidth@
3298 -\eql@intercolumns@\eql@colsep@\relax
3299 \ifdim\dimen@<\eql@marginleft@
3300 \ifdim\dimen@<\eql@marginleft@min@
3301 \eql@marginleft@\eql@marginleft@min@
3302 \else
3303 \eql@marginleft@\dimen@
3304 \fi
3305 \fi
3306 \fi
3307 \else

```

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

```

3308 \ifdefined\eql@columns@fulllength
3309 \eql@marginleft@z@

```

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

```

3310 \else
3311 \eql@marginleft@dimexpr(\displaywidth-\eql@totalwidth@
3312 -\eql@intercolumns@\eql@colsep@-\eql@tagmargin@)/\tw@relax
3313 \ifdim\eql@marginleft@<z@
3314 \eql@marginleft@z@
3315 \fi
3316 \fi

```

Add tag margin in case of left tags:

```

3317 \ifdefined\eql@tagsleft
3318 \advance\eql@marginleft@\eql@tagmargin@
3319 \fi
3320 \fi

```

Find the best row for tag placement:

```

3321 \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:

```

3322 \ifdefined\eql@tagsleft
3323 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsleft
3324 \else
3325 \let\eql@adjust@columns@test\eql@adjust@columns@test@tagsright
3326 \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

```

3327 \ifdefined\eql@numbering@multi
3328 \eql@dimensions@for{%
3329 \ifdim\eql@tagwidth@>z@
3330 \eql@dimensions@calc
3331 \eql@adjust@columns@test
3332 \fi
3333 }%
3334 \else
3335 \ifnum\eql@tagpos@row@>z@
3336 \ifnum\eql@tagpos@row@>\eql@totalrows@\else
3337 \eql@dimensions@get\eql@tagpos@row@
3338 \eql@tagwidth@\eql@tagwidth@block@
3339 \eql@dimensions@calc
3340 \eql@adjust@columns@test
3341 \fi
3342 \fi
3343 \ifnum\eql@tagpos@prevrow@>z@
3344 \eql@dimensions@get\eql@tagpos@prevrow@
3345 \eql@tagwidth@\eql@tagwidth@block@
3346 \eql@dimensions@calc
3347 \eql@adjust@columns@test
3348 \fi

```

```
3349 \fi
```

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

```
3350 \advance\eq@totalwidth@\dimexpr
3351 \eq@intercolumns@\eq@colsep@+\eq@marginleft@\relax
3352 }
```

Placement for Right Tags.

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

```
3353 \def\eq@adjust@columns@test@tagsright{%
```

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

```
3354 \@tempdima\dimexpr\displaywidth-\eq@linewidth-\eq@tagwidth@\relax
```

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

```
3355 \ifdim\@tempdima<\dimexpr
3356 \eq@marginleft@+\eq@linewidthsep@\eq@colsep@\relax
```

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

```
3357 \ifdim\@tempdima<\dimexpr
3358 \eq@marginleft@min@+\eq@linewidthsep@\eq@colsepmin@\relax\else
```

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

```
3359 \eq@adjust@columns@modify@tagsright
3360 \fi
3361 \fi
3362 }
```

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

```
3363 \def\eq@adjust@columns@modify@tagsright{%
```

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

```
3364 \ifnum\eq@linewidthsep@>\z@
3365 \dimen@\@tempdima
3366 \count@\eq@linewidthsep@
3367 \ifdefined\eq@layoutleft
3368 \advance\dimen@-\eq@marginleft@
3369 \else
3370 \ifdefined\eq@columns@fulllength\else
3371 \advance\count@\@ne
3372 \fi
3373 \fi
3374 \divide\dimen@\count@
```

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

```

3375 \ifdim\dimen@<\eq\colsep@
3376 \ifdim\dimen@<\eq\colsepmin@
3377 \eq\colsep@\eq\colsepmin@
3378 \else
3379 \eq\colsep@\dimen@
3380 \fi
3381 \fi
3382 \fi

```

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

```

3383 \dimen@\dimexpr\@tempdima-\eq\line@widthsep@\eq\colsep@\relax
3384 \ifdim\eq\marginleft@>\dimen@
3385 \eq\marginleft@\dimen@
3386 \fi
3387 }

```

Placement for Left Tags.

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

```

3388 \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.

```

3389 \count@\numexpr\eq\intercolumns@-\eq\line@availsep@\relax
3390 \@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.

```

3391 \ifdim\@tempdima>\dimexpr
3392 \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).

```

3393 \ifdim\eq>tagwidth@<%
3394 \ifdefined\eq\layoutleft
3395 \glueexpr\eq\layoutleftmarginmax\relax
3396 \else
3397 \displaywidth
3398 \fi

```

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

```

3399 \ifdim\@tempdima>\dimexpr
3400 \displaywidth-\eq\totalwidth@-\count@\eq\colsepmin@\relax\else

```

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

```

3401 \eq\adjust@columns@modify@tagsleft
3402 \fi
3403 \fi
3404 \fi
3405 }

```

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

```
3406 \def\eql@adjust@columns@modify@tagsleft{%
```

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

```
3407 \ifnum\count@>\z@
3408 \dimen@ \dimexpr\displaywidth-\eql@totalwidth@-\@tempdima\relax
3409 \ifdefined\eql@columns@fulllength\else
3410 \advance\count@\@ne
3411 \fi
3412 \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.

```
3413 \ifdim\dimen@<\eql@colsep@
3414 \ifdim\dimen@<\eql@colsepmin@
3415 \dimen@\eql@colsepmin@
3416 \fi
3417 \advance\dimen@-\eql@colsep@
3418 \advance\eql@marginleft@-\eql@intercolumns@\dimen@
3419 \advance\eql@colsep@\dimen@
3420 \fi
3421 \fi
```

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

```
3422 \dimen@\dimexpr\@tempdima-\eql@line@availsep@\eql@colsep@\relax
3423 \ifdim\eql@marginleft@<\dimen@
3424 \eql@marginleft@\dimen@
3425 \fi
3426 }
```

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:

```
3427 \ifdefined\inf@bad\else%
3428 \newcount\inf@bad
3429 \inf@bad1000000\relax
3430 \fi
```

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

```
3431 \let\eql@restore@hfuzz\@empty
3432 \def\eql@save@hfuzz{\edef\eql@restore@hfuzz{\hfuzz\the\hfuzz\relax}}
```


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

```
3433 \newcount\eql@alignbadness@
3434 \newcount\eql@tagbadness@
3435 \newcount\eql@arrange@badness@
3436 \eql@alignbadness@ \inf@bad
3437 \eql@tagbadness@ \inf@bad
```

10.2 Arrangement Methods

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

```
3438 \def\eql@arrange@try#1{%
3439   \ifdim#1>\dimexpr\displaywidth-\eql@cellwidth@\relax
3440     \setbox\eql@cellbox@\hbox to\displaywidth{%
3441       \unhbox\eql@cellbox@\unkern\kern#1}%
3442     \eql@arrange@badness@\badness
3443   \else
3444     \eql@arrange@badness@\m@ne
3445   \fi
3446 }
```

`\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

```
3447 \def\eql@arrange@print#1#2{%
3448   \eql@restore@hfuzz
3449   \if@eqnsw
3450     \ifdefined\eql@tagsleft
3451       \eql@tagbox@print@tagsleft
3452     \fi
3453   \fi
3454   \hbox to\displaywidth{%
3455     #1%
3456     \unhbox\eql@cellbox@\unkern
3457     #2%
3458     \eql@tagging@mathaddlast
3459   }%
3460   \if@eqnsw
3461     \ifdefined\eql@tagsleft\else
3462       \eql@tagbox@print@tagsright
3463     \fi
3464   \fi
3465 }
```

`\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:

```
3466 \def\eql@arrange@print@alignleft#1#2{%
3467   \eql@display@firstavail@set{\dimexpr#1\relax}%
3468   \eql@arrange@print{\kern#1}{\kern#2}%
3469 }
```

```

3470 \def\eql@arrange@print@alignright#1#2{%
3471   \eql@display@firstavail@set{\dimexpr\displaywidth-\eql@cellwidth@-#2\relax}%
3472   \eql@arrange@print{\kern#1\hfil}{\unskip\kern#2}%
3473 }

3474 \def\eql@arrange@print@aligncenter#1{%
3475   \eql@display@firstavail@set{\dimexpr
3476     (\displaywidth-\eql@cellwidth@+#1)/\tw@\relax}%
3477   \ifdim#1>\z@
3478     \eql@arrange@print{\kern#1\hfil}{}%
3479   \else
3480     \eql@arrange@print{\hfil}{\kern-#1}%
3481   \fi
3482 }

```

`\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:

```

3483 \def\eql@arrange@init{%
3484   \eql@save@hfuzz
3485   \hfuzz\maxdimen
3486   \eql@shape@select
3487 }

```

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

```

3488 \def\eql@arrange@print@line{%
3489   \eql@tagging@tagaddbox
3490   \csname eql@arrange%
3491     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3492     @init\endcsname
3493   \csname eql@arrange%
3494     @\ifcase\eql@shape@pos@ alignleft\or aligncenter\or alignright\fi
3495     @\ifdefined\eql@tagpos@reserve
3496       \ifdefined\eql@tagsleft tagsleft\else tagsright\fi\else
3497       notag\fi\endcsname
3498 }

```

10.3 Central Alignment

TODO: describe

```

3499 \def\eql@arrange@aligncenter@init{%
3500   \eql@tagging@aligncenter
3501   \eql@line@offset@\dimexpr\tw@\eql@shape@amount@
3502     +\eql@marginleft@-\eql@marginright@+\eql@centeroffset@\relax
3503 }

```

TODO: describe

```

3504 \def\eql@arrange@aligncenter@notag{%
3505   \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3506     \dimexpr\eql@line@offset@<\eql@marginleft@min@
3507       \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3508   \else
3509     \eql@line@offset@
3510   \fi

```

```

3511 \eql@arrange@print@aligncenter\eql@line@offset@
3512 \else
3513 \ifdim\eql@line@offset@<\eql@marginleft@min@
3514 \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3515 \else
3516 \eql@arrange@print@alignright\eql@marginleft@min@\z@
3517 \fi
3518 \fi
3519 }

```

TODO: describe

```

3520 \def\eql@arrange@aligncenter@tagsright{%
3521 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3522 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3523 \dimexpr\tw@\eql@marginleft@min@-\eql@line@offset@\relax
3524 \else
3525 \dimexpr\tw@\eql@tagwidth@+\eql@line@offset@\relax
3526 \fi
3527 \eql@arrange@print@aligncenter\eql@line@offset@
3528 \else
3529 \eql@arrange@try{\dimexpr\eql@tagwidth@+\eql@marginleft@min@\relax}%
3530 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3531 \ifdim\eql@line@offset@<\dimexpr\eql@marginleft@min@-\eql@tagwidth@\relax
3532 \eql@arrange@print@alignleft\eql@marginleft@min@\eql@tagwidth@
3533 \else
3534 \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3535 \fi
3536 \else
3537 \let\eql@tagpos@reserve\eql@false
3538 \eql@arrange@aligncenter@notag
3539 \fi
3540 \fi
3541 }

```

```

3542 \def\eql@arrange@aligncenter@tagsleft{%
3543 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3544 \ifdim\dimexpr\displaywidth-\eql@cellwidth@\relax>%
3545 \ifdim\eql@line@offset@<\eql@tagwidth@
3546 \dimexpr\tw@\eql@tagwidth@-\eql@line@offset@\relax
3547 \else
3548 \eql@line@offset@
3549 \fi
3550 \eql@arrange@print@aligncenter\eql@line@offset@
3551 \else
3552 \eql@arrange@try\eql@tagwidth@
3553 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3554 \ifdim\eql@line@offset@<\eql@tagwidth@
3555 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3556 \else
3557 \eql@arrange@print@alignright\eql@tagwidth@\z@
3558 \fi
3559 \else
3560 \let\eql@tagpos@reserve\eql@false
3561 \eql@arrange@aligncenter@notag
3562 \fi
3563 \fi
3564 \else
3565 \eql@arrange@aligncenter@notag

```

```

3566 \fi
3567 }

```

10.4 Left Alignment

```

3568 \def\eql@arrange@alignleft@init{%
3569   \eql@tagging@alignleft
3570   \eql@line@offset@dimexpr\eql@marginleft@+\eql@shape@amount@relax
3571   \ifdim\eql@line@offset@<\eql@marginleft@min@
3572     \eql@line@offset@\eql@marginleft@min@
3573   \fi
3574 }

3575 \def\eql@arrange@alignleft@notag{%
3576   \ifdim\eql@line@offset@>\eql@marginleft@min@
3577     \eql@arrange@try\eql@line@offset@
3578     \ifnum\eql@arrange@badness@<\eql@alignbadness@
3579       \eql@arrange@print@alignleft\eql@line@offset@\z@
3580     \else
3581       \eql@arrange@print@alignright\eql@marginleft@min@\z@
3582     \fi
3583   \else
3584     \eql@arrange@print@alignleft\eql@marginleft@min@\z@
3585   \fi
3586 }

3587 \def\eql@arrange@alignleft@tagsright{%
3588   \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@relax}%
3589   \ifnum\eql@arrange@badness@<\eql@alignbadness@
3590     \eql@arrange@print@alignleft\eql@line@offset@\eql@tagwidth@
3591   \else
3592     \ifdim\eql@line@offset@>\eql@marginleft@min@
3593       \eql@arrange@try{\dimexpr\eql@marginleft@min@+\eql@tagwidth@relax}%
3594     \fi
3595     \ifnum\eql@arrange@badness@<\eql@tagbadness@
3596       \eql@arrange@print@alignright\eql@marginleft@min@\eql@tagwidth@
3597     \else
3598       \let\eql@tagpos@reserve\eql@false
3599       \eql@arrange@alignleft@notag
3600     \fi
3601   \fi
3602 }

3603 \def\eql@arrange@alignleft@tagsleft{%
3604   \ifdim\eql@tagwidth@>\eql@marginleft@min@
3605     \ifdim\eql@line@offset@>\eql@tagwidth@
3606       \eql@arrange@try\eql@line@offset@
3607       \ifnum\eql@arrange@badness@<\eql@alignbadness@
3608         \eql@arrange@print@alignleft\eql@line@offset@\z@
3609       \else
3610         \eql@arrange@try\eql@tagwidth@
3611         \ifnum\eql@arrange@badness@<\eql@tagbadness@
3612           \eql@arrange@print@alignright\eql@tagwidth@\z@
3613         \else
3614           \let\eql@tagpos@reserve\eql@false
3615           \eql@arrange@print@alignright\eql@marginleft@min@\z@
3616         \fi
3617       \fi
3618   \else
3619     \eql@arrange@try\eql@tagwidth@

```

```

3620     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3621     \eqL@arrange@print@alignleft\eqL@tagwidth@z@
3622     \else
3623     \let\eqL@tagpos@reserve\eqL@false
3624     \eqL@arrange@alignleft@notag
3625     \fi
3626   \fi
3627 \else
3628   \eqL@arrange@alignleft@notag
3629 \fi
3630 }

```

10.5 Right Alignment

```

3631 \def\eqL@arrange@alignright@init{%
3632   \eqL@tagging@alignright
3633   \eqL@line@offset@dimexpr\eqL@marginright@-\eqL@shape@amount@relax
3634   \ifdim\eqL@line@offset@<z@
3635     \eqL@line@offset@z@
3636   \fi
3637 }

```

TODO: describe

```

3638 \def\eqL@arrange@alignright@notag{%
3639   \ifdim\eqL@line@offset@>z@
3640     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3641     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3642       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3643     \else
3644       \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3645     \fi
3646   \else
3647     \eqL@arrange@print@alignright\eqL@marginleft@min@z@
3648   \fi
3649 }

```

TODO: describe

```

3650 \def\eqL@arrange@alignright@tagsright{%
3651   \ifdim\eqL@line@offset@>\eqL@tagwidth@
3652     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@line@offset@relax}%
3653     \ifnum\eqL@arrange@badness@<\eqL@alignbadness@
3654       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@line@offset@
3655     \else
3656       \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3657       \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3658         \eqL@arrange@print@alignleft\eqL@marginleft@min@\eqL@tagwidth@
3659       \else
3660         \let\eqL@tagpos@reserve\eqL@false
3661         \eqL@arrange@print@alignleft\eqL@marginleft@min@z@
3662       \fi
3663     \fi
3664   \else
3665     \eqL@arrange@try{\dimexpr\eqL@marginleft@min@+\eqL@tagwidth@relax}%
3666     \ifnum\eqL@arrange@badness@<\eqL@tagbadness@
3667       \eqL@arrange@print@alignright\eqL@marginleft@min@\eqL@tagwidth@
3668     \else
3669       \let\eqL@tagpos@reserve\eqL@false
3670       \eqL@arrange@alignright@notag
3671     \fi

```

```

3672 \fi
3673 }

```

TODO: describe

```

3674 \def\eql@arrange@alignright@tagsleft{%
3675 \ifdim\eql@tagwidth@>\eql@marginleft@min@
3676 \eql@arrange@try{\dimexpr\eql@line@offset@+\eql@tagwidth@\relax}%
3677 \ifnum\eql@arrange@badness@<\eql@alignbadness@
3678 \eql@arrange@print@alignright\eql@tagwidth@\eql@line@offset@
3679 \else
3680 \ifdim\eql@line@offset@>\z@
3681 \eql@arrange@try\eql@tagwidth@
3682 \fi
3683 \ifnum\eql@arrange@badness@<\eql@tagbadness@
3684 \eql@arrange@print@alignleft\eql@tagwidth@\z@
3685 \else
3686 \let\eql@tagpos@reserve\eql@false
3687 \eql@arrange@alignright@notag
3688 \fi
3689 \fi
3690 \else
3691 \eql@arrange@alignright@notag
3692 \fi
3693 }

```

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`

```

3694 \def\eql@box@cr{%
3695 \ifmmode\else\unskip\fi
3696 \eql@vspaceskip@\z@skip
3697 \let\eql@punct@term\eql@false
3698 \let\eql@class@rel@composed@empty
3699 \eql@ampprotect\eql@box@cr@test\eql@box@cr@process
3700 }

```

TODO: describe

```

3701 \def\eql@box@cr@test@setopt{\let\eql@box@cr@test\eql@box@cr@testopt}
3702 \def\eql@box@cr@test@setall{\let\eql@box@cr@test\eql@box@cr@testall}

```

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

```

3703 \def\eql@box@cr@testopt#1{\eql@teststaropt@tight
3704 {\eql@box@cr@testopt@set{#1}}{\eql@box@cr@testopt@set{#1}}{0pt}}
3705 \def\eql@box@cr@testopt@set#1[#2]{\advance\eql@vspaceskip@glueexpr#2\relax#1}

```

```

\eql@box@cr@testall TODO: describe
box@cr@testall@parse
3706 \def\eql@box@cr@testall{\eql@parseopt@cr\eql@box@cr@testall@parse}
3707 \def\eql@box@cr@testall@parse{%
3708   \ifx\eql@parseopt@token[%
3709     \let\eql@parseopt@next\eql@parseopt@vspace
3710   \fi
3711   \ifx\eql@parseopt@token*%
3712     \let\eql@parseopt@next\eql@parseopt@gobble
3713   \fi
3714   \ifx\eql@parseopt@token.%
3715     \let\eql@parseopt@next\eql@parseopt@punctpass
3716   \fi
3717   \ifx\eql@parseopt@token,%
3718     \let\eql@parseopt@next\eql@parseopt@punctpass
3719   \fi
3720   \ifx\eql@parseopt@token~%
3721     \let\eql@parseopt@next\eql@parseopt@punctpass
3722   \fi
3723   \ifx\eql@parseopt@token'%
3724     \let\eql@parseopt@next\eql@parseopt@punctnext
3725   \fi
3726   \ifx\eql@parseopt@token!%
3727     \let\eql@parseopt@next\eql@parseopt@punctterm
3728   \fi
3729   \ifx\eql@parseopt@token/%
3730     \let\eql@parseopt@next\eql@parseopt@punctclear
3731   \fi
3732   \ifx\eql@parseopt@token=%
3733     \let\eql@parseopt@next\eql@parseopt@relsyb
3734   \fi
3735   \ifx\eql@parseopt@token;%
3736     \let\eql@parseopt@next\eql@parseopt@relcont
3737   \fi
3738   \ifx\eql@parseopt@token:%
3739     \let\eql@parseopt@next\eql@parseopt@relstart
3740   \fi
3741   \ifx\eql@parseopt@token|%
3742     \let\eql@parseopt@next\eql@parseopt@relord
3743   \fi
3744   \ifx\eql@parseopt@token?%
3745     \let\eql@parseopt@next\eql@box@cr@parse@rel
3746   \fi
3747   \ifx\eql@parseopt@token&%
3748     \let\eql@parseopt@next\eql@parseopt@end
3749   \fi
3750 }
3751 \def\eql@box@cr@parse@rel#1#2{%
3752   \def\eql@tmp{#2}%
3753   \ifx\eql@tmp\eql@relax\else
3754     \eql@punct@next@clear
3755   \fi
3756   \ifx\eql@box@open\eql@box@lines@open
3757     \ifx\eql@tmp\@empty
3758       \def\eql@class@rel@composed{\eql@shape@cont}%
3759     \else
3760       \def\eql@class@rel@composed{\eql@shape@rel#2}%
3761     \fi
3762   \fi

```

```

3763 \ifx\eql@box@open\eql@box@columns@open
3764 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3765 \fi
3766 \ifx\eql@box@open\eql@box@cases@open
3767 \def\eql@class@rel@composed{&#2}%
3768 \fi
3769 \eql@parseopt@end}

```

\eql@box@cr@process

```

3770 \def\eql@box@cr@process{%
3771 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3772 \edef\eql@tmp{%
3773 \unexpanded{%
3774 \eql@box@endline
3775 \eql@box@lastcell
3776 \cr
3777 }%
3778 \noalign{%
3779 \vskip\the\eql@vspaceskip@relax
3780 }%
3781 \unexpanded\expandafter{\eql@class@rel@composed}%
3782 }%
3783 \eql@tmp
3784 }

```

\eql@box@endline

```

3785 \def\eql@box@endline{%
3786 \eql@punct@apply@line
3787 \eql@hook@lineout
3788 }

```

11.2 Column Breaks

TODO: describe

```

3789 \def\eql@box@amp{%
3790 \eql@ampprotecttwo\eql@box@amp@testescape\eql@amp@org
3791 \eql@box@amp@process}
3792 \def\eql@box@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
3793 \relax
3794 \let\eql@punct@term\eql@false
3795 \let\eql@class@rel@composed\@empty
3796 \eql@box@amp@test{#2}}}%

```

TODO: describe

```

3797 \def\eql@box@amp@test@setopt{%
3798 \let\eql@box@amp@test\eql@box@amp@testopt}
3799 \def\eql@box@amp@test@setall{%
3800 \let\eql@box@amp@test\eql@box@amp@testall}

```

\eql@box@amp@testopt **TODO:** describe

```

3801 \let\eql@box@amp@testopt\@empty

```

\eql@box@amp@testall **TODO:** describe

ox@amp@testall@parse


```

3802 \def\eql@box@amp@testall{\eql@parseopt@cr\eql@box@amp@testall@parse}
3803 \def\eql@box@amp@testall@parse{%
3804   \ifx\eql@parseopt@token.%
3805     \let\eql@parseopt@next\eql@parseopt@punctpass
3806   \fi
3807   \ifx\eql@parseopt@token,%
3808     \let\eql@parseopt@next\eql@parseopt@punctpass
3809   \fi
3810   \ifx\eql@parseopt@token~%
3811     \let\eql@parseopt@next\eql@parseopt@punctpass
3812   \fi
3813   \ifx\eql@parseopt@token'%
3814     \let\eql@parseopt@next\eql@parseopt@punctnext
3815   \fi
3816   \ifx\eql@parseopt@token!%
3817     \let\eql@parseopt@next\eql@parseopt@punctterm
3818   \fi
3819   \ifx\eql@parseopt@token=%
3820     \let\eql@parseopt@next\eql@parseopt@relsymp
3821   \fi
3822   \ifx\eql@parseopt@token;%
3823     \let\eql@parseopt@next\eql@parseopt@relcont
3824   \fi
3825   \ifx\eql@parseopt@token:%
3826     \let\eql@parseopt@next\eql@parseopt@relstart
3827   \fi
3828   \ifx\eql@parseopt@token|%
3829     \let\eql@parseopt@next\eql@parseopt@relord
3830   \fi
3831   \ifx\eql@parseopt@token?%
3832     \let\eql@parseopt@next\eql@box@amp@parse@rel
3833   \fi
3834   \ifx\eql@parseopt@token&%
3835     \let\eql@parseopt@next\eql@parseopt@end
3836   \fi
3837 }
3838 \def\eql@box@amp@parse@rel#1#2{%
3839   \ifx\eql@box@open\eql@box@columns@open
3840     \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
3841   \else
3842     \def\eql@class@rel@composed{#2}%
3843   \fi
3844   \eql@parseopt@end}

```

\eql@box@amp@process

```

3845 \def\eql@box@amp@process{%
3846   \ifdefined\eql@punct@term\eql@punct@apply@top\fi
3847   \ifx\eql@box@open\eql@box@columns@open
3848     \edef\eql@tmp{%
3849       \ifx\eql@class@rel@composed\@empty
3850         \ifx\eql@box@lastcell\eql@box@columns@lastcell@odd
3851           &\noexpand\eql@punct@next@clear\fi&%
3852       \else
3853         \ifx\eql@box@lastcell\eql@box@columns@lastcell@even&\fi%
3854         \unexpanded\expandafter{\eql@class@rel@composed}%
3855       \fi
3856     }%
3857   \else

```

```

3858 \edef\eql@tmp{%
3859 \ifx\eql@class@rel@composed\@empty
3860 &%
3861 \else
3862 \unexpanded\expandafter{\eql@class@rel@composed}%
3863 \fi
3864 }%
3865 \fi
3866 \eql@tmp
3867 }

```

11.3 Lines Mode

```

3868 \def\eql@box@lines@lastcell{&\omit\kern-2\eql@colsep@}

```

TODO: templates

```

3869 \def\eql@box@lines@open{%
3870 \eql@shape@align@enable
3871 \let\eql@box@lastcell\eql@box@lines@lastcell
3872 \everycr{\noalign{%
3873 \eql@verbose@info\eql@verbose@msg@startline@number
3874 \global\advance\eql@row@\@ne
3875 }}%
3876 \tabskip\z@skip
3877 \halign\bgroup
3878 &%
3879 \global\let\eql@cell@container\@empty
3880 \setbox\eql@cellbox@\hbox{%
3881 \eql@strut@cell
3882 \@lign
3883 $\m@th\eql@mathstyle
3884 \eql@hook@colin
3885 ##%
3886 \eql@punct@apply@col
3887 \eql@hook@colout
3888 \eql@tagging@mathsave
3889 $%
3890 \eql@tagging@mathaddlast
3891 }%
3892 \ifdefined\eql@shape@lastrow
3893 \eql@totalrows@\eql@row@
3894 \fi
3895 \eql@shape@eval
3896 \eql@cell@container
3897 \ifdefined\eql@frame@cmd
3898 \ifcase\eql@shape@pos@
3899 \eql@frame@measure
3900 \advance\eql@shape@amount@-\eql@frame@margin@
3901 \or\or
3902 \eql@frame@measure
3903 \advance\eql@shape@amount@+\eql@frame@margin@
3904 \fi
3905 \eql@frame@print
3906 \fi
3907 \ifcase\eql@shape@pos@
3908 \kern\eql@shape@amount@
3909 \box\eql@cellbox@
3910 \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@

```

```

3911         -\eql@shape@amount@+\@flushglue\relax
3912         \eql@tagging@alignleft
3913     \or
3914         \hskip\glueexpr\eql@paddingleft@+\eql@shape@amount@+\@flushglue\relax
3915         \box\eql@cellbox@
3916         \hskip\glueexpr\eql@paddingright@-\eql@shape@amount@+\@flushglue\relax
3917         \eql@tagging@aligncenter
3918     \or
3919         \hskip\glueexpr\eql@paddingleft@+\eql@paddingright@
3920         +\eql@shape@amount@+\@flushglue\relax
3921         \box\eql@cellbox@
3922         \kern-\eql@shape@amount@
3923         \eql@tagging@alignright
3924     \fi
3925     \tabskip\eql@colsep@\relax
3926 \crr
3927 \noalign{%
3928     \global\let\eql@shape@lastrow\eql@false
3929     \eql@hook@blockbefore
3930 }%
3931 \eql@hook@blockin
3932 }
3933 \def\eql@box@lines@set{\let\eql@box@open\eql@box@lines@open}

```

11.4 Columns Mode

```

3934 \def\eql@box@columns@lastcell@odd{%
3935     &\omit
3936     \eql@prevwidth@\wd\eql@cellbox@
3937     \let\eql@frame@cmd\eql@frame@prevcmd
3938     \ifdefined\eql@frame@cmd
3939         \eql@frame@measure
3940         \advance\eql@prevwidth@\eql@frame@margin@
3941         \eql@frame@print
3942     \fi
3943     \kern-\eql@prevwidth@
3944     \unhbox\eql@cellbox@
3945     \hfil
3946     &\omit\kern-\eql@colsep@
3947 }%
3948 \def\eql@box@columns@lastcell@even{&\omit\kern-\eql@colsep@}
3949 \def\eql@box@columns@open{%
3950 % \TODO templates
3951     \eql@shape@align@disable
3952     \let\eql@box@lastcell\@empty
3953     \everycr{\noalign{%
3954         \eql@verbose@info\eql@verbose@msg@startline@new
3955     }}%
3956     \tabskip\z@skip
3957     \halign\bgroup
3958     &%
3959     \let\eql@box@lastcell\eql@box@columns@lastcell@odd
3960     \global\let\eql@cell@container\@empty
3961     \global\setbox\eql@cellbox@\hbox{%
3962         \eql@strut@cell
3963         \@lign
3964         $\m@th\eql@mathstyle
3965         \eql@hook@colin

```

```

3966      ##%
3967      \eql@punct@apply@next
3968      \eql@class@innerleft
3969      \eql@hook@innerleft
3970      \eql@tagging@mathsave
3971      $%
3972      \eql@tagging@mathaddlast
3973  }%
3974  \eql@cell@container
3975  \hfil
3976  \kern\wd\eql@cellbox@
3977  \ifdefined\eql@frame@cmd
3978      \eql@frame@measure
3979      \kern\eql@frame@margin@
3980  \fi
3981  \global\let\eql@frame@prevcmd\eql@frame@cmd
3982  \tabskip\z@skip
3983  &%
3984  \eql@prevwidth@\wd\eql@cellbox@
3985  \let\eql@box@lastcell\eql@box@columns@lastcell@even
3986  \let\eql@frame@cmd\eql@frame@prevcmd
3987  \global\let\eql@cell@container\@empty
3988  \setbox\eql@cellbox@\hbox{%
3989      \unhbox\eql@cellbox@
3990      \eql@strut@cell
3991      \@lign
3992      $\m@th\eql@mathstyle
3993      \eql@hook@innerright
3994      \eql@class@innerright@sel
3995      ##%
3996      \eql@punct@apply@col
3997      \eql@hook@colout
3998      \eql@tagging@mathsave
3999      $%
4000      \eql@tagging@mathaddlast
4001  }%
4002  \eql@cell@container
4003  \ifdefined\eql@frame@cmd
4004      \eql@frame@measure
4005      \advance\eql@prevwidth@\eql@frame@margin@
4006      \eql@frame@print
4007  \fi
4008  \kern-\eql@prevwidth@
4009  \unhbox\eql@cellbox@
4010  \hfil
4011  \tabskip\eql@colsep@\relax
4012  \crrc
4013  \noalign{%
4014      \eql@hook@blockbefore
4015  }%
4016  \eql@hook@blockin
4017  }
4018  \def\eql@box@columns@set{\let\eql@box@open\eql@box@columns@open}

```

11.5 Cases Mode

TODO: describe

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

alignment?

```
4019 \def\eq@box@cases@lastcell{&}%

4020 \let\eq@box@cases@condtext\eq@false
4021 \let\eq@box@cases@condintro\@empty

4022 \def\eq@box@cases@open{%
4023   \eq@shape@align@disable
4024   \let\eq@box@lastcell\@empty
4025   \everycr{\noalign{%
4026     \eq@verbose@info\eq@verbose@msg@startline@new
4027   }}%
4028   \tabskip\z@skip
4029   \halign\bgroup
4030     \let\eq@box@lastcell\eq@box@cases@lastcell
4031     \global\let\eq@cell@container\@empty
4032     \global\setbox\eq@cellbox@\hbox{%
4033       \eq@strut@cell
4034       \@lign
4035       $\m@th\eq@mathstyle
4036       \eq@hook@colin
4037       ##%
4038       \eq@punct@apply@next
4039       \eq@tagging@mathsave
4040       $%
4041       \eq@tagging@mathaddlast
4042     }%
4043     \eq@cell@container
4044     \unhbox\eq@cellbox@
4045     \hfil
4046     \eq@tagging@alignleft
4047     \tabskip\eq@colsep@\relax
4048   &%
4049   \let\eq@box@lastcell\@empty
4050   \global\let\eq@cell@container\@empty
4051   \setbox\eq@cellbox@\hbox{%
4052     \unhbox\eq@cellbox@
4053     \eq@strut@cell
4054     \@lign
4055     $\m@th\eq@mathstyle
4056     \ifdefined\eq@box@cases@condtext
4057       \expandafter\hbox\else\expandafter\@firstofone\fi\bgroup
4058     \eq@box@cases@condintro
4059     ##%
4060     \eq@punct@apply@col
4061     \egroup
4062     \eq@hook@colout
4063     \eq@tagging@mathsave
4064     $%
4065     \eq@tagging@mathaddlast
4066   }%
4067   \eq@cell@container
4068   \unhbox\eq@cellbox@
4069   \hfil
4070   \eq@tagging@alignleft
4071   \tabskip\z@skip
4072   \crr
4073   \noalign{%
```

```

4074     \eql@hook@blockbefore
4075 }%
4076     \eql@hook@blockin
4077 }

4078 \def\eql@box@cases@set{%
4079     \ifdefined\eql@cases@mathstyle\let\eql@mathstyle\eql@cases@mathstyle\fi
4080     \let\eql@box@open\eql@box@cases@open}

```

11.6 Main

```

4081 \let\eql@box@box\vcenter
4082 \let\eql@box@open\@undefined
4083 \let\eql@box@frame\@firstofone
4084 \def\eql@box@wrap#1#2{\def\eql@box@frame##1{#1##1#2}}

4085 \def\eql@box@delim#1#2{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4086     \left#1##1\right#2}}
4087 \def\eql@box@getdim{\setbox\@ne\hbox{\@ne\ht\@ne\ht\z@dp\@ne\dp\z@}
4088 \def\eql@box@deldim#1{\hbox{$\m@th\hbox{\null\delimiterspace\z@\left#1
4089     \ifx\eql@box@box\vcenter\vcenter{\box\@ne}\else\box\@ne\fi\right.$}}
4090 \def\eql@box@ldelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4091     \eql@box@getdim\eql@box@deldim#1##1}}
4092 \def\eql@box@rdelim#1{\def\eql@box@frame##1{\mathopen{}}\mathclose{%
4093     \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!

```

4094 \def\eql@box@close{%
4095     \ifvmode\else
4096         \ifmmode\else\unskip\fi
4097         \global\let\eql@shape@lastrow\eql@true
4098         \eql@punct@apply@block
4099         \ifdefined\eql@box@punct@term
4100             \eql@punct@apply@top
4101         \fi
4102         \eql@box@endline
4103         \eql@box@lastcell
4104         \cr
4105     \fi
4106     \noalign{%
4107         \eql@hook@blockafter
4108         \global\let\eql@shape@lastrow\eql@false
4109     }%
4110     \eql@tagging@tablesaveinner
4111 \egroup
4112 }

```

`\eql@box@vcenter`

```

4113 \def\eql@box@vcenter#1{%
4114     \ifmmode
4115         \vcenter{#1}%
4116     \else
4117         $\m@th\vcenter{#1}$%
4118     \fi

```

4119 }

\eql@box@start

```

4120 \let\eql@box@endmath\eql@false
4121 \def\eql@box@start{%
4122   \relax
4123   \ifmmode
4124     \let\eql@box@endmath\eql@false
4125   \else
4126     \let\eql@box@endmath\eql@true
4127     \expandafter$%$
4128   \fi
4129   \eql@stack@save@box
4130   \let\eql@frame@cmd\@undefined
4131   \let\eql@layoutleft\eql@false
4132   \eql@row@z@
4133   \eql@totalrows@\@M
4134   \eql@shape@select
4135   \setbox\z@\ifx\eql@box@box\vcenter
4136     \expandafter\ vbox
4137   \else
4138     \expandafter\eql@box@box
4139   \fi\bgroup
4140   \let\eqnpunct\eql@punct@setnext
4141   \eql@display@nest
4142   \let\\\eql@box@cr
4143   \ifdefined\eql@amp@mode
4144     \let\&\eql@box@amp
4145   \fi
4146   \eql@spread@set
4147   \eql@strut@make
4148   \eql@box@open
4149 }
```

\eql@box@end

```

4150 \def\eql@box@end{%
4151   \eql@box@close
4152   \egroup
4153   \eql@box@frame{%
4154     \ifdefined\eql@display@marginleft
4155       \hskip\glueexpr\eql@display@marginleft\relax
4156     \fi
4157     \ifx\eql@box@box\vcenter
4158       \eql@box@vcenter{\unvbox\z@}%
4159     \else
4160       \box\z@
4161     \fi
4162     \eql@tagging@tableaddinner
4163     \ifdefined\eql@display@marginright
4164       \hskip\glueexpr\eql@display@marginright\relax
4165     \fi
4166   }%
4167   \eql@stack@restore
4168   \ifdefined\eql@box@endmath
4169     \expandafter$%$
4170   \fi
4171 }
```

`\eql@box@main` Combined opening, body and closing for pre-scanned body:

```
4172 \def\eql@box@main{%
4173   \eql@box@start
4174   \eql@scan@body
4175   \eql@box@end
4176 }
```

11.7 Options Processing

TODO: describe

```
4177 \def\eql@box@test@setopt{\let\eql@box@test\eql@box@testopt}
4178 \def\eql@box@test@setall{\let\eql@box@test\eql@box@testall}
```

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

```
4179 \def\eql@box@testopt#1{\eql@testopt@tight{\eql@box@testopt@set{#1}}{}}
4180 \def\eql@box@testopt@set#1[#2]{\eqnaddopt{#2}#1}

4181 \def\eql@box@testall{\eql@parseopt@env\eql@box@testall@parse}
4182 \def\eql@box@testall@parse{%
4183   \ifx\eql@parseopt@token[%
4184     \let\eql@parseopt@next\eql@parseopt@opt
4185   \fi
4186   \ifx\eql@parseopt@token.%
4187     \let\eql@parseopt@next\eql@parseopt@punctpass
4188   \fi
4189   \ifx\eql@parseopt@token,%
4190     \let\eql@parseopt@next\eql@parseopt@punctpass
4191   \fi
4192   \ifx\eql@parseopt@token~%
4193     \let\eql@parseopt@next\eql@parseopt@punctpass
4194   \fi
4195   \ifx\eql@parseopt@token' %
4196     \let\eql@parseopt@next\eql@parseopt@punctopt
4197   \fi
4198   \ifx\eql@parseopt@token!%
4199     \let\eql@parseopt@next\eql@box@parse@punctterm
4200   \fi
4201   \ifx\eql@parseopt@token=%
4202     \let\eql@parseopt@next\eql@parseopt@lines
4203   \fi
4204   \ifx\eql@parseopt@token| %
4205     \let\eql@parseopt@next\eql@parseopt@columns
4206   \fi
4207   \ifx\eql@parseopt@token<%
4208     \let\eql@parseopt@next\eql@parseopt@ampeq
4209   \fi
4210   \ifx\eql@parseopt@token>%
4211     \let\eql@parseopt@next\eql@parseopt@eqamp
4212   \fi
4213 }
4214 \def\eql@box@parse@punctterm#1{\eqnaddopt{punctterm}\eql@parseopt@peek}
```

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

```
4215 \def\eql@box@end@testall{\eql@parseopt@env\eql@box@end@testall@parse}
4216 \def\eql@box@end@testall@parse{%
```



```

4217 \ifx\eql@parseopt@token.%
4218   \let\eql@parseopt@next\eql@parseopt@punctpass
4219 \fi
4220 \ifx\eql@parseopt@token,%
4221   \let\eql@parseopt@next\eql@parseopt@punctpass
4222 \fi
4223 \ifx\eql@parseopt@token~%
4224   \let\eql@parseopt@next\eql@parseopt@punctpass
4225 \fi
4226 \ifx\eql@parseopt@token'%
4227   \let\eql@parseopt@next\eql@parseopt@punctblock
4228 \fi
4229 \ifx\eql@parseopt@token!%
4230   \let\eql@parseopt@next\eql@box@end@parse@punctterm
4231 \fi
4232 }
4233 \def\eql@box@end@parse@punctterm#1{%
4234   \let\eql@box@punct@term\eql@true\eql@parseopt@peek}

```

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

```

4235 \def\eql@box@processopt{%
4236   \let\eql@box@frame\@firstofone
4237   \let\eql@display@marginleft\@undefined
4238   \let\eql@display@marginright\@undefined
4239   \let\eql@box@punct@term\eql@false
4240   \let\eql@punct@block\@undefined
4241   \eql@nextopt@process{box}%
4242   \let\eql@punct@next\@undefined
4243   \eql@colsep@\glueexpr\eql@box@colsep\relax
4244   \ifdefined\eql@paddingleft@val
4245     \eql@paddingleft@\glueexpr\eql@paddingleft@val\relax
4246   \else
4247     \eql@paddingleft@\z@
4248   \fi
4249   \ifdefined\eql@paddingright@val
4250     \eql@paddingright@\glueexpr\eql@paddingright@val\relax
4251   \else
4252     \eql@paddingright@\z@
4253   \fi
4254   \eql@indent@\glueexpr\eql@indent@val\relax
4255 }

```

11.8 Environment

`equationsbox` (*env.*)

```

4256 \newenvironment{equationsbox}{%
4257   \eql@verbose@info\eql@verbose@msg@enterenv
4258   \ifdefined\eql@box@env@modifier
4259     \eql@box@test@setall
4260   \else
4261     \eql@box@nomodifier
4262   \fi
4263   \eql@ampprotect\eql@box@test\eql@box@env@start
4264 }{%
4265   \ifdefined\eql@box@doscan\else
4266     \expandafter\eql@box@end

```

```

4267 \fi
4268 \eql@verbose@info\eql@verbose@msg@leaveenv
4269 }

```

`\eql@box@env@start`

```

4270 \def\eql@box@env@start{%
4271   \eql@box@processopt
4272   \ifdefined\eql@box@doscan
4273     \eql@box@call@set
4274     \expandafter\eql@scan@env
4275   \else
4276     \expandafter\eql@box@start
4277   \fi
4278 }

```

`\eql@box@call`

```

4279 \def\eql@box@call{\eql@box@main\eql@scan@end}
4280 \def\eql@box@call@test{%
4281   \eql@ampprotect\eql@box@end@testall\eql@box@call}
4282 \def\eql@box@call@set{%
4283   \ifdefined\eql@box@end@modifier
4284     \let\eql@scan@call\eql@box@call@test
4285   \else
4286     \let\eql@scan@call\eql@box@call
4287   \fi
4288 }

```

`\eql@box@ang@open`

```

4289 \newenvironment{equationsbox@ang}{}{}
4290 \def\eql@box@ang@open{%
4291   \expandafter\eqnaddopt\expandafter{\eql@box@ang@opt}%
4292   \begin{equationsbox@ang}%
4293   \eql@verbose@info\eql@verbose@msg@enterenv
4294   \let\>\eql@box@ang@close
4295   \ifdefined\eql@box@ang@modifier
4296     \eql@box@test@setall
4297   \else
4298     \eql@box@nomodifier
4299   \fi
4300   \eql@ampprotect\eql@box@test\eql@box@ang@start
4301 }

```

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

```

4302 \def\eql@box@ang@start{%
4303   \eql@box@processopt
4304   \ifdefined\eql@box@doscan
4305     \eql@box@call@set
4306     \expandafter\eql@scan@ang
4307   \else
4308     \expandafter\eql@box@start
4309   \fi
4310 }

```

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

```

4311 \def\eql@box@ang@close{%
4312   \ifdefined\eql@box@doscan
4313     \let\eql@box@end@modifier\eql@false
4314   \fi
4315   \ifdefined\eql@box@end@modifier
4316     \expandafter\eql@ampprotect\expandafter\eql@box@end@testall
4317   \fi
4318   \eql@box@ang@end
4319 }

```

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

```

4320 \def\eql@box@ang@end{%
4321   \ifdefined\eql@box@doscan\else
4322     \expandafter\eql@box@end
4323   \fi
4324   \eql@verbose@info\eql@verbose@msg@leaveenv
4325   \end{equationsbox@ang}%
4326   \ignorespaces
4327 }

```

12 Single-Line Equation

TODO: describe

12.1 Native Mode

```

4328 \def\eql@single@start@native{%
4329   \eql@display@init
4330   \eql@display@print
4331   \let\raisetag\eql@raisetag@default
4332   \eql@shape@align@disable
4333   \eql@hook@eqin
4334   % \mathopen{}%
4335 }%

```

TODO: describe

```

4336 \def\eql@single@end@native{%
4337   % \mathclose{}%
4338   \eql@tags@container
4339   \eql@numbering@single@eval
4340   \if@eqnsw
4341     \ifdefined\eql@tagsleft
4342       \leqno
4343     \else
4344       \eqno
4345     \fi
4346     \eql@composetag@print
4347   \fi
4348   \eql@interline@container
4349   \advance\eql@belowspace@\eql@vspaceskip@
4350   \eql@display@container
4351   \eql@display@penalty
4352   \eql@display@vspace@native
4353 }%

```

12.2 Print

```

4354 \def\eql@single@start@print{%
4355   \eql@display@init
4356   \eql@display@print
4357   \eql@shape@align@enable
4358   \eql@totalrows@\@ne
4359   \eql@row@\@ne
4360   \eql@arrange@init
4361   \global\let\eql@cell@container\@empty
4362   \prevgraf\numexpr\prevgraf+\@ne\relax
4363   \setbox\eql@cellbox@\hbox\bgroup
4364     \eql@restore@hfuzz
4365     \eql@strut@cell
4366     $\m@th\eql@mathstyle%$
4367     \eql@hook@eqin
4368 }
4369 \def\eql@single@end@print{%
4370   \eql@tagging@mathsave
4371   $%$
4372   \hfil
4373   \kern\z@
4374   \egroup
4375   \prevgraf\numexpr\prevgraf-\@ne\relax
4376   \eql@shape@eval
4377   \eql@cell@container
4378   \ifdefined\eql@frame@cmd
4379     \eql@frame@adjust
4380   \fi
4381   \eql@cellwidth@\wd\eql@cellbox@
4382   \eql@line@height@\ht\eql@cellbox@
4383   \eql@line@depth@\dp\eql@cellbox@
4384   \eql@totalwidth@\eql@cellwidth@
4385   \eql@totalheight@\dimexpr\eql@line@height@+\eql@line@depth@\relax
4386   \eql@topheight@\eql@line@height@
4387   \eql@bottomdepth@\eql@line@depth@
4388   \eql@tags@container
4389   \eql@numbering@single@eval
4390   \if@eqnsw
4391     \eql@tagbox@make\eql@composetag@print
4392     \eql@tagrows@\@ne
4393     \ifdefined\eql@tagpos@reserve\else
4394       \eql@tagwidth@\z@
4395     \fi
4396     \eql@tagheight@block@\ht\eql@tagbox@
4397     \eql@tagdepth@block@\dp\eql@tagbox@
4398   \else
4399     \eql@numbering@warnunused
4400     \eql@tagwidth@\z@
4401     \eql@tagrows@\z@
4402   \fi
4403   \eql@tagwidth@max@\eql@tagwidth@
4404   \eql@tagpos@single@eval
4405   \eql@tagpos@print@line@eval
4406   \eql@intercolumns@\z@
4407   \eql@adjust@calc@lines

```

```

4408 \eqldisplay@halign@init{}%
4409 \halign{##\crr
4410   \noalign{\eqldisplay@halign@start}%
4411   \eql@arrange@print@line
4412   \cr
4413   \noalign{\eqldisplay@halign@end}%
4414   \eql@tagging@tablesavelines
4415 }%
4416 \eql@tagpos@print@line@end
4417 \eqldisplay@close
4418 }

```

13 Multi-Line with Single Column

TODO: outline sequence of calls

13.1 Measure

TODO: describe

```

4419 \def\eql@lines@measure@line@begin{%
4420   \eql@verbose@info\eql@verbose@msg@startline@number
4421   \eql@numbering@measure@line@begin
4422   \eql@hook@linein
4423 }

```

TODO: describe

```

4424 \def\eql@lines@measure@line@end{%
4425   \eql@punct@apply@line
4426   \eql@hook@lineout
4427 }

```

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

```

4428 \def\eql@lines@measure@cell{%
4429   \ifdefined\eql@frame@cmd
4430     \ifcase\eql@shape@pos@
4431       \eql@frame@measure
4432       \advance\eql@shape@amount@-\eql@frame@margin@
4433     \or\or
4434       \eql@frame@measure
4435       \advance\eql@shape@amount@+\eql@frame@margin@
4436     \fi
4437     \eql@frame@print
4438   \fi
4439   \eql@cellwidth@\wd\eql@cellbox@
4440   \eql@line@height@\ht\eql@cellbox@
4441   \eql@line@depth@\dp\eql@cellbox@
4442   \eql@dimensions@startrow
4443   \eql@dimensions@savercell
4444   \kern\eql@cellwidth@
4445 }

```

\eql@lines@measure

```

4446 \def\eql@lines@measure{%

```

```

4447 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@lines@measure
4448 \eql@measure@init\eql@lines@measure@line@begin\eql@lines@measure@line@end
4449 \ifdefined\eql@amp@mode
4450 \let\&\eql@break@amp
4451 \fi
4452 \eql@totalrows@\@M
4453 \eql@shape@select

4454 \setbox\z@\vbox{\measuring@true\halign{%
4455 \global\let\eql@cell@container\@empty
4456 \setbox\eql@cellbox@\hbox{%
4457 \eql@strut@cell
4458 \@lign
4459 $\m@th\eql@mathstyle
4460 \eql@hook@colin
4461 ##%
4462 \eql@punct@apply@col
4463 \eql@hook@colout
4464 $%
4465 }%
4466 \ifdefined\eql@shape@lastrow
4467 \eql@totalrows@\eql@row@
4468 \fi
4469 \eql@shape@eval
4470 \eql@cell@container
4471 \eql@lines@measure@cell
4472 \eql@measure@tag
4473 \eql@measure@endrow
4474 \crcr

4475 \noalign{%
4476 \global\let\eql@shape@lastrow\eql@false
4477 \eql@hook@blockbefore
4478 }%
4479 \eql@hook@blockin
4480 \eql@scan@body
4481 \ifvmode\else
4482 \global\let\eql@shape@lastrow\eql@true
4483 \eql@punct@apply@block
4484 \eql@hook@blockout
4485 \eql@multi@endline
4486 \cr
4487 \fi
4488 \omit
4489 \cr
4490 \noalign{%
4491 \eql@hook@blockafter
4492 \global\let\eql@shape@lastrow\eql@false
4493 }%
4494 }%

4495 \eql@measure@close

4496 \setbox\z@\vbox{%
4497 \unvbox\z@
4498 \unpenalty
4499 \global\setbox\@ne\lastbox
4500 }%
4501 \eql@totalwidth@\wd\@ne

```

```

4502 \eql@verbose@infoarg\eql@verbose@msg@leave\eql@lines@measure
4503 }

```

13.2 Column Placement

TODO: describe Find the best row for tag placement:

```

4504 \def\eql@lines@adjust{%
4505   \eql@tagpos@adjust@eval
4506   \eql@adjust@calc@lines
4507   \eql@numbering@best@eval
4508 }

```

13.3 Print

TODO: describe

```

mes@print@line@begin

```

```

4509 \def\eql@lines@print@line@begin{%
4510   \eql@verbose@info\eql@verbose@msg@startline@number
4511   \eql@numbering@print@line@begin
4512   \eql@hook@linein
4513 }

```

TODO: describe

```

4514 \def\eql@lines@print@line@end{%
4515   \eql@punct@apply@line
4516   \eql@hook@lineout
4517 }

```

TODO: describe

```

4518 \def\eql@lines@print@line@adjust{%
4519   \ifdefined\eql@frame@cmd
4520     \ifcase\eql@shape@pos@
4521       \eql@frame@measure
4522       \advance\eql@shape@amount@-\eql@frame@margin@
4523     \or\or
4524       \eql@frame@measure
4525       \advance\eql@shape@amount@+\eql@frame@margin@
4526     \fi
4527     \eql@frame@adjust
4528   \fi
4529   \eql@cellwidth@\wd\eql@cellbox@
4530   \eql@line@height@\ht\eql@cellbox@
4531   \eql@line@depth@\dp\eql@cellbox@
4532   \eql@numbering@print@line@eval
4533   \if@eqnsw
4534     \eql@tagbox@make\eql@composetag@print
4535   \fi
4536   \eql@tagpos@print@line@eval
4537   \eql@arrange@print@line
4538   \eql@tagpos@print@line@end
4539 }

```

TODO: describe

```

4540 \def\eq@lines@print{%
4541   \eq@verbose@infoarg\eq@verbose@msg@enter\eq@lines@print
4542   \eq@arrange@init
4543   \eq@display@halign@init\eq@lines@print@line@begin
4544   \eq@multi@cr@let\eq@lines@print@line@end
4545   \ifdefined\eq@amp@mode
4546     \let\&\eq@break@amp
4547   \fi
4548   \tabskip\z@skip

4549   \halign{%
4550     \global\let\eq@cell@container\@empty
4551     \setbox\eq@cellbox@\hbox{%
4552       \eq@restore@hfuzz
4553       \eq@strut@cell
4554       \@lign
4555       $\m@th\eq@mathstyle
4556       \eq@hook@colin
4557       ##%
4558       \eq@punct@apply@col
4559       \eq@hook@colout
4560       \eq@tagging@mathsave
4561       $%
4562       \hfil
4563       \kern\z@
4564     }%
4565     \eq@shape@eval
4566     \eq@cell@container
4567     \eq@lines@print@line@adjust
4568   \crrr

4569   \noalign{%
4570     \eq@display@halign@start
4571     \eq@numbering@print@block@begin
4572     \eq@hook@blockbefore
4573   }%
4574   \eq@hook@blockin
4575   \eq@scan@body
4576   \ifvmode\else
4577     \relax
4578     \eq@punct@apply@block
4579     \eq@hook@blockout
4580     \eq@multi@endline
4581     \cr
4582   \fi
4583   \noalign{%
4584     \eq@hook@blockafter
4585     \eq@display@halign@end
4586     \eq@verbose@infoarg\eq@verbose@msg@leave\eq@lines@print
4587   }%
4588   \eq@tagging@tablesavelines
4589 }%
4590 }

```

14 Multi-Line with Multiple Columns

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

14.1 Support

TODO: describe

```
\eql@columns@add@amp
@columns@completerow
4591 \def\eql@columns@add@amp#1{\if m#1&\omit\expandafter\eql@columns@add@amp\fi}
4592 \def\eql@columns@completerow{%
4593   \count@=\numexpr\eql@totalcolumns@+\@ne-\eql@column@\relax
4594   \edef\eql@tmp{%
4595     \expandafter\eql@columns@add@amp\romannumeral\number\count@ 000q}%
4596   \eql@tmp
4597 }

4598 \def\eql@columns@overfull{%
4599   \dimen@=\eql@line@width@
4600   \advance\dimen@-\hfuzz
4601   \ifdim\dimen@>\displaywidth
4602     \setbox\z@ \hbox to \displaywidth{\hbox to \eql@line@width@{\hfil}}%
4603     \wd\z@=\z@
4604     \ht\z@=\eql@line@height@
4605     \dp\z@=\eql@line@depth@
4606     \box\z@
4607   \fi
4608 }
```

14.2 Column Breaks

TODO: describe

TODO: describe

```
4609 \let\eql@amp@org\&

4610 \def\eql@columns@amp{%
4611   \eql@ampprotecttwo\eql@columns@amp@testescape\eql@amp@org
4612   \eql@columns@amp@process}
4613 \def\eql@columns@amp@testescape#1#2{\eql@ifnextgobble@tight/{#1}{%
4614   \relax
4615   \let\eql@punct@term\eql@false
4616   \let\eql@class@rel@composed\@empty
4617   \eql@columns@amp@test{#2}}}
```

TODO: describe

```
4618 \def\eql@columns@amp@test@setopt{%
4619   \let\eql@columns@amp@test\eql@columns@amp@testopt}
4620 \def\eql@columns@amp@test@setall{%
4621   \let\eql@columns@amp@test\eql@columns@amp@testall}
```

@columns@amp@testopt **TODO:** describe

```
4622 \let\eql@columns@amp@testopt\@empty
```

@columns@amp@testall **TODO:** describe

ns@amp@testall@parse

```
4623 \def\eql@columns@amp@testall{\eql@parseopt@cr\eql@columns@amp@testall@parse}
4624 \def\eql@columns@amp@testall@parse{%
4625   \ifx\eql@parseopt@token.%
```

```

4626 \let\eql@parseopt@next\eql@parseopt@punctpass
4627 \fi
4628 \ifx\eql@parseopt@token,%
4629 \let\eql@parseopt@next\eql@parseopt@punctpass
4630 \fi
4631 \ifx\eql@parseopt@token~%
4632 \let\eql@parseopt@next\eql@parseopt@punctpass
4633 \fi
4634 \ifx\eql@parseopt@token'%
4635 \let\eql@parseopt@next\eql@parseopt@punctnext
4636 \fi
4637 \ifx\eql@parseopt@token!%
4638 \let\eql@parseopt@next\eql@parseopt@punctterm
4639 \fi
4640 \ifx\eql@parseopt@token=%
4641 \let\eql@parseopt@next\eql@parseopt@relsymp
4642 \fi
4643 \ifx\eql@parseopt@token;%
4644 \let\eql@parseopt@next\eql@parseopt@relcont
4645 \fi
4646 \ifx\eql@parseopt@token:%
4647 \let\eql@parseopt@next\eql@parseopt@relstart
4648 \fi
4649 \ifx\eql@parseopt@token|%
4650 \let\eql@parseopt@next\eql@parseopt@relord
4651 \fi
4652 \ifx\eql@parseopt@token?%
4653 \let\eql@parseopt@next\eql@columns@amp@parse@rel
4654 \fi
4655 \ifx\eql@parseopt@token&%
4656 \let\eql@parseopt@next\eql@parseopt@end
4657 \fi
4658 }
4659 \def\eql@columns@amp@parse@rel#1#2{%
4660 \def\eql@class@rel@composed{\eql@class@rel@make{#2}}%
4661 \eql@parseopt@end}

```

@columns@amp@process

```

4662 \def\eql@columns@amp@process{%
4663 \ifdefined\eql@punct@term\eql@punct@apply@top\fi
4664 \edef\eql@tmp{%
4665 \ifx\eql@class@rel@composed\@empty
4666 \ifodd\eql@column@&\noexpand\eql@punct@next@clear\fi%
4667 \else
4668 \ifodd\eql@column@\else&\fi%
4669 \unexpanded\expandafter{\eql@class@rel@composed}%
4670 \fi
4671 }%
4672 \eql@tmp
4673 }

```

14.3 Transpose

TODO: describe

TODO: adjust to \&?!

TODO: describe

```
4674 \let\eql@transpose@active\eql@false
4675 \def\eql@transpose@end{\eql@transpose@end}
4676 \def\eql@transpose@skip{&\eql@punct@next@clear}
4677 \def\eql@transpose@complete{%
4678   \relax\ifodd\eql@column@\expandafter\eql@transpose@skip\fi&}
```

TODO: describe

```
4679 \def\eql@transpose{%
4680   \eql@totalcolumns@\z@
4681   \eql@totalrows@\z@
4682   \expandafter\eql@transpose@scan@col\the\eql@scan@reg@&\eql@transpose@end&
4683   \eql@scan@reg@{ }%
4684   \eql@row@\z@
4685   \eql@transpose@output@row
4686 }
```

TODO: describe

```
4687 \def\eql@transpose@save@col#1{%
4688   \@namedef{eql@transpose@data@col@\the\eql@totalcolumns@}{%
4689     \ifcase\eql@row@#1\else\let\eql@tmp\eql@transpose@skip\fi}}
```

TODO: describe

```
4690 \def\eql@transpose@scan@col#1&{%
4691   \def\eql@tmpa{#1}%
4692   \ifx\eql@tmpa\eql@transpose@end\else
4693     \advance\eql@totalcolumns@\@ne
4694     \eql@row@\z@
4695     \let\eql@transpose@data@col\empty
4696     \eql@transpose@scan@row#1\\eql@transpose@end\\%
4697     \ifnum\eql@row@>\eql@totalrows@
4698       \eql@totalrows@\eql@row@
4699     \fi
4700     \expandafter\eql@transpose@save@col\expandafter{\eql@transpose@data@col}%
4701     \expandafter\eql@transpose@scan@col
4702   \fi
4703 }
```

TODO: describe

```
4704 \def\eql@transpose@append@row#1{%
4705   \advance\eql@row@\@ne
4706   \eql@append\eql@transpose@data@col{\or\def\eql@tmp{#1}}}
```

TODO: describe

```
4707 \def\eql@transpose@scan@row#1\\{%
4708   \def\eql@tmpa{#1}%
4709   \ifx\eql@tmpa\eql@transpose@end\else
4710     \ifx\eql@transpose@active+
4711       \eql@transpose@scan@cell#1&\eql@transpose@end&%
4712     \else
4713       \eql@transpose@append@row{#1}%
4714     \fi
4715     \expandafter\eql@transpose@scan@row
4716   \fi
4717 }
```

TODO: describe

```
4718 \def\eql@transpose@scan@cell#1&#2&{%
4719   \def\eql@tmpa{#2}%
4720   \ifx\eql@tmpa\eql@transpose@end
4721     \eql@transpose@append@row{#1}%
4722   \else
4723     \eql@transpose@append@row{#1&#2}%
4724     \expandafter\eql@transpose@scan@cell@next
4725   \fi
4726 }
```

TODO: describe

```
4727 \def\eql@transpose@scan@cell@next#1&{%
4728   \def\eql@tmpa{#1}%
4729   \ifx\eql@tmpa\eql@transpose@end\else
4730     \eql@transpose@append@row{&#1}%
4731     \expandafter\eql@transpose@scan@cell@next
4732   \fi
4733 }
```

TODO: describe

```
4734 \def\eql@transpose@output@row{%
4735   \ifnum\eql@row@<\eql@totalrows@
4736     \advance\eql@row@\@ne
4737     \eql@column@\z@
4738     \eql@transpose@output@col
4739     \ifnum\eql@row@<\eql@totalrows@
4740       \eql@scan@addto\\%
4741     \fi
4742     \expandafter\eql@transpose@output@row
4743   \fi
4744 }
```

TODO: describe

```
4745 \def\eql@transpose@output@col{%
4746   \ifnum\eql@column@<\eql@totalcolumns@
4747     \advance\eql@column@\@ne
4748     \csname eql@transpose@data@col@\the\eql@column@\endcsname
4749     \expandafter\eql@scan@addto\expandafter{\eql@tmp}%
4750     \ifnum\eql@column@<\eql@totalcolumns@
4751       \eql@scan@addto{\eql@transpose@complete}%
4752     \fi
4753     \expandafter\eql@transpose@output@col
4754   \fi
4755 }
```

14.4 Measure

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

s@measure@line@begin

```
4756 \def\eql@columns@measure@line@begin{%
4757   \eql@verbose@info\eql@verbose@msg@startline@number
4758   \global\eql@column@\z@
4759   \global\eql@line@height@\z@
```

```

4760 \global\eql@line@depth@\z@
4761 \eql@numbering@measure@line@begin
4762 \eql@hook@linein
4763 }

4764 \def\eql@columns@measure@cell{%
4765 \eql@cellwidth@\wd\eql@cellbox@
4766 \ifdefined\eql@frame@cmd
4767 \eql@frame@measure
4768 \advance\eql@cellwidth@\eql@frame@margin@
4769 \fi
4770 \ifdim\ht\eql@cellbox@>\eql@line@height@
4771 \global\eql@line@height@\ht\eql@cellbox@
4772 \fi
4773 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4774 \global\eql@line@depth@\dp\eql@cellbox@
4775 \fi
4776 \ifnum\eql@column@=\@ne
4777 \eql@dimensions@startrow
4778 \fi
4779 \ifodd\eql@column@
4780 \eql@shape@pos@\tw@
4781 \else
4782 \eql@shape@pos@\z@
4783 \fi
4784 \eql@shape@amount@\z@
4785 \eql@dimensions@savecell
4786 \ifodd\eql@column@\else
4787 \eql@dimensions@savesep
4788 \fi
4789 \kern\eql@cellwidth@
4790 }

```

mns@measure@line@end

```

4791 \def\eql@columns@measure@line@end{%
4792 \eql@punct@apply@line
4793 \eql@hook@lineout
4794 &\omit
4795 \ifnum\eql@column@>\eql@totalcolumns@
4796 \global\eql@totalcolumns@\eql@column@
4797 \fi

```

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

```

4798 \ifdefined\eql@frame@cmd
4799 \advance\eql@column@\@ne
4800 \wd\eql@cellbox@\z@
4801 \eql@columns@measure@cell
4802 \fi
4803 \eql@measure@tag
4804 \eql@measure@endrow
4805 }

```

\eql@columns@measure

```

4806 \def\eql@columns@measure{%

```

```

4807 \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@measure
4808 \eql@totalcolumns@z@
4809 \eql@measure@init\eql@columns@measure@line@begin\eql@columns@measure@line@end
4810 \ifdefined\eql@amp@mode
4811 \let\&\eql@columns@amp
4812 \fi

4813 \setbox\z@\vbox{\measuring@true\halign{%
4814 &%
4815 \global\advance\eql@column@ \@ne
4816 \global\let\eql@cell@container\@empty
4817 \global\setbox\eql@cellbox@\hbox{%
4818 \eql@strut@cell
4819 \@lign
4820 $\m@th\eql@mathstyle
4821 \eql@hook@colin
4822 ##%
4823 \eql@punct@apply@next
4824 \eql@class@innerleft
4825 \eql@hook@innerleft
4826 $%
4827 }%
4828 \eql@cell@container
4829 \hfil
4830 \eql@columns@measure@cell
4831 \global\let\eql@frame@prevcmd\eql@frame@cmd
4832 &%
4833 \eql@prevwidth@\wd\eql@cellbox@
4834 \let\eql@frame@cmd\eql@frame@prevcmd
4835 \global\advance\eql@column@ \@ne
4836 \global\let\eql@cell@container\@empty
4837 \setbox\eql@cellbox@\hbox{%
4838 \eql@strut@cell
4839 \@lign
4840 $\m@th\eql@mathstyle
4841 \eql@hook@innerright
4842 \eql@class@innerright@sel
4843 ##%
4844 \eql@punct@apply@col
4845 \eql@hook@colout
4846 $%
4847 }%
4848 \eql@cell@container
4849 \eql@columns@measure@cell
4850 \hfil
4851 \crcr

4852 \noalign{%
4853 \eql@hook@blockbefore
4854 }%
4855 \eql@hook@blockin
4856 \eql@scan@body

4857 \ifvmode\else
4858 \eql@punct@apply@block
4859 \eql@hook@blockout
4860 \eql@multi@endline
4861 \cr
4862 \fi

```

```

4863 \noalign{%
4864 \eq@hook@blockafter
4865 }%

```

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

```

4866 \omit
4867 \eq@column@ \@ne
4868 \eq@columns@completerow
4869 \cr
4870 }}%

4871 \eq@measure@close

4872 \setbox\z@\vbox{%
4873 \unvbox\z@
4874 \unpenalty
4875 \global\setbox\@ne\lastbox
4876 }%
4877 \eq@totalwidth@\wd\@ne

```

TODO: why not recycle box contents altogether?!

```

4878 \let\eq@colwidth@tab\@empty
4879 \loop
4880 \setbox\@ne\hbox{%
4881 \unhbox\@ne
4882 \unskip
4883 \global\setbox\thr@\@ne\lastbox
4884 }%
4885 \ifhbox\thr@@
4886 \eq@colwidth@save{\wd\thr@@}%
4887 \repeat

4888 \eq@verbose@infoarg\eq@verbose@msg@leave\eq@columns@measure
4889 }

```

14.5 Columns Placement

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

```

4890 \def\eq@columns@adjust{%
4891 \ifodd\eq@totalcolumns@
4892 \advance\eq@totalcolumns@\@ne
4893 \fi
4894 \eq@tagpos@adjust@eval
4895 \eq@adjust@calc@columns
4896 }

```

14.6 Print

TODO: describe

```

\eq@columns@print@line@begin

```

```

4897 \def\eq@columns@print@line@begin{%
4898 \eq@verbose@info\eq@verbose@msg@startline@number

```

```

4899 \global\eql@column@\z@
4900 \global\eql@line@pos@\eql@marginleft@
4901 \global\eql@line@width@\z@
4902 \global\eql@line@avail@\eql@totalwidth@
4903 \global\eql@line@height@\z@
4904 \global\eql@line@depth@\z@
4905 \eql@numbering@print@line@begin
4906 \eql@hook@linein
4907 }

```

l@columns@print@cell

```

4908 \def\eql@columns@print@cell{%
4909 \eql@cellwidth@\wd\eql@cellbox@
4910 \ifodd\eql@column@
4911 \ifdefined\eql@frame@cmd
4912 \eql@frame@measure
4913 \advance\eql@cellwidth@\eql@frame@margin@
4914 \fi
4915 \dimen@\z@
4916 \else
4917 \advance\eql@cellwidth@-\eql@prevwidth@

```

draw a frame

```

4918 \ifdefined\eql@frame@cmd
4919 \eql@frame@measure
4920 \advance\eql@cellwidth@\eql@frame@margin@
4921 \advance\eql@prevwidth@\eql@frame@margin@
4922 \eql@frame@print
4923 \fi

```

update height and depth

```

4924 \ifdim\ht\eql@cellbox@>\eql@line@height@
4925 \global\eql@line@height@\ht\eql@cellbox@
4926 \fi
4927 \ifdim\dp\eql@cellbox@>\eql@line@depth@
4928 \global\eql@line@depth@\dp\eql@cellbox@
4929 \fi

```

print box

```

4930 \kern-\eql@prevwidth@
4931 \unhbox\eql@cellbox@
4932 \dimen@-\eql@cellwidth@
4933 \fi

```

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

```

4934 \advance\dimen@\eql@colwidth@get\eql@column@\relax
4935 \kern\dimen@

```

update available and used space

```

4936 \dimen@\eql@colwidth@get\eql@column@\relax
4937 \ifdim\eql@cellwidth@>\z@
4938 \ifdim\eql@line@width@=\z@
4939 \eql@line@avail@\eql@line@pos@
4940 \ifodd\eql@column@
4941 \advance\eql@line@avail@\dimen@

```



```

4942     \advance\eql@line@avail@-\eql@cellwidth@
4943     \fi
4944     \global\eql@line@avail@\eql@line@avail@
4945     \fi
4946     \eql@line@width@\eql@line@pos@
4947     \ifodd\eql@column@
4948         \advance\eql@line@width@\dimen@
4949     \else
4950         \advance\eql@line@width@\eql@cellwidth@
4951     \fi
4952     \global\eql@line@width@\eql@line@width@
4953 \fi
4954 \advance\eql@line@pos@\dimen@
4955 \ifodd\eql@column@\else
4956     \advance\eql@line@pos@\eql@colsep@
4957 \fi
4958 \global\eql@line@pos@\eql@line@pos@
4959 }

4960 \def\eql@columns@print@trailright{%
4961     &\omit
4962     \eql@prevwidth@\wd\eql@cellbox@
4963     \let\eql@frame@cmd\eql@frame@prevcmd
4964     \global\advance\eql@column@\@ne
4965     \eql@columns@print@cell
4966 }

```

lums@print@line@end

```

4967 \def\eql@columns@print@line@end{%
4968     \eql@punct@apply@line
4969     \eql@hook@lineout
4970 % \TODO add an even column with empty stuff if box processing deferred
4971     \ifodd\eql@column@
4972         \expandafter\eql@columns@print@trailright
4973     \fi
4974     \eql@columns@completerow
4975     \eql@columns@print@tag
4976 }

```

ql@columns@print@tag

```

4977 \def\eql@columns@print@tag{%
4978     \kern-\dimexpr\eql@totalwidth@+\eql@colsep@\relax

```

determine first line available space

```

4979     \eql@display@firstavail@set\eql@line@avail@
4980     \eql@columns@overfull
4981     \eql@numbering@print@line@eval
4982     \if@eqnsw
4983         \eql@tagbox@make\eql@composetag@print
4984     \fi
4985     \eql@tagpos@print@line@eval
4986     \eql@tagbox@print@cell
4987     \eql@tagpos@print@line@end
4988 }

```

\eql@columns@print

```

4989 \def\eql@columns@print{%
4990   \eql@verbose@infoarg\eql@verbose@msg@enter\eql@columns@print
4991   \eql@shape@align@disable
4992   \eql@display@halign@init\eql@columns@print@line@begin
4993   \eql@multi@cr@let\eql@columns@print@line@end
4994   \ifdefined\eql@amp@mode
4995     \let\&\eql@columns@amp
4996   \fi
4997   \tabskip\eql@marginleft@

4998   \halign{%
4999     &%
5000     \global\advance\eql@column@ \@ne
5001     \global\let\eql@cell@container\@empty
5002     \global\setbox\eql@cellbox@\hbox{%
5003       \eql@strut@cell
5004       \@lign
5005       $\m@th\eql@mathstyle
5006       \eql@hook@colin
5007       ##%
5008       \eql@punct@apply@next
5009       \eql@class@innerleft
5010       \eql@hook@innerleft
5011       \eql@tagging@mathsave
5012       $%
5013       \eql@tagging@mathaddlast
5014     }%
5015     \eql@cell@container
5016     \hfil
5017     \eql@columns@print@cell
5018     \global\let\eql@frame@prevcmd\eql@frame@cmd
5019     \tabskip\z@skip
5020     &%
5021     \eql@prevwidth@\wd\eql@cellbox@
5022     \let\eql@frame@cmd\eql@frame@prevcmd
5023     \global\advance\eql@column@ \@ne
5024     \global\let\eql@cell@container\@empty
5025     \setbox\eql@cellbox@\hbox{%
5026       \unhbox\eql@cellbox@
5027       \eql@strut@cell
5028       \@lign
5029       $\m@th\eql@mathstyle
5030       \eql@hook@innerright
5031       \eql@class@innerright@sel
5032       ##%
5033       \eql@punct@apply@col
5034       \eql@hook@colout
5035       \eql@tagging@mathsave
5036       $%
5037       \eql@tagging@mathaddlast
5038     }%
5039     \eql@cell@container
5040     \eql@columns@print@cell
5041     \hfil
5042     \tabskip\eql@colsep@\relax
5043   \crcr

5044   \noalign{%
5045     \eql@display@halign@start

```

```

5046     \eql@numbering@print@block@begin
5047     \eql@hook@blockbefore
5048 }%
5049 \eql@hook@blockin
5050 \eql@scan@body
5051 \ifvmode\else
5052     \relax
5053     \eql@punct@apply@block
5054     \eql@hook@blockout
5055     \eql@multi@endline
5056     \cr
5057 \fi
5058 \noalign{%
5059     \eql@hook@blockafter
5060     \eql@display@halign@end
5061     \eql@verbose@infoarg\eql@verbose@msg@leave\eql@columns@print
5062 }%
5063 \eql@tagging@tables@savealign
5064 }%
5065 }

```

15 Interface

15.1 Options Processing

TODO: describe

```

5066 \def\eql@equations@test@setopt{\let\eql@equations@test\eql@equations@testopt}
5067 \def\eql@equations@test@setall{\let\eql@equations@test\eql@equations@testall}

```

`\eql@equations@testopt` **TODO:** describe

```

5068 \def\eql@equations@testopt#1{%
5069     \eql@testopt@tight{\eql@equations@testopt@set{#1}}{}}
5070 \def\eql@equations@testopt@set#1[#2]{\eqnaddopt{#2}#1}

```

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

```

5071 \def\eql@equations@testall{\eql@parseopt@env\eql@equations@testall@parse}
5072 \def\eql@equations@testall@parse{%
5073     \ifx\eql@parseopt@token*%
5074         \let\eql@parseopt@next\eql@parseopt@nonumber
5075     \fi
5076     \ifx\eql@parseopt@token!%
5077         \let\eql@parseopt@next\eql@parseopt@donumber
5078     \fi
5079     \ifx\eql@parseopt@token/%
5080         \let\eql@parseopt@next\eql@parseopt@transpose
5081     \fi
5082     \ifx\eql@parseopt@token[%
5083         \let\eql@parseopt@next\eql@parseopt@opt
5084     \fi
5085     \ifx\eql@parseopt@token\eql@atxi
5086         \let\eql@parseopt@next\eql@parseopt@label

```

```

5087 \fi
5088 \ifx\eql@parseopt@token\eql@atxii
5089 \let\eql@parseopt@next\eql@parseopt@label
5090 \fi
5091 \ifx\eql@parseopt@token.%
5092 \let\eql@parseopt@next\eql@parseopt@punctpass
5093 \fi
5094 \ifx\eql@parseopt@token,%
5095 \let\eql@parseopt@next\eql@parseopt@punctpass
5096 \fi
5097 \ifx\eql@parseopt@token~%
5098 \let\eql@parseopt@next\eql@parseopt@punctpass
5099 \fi
5100 \ifx\eql@parseopt@token'
5101 \let\eql@parseopt@next\eql@parseopt@punctopt
5102 \fi
5103 \ifx\eql@parseopt@token-%
5104 \let\eql@parseopt@next\eql@parseopt@single
5105 \fi
5106 \ifx\eql@parseopt@token=%
5107 \let\eql@parseopt@next\eql@parseopt@lines
5108 \fi
5109 \ifx\eql@parseopt@token|
5110 \let\eql@parseopt@next\eql@parseopt@columns
5111 \fi
5112 \ifx\eql@parseopt@token<%
5113 \let\eql@parseopt@next\eql@parseopt@ampeq
5114 \fi
5115 \ifx\eql@parseopt@token>%
5116 \let\eql@parseopt@next\eql@parseopt@eqamp
5117 \fi
5118 \ifx\eql@parseopt@token\label
5119 \let\eql@parseopt@next\eql@parseopt@end
5120 \fi
5121 \ifx\eql@parseopt@token\begin
5122 \let\eql@parseopt@next\eql@parseopt@end
5123 \fi
5124 }

```

equations@end@testall **TODO:** describe

```

5125 \def\eql@equations@end@testall{%
5126 \eql@parseopt@env\eql@equations@end@testall@parse}
5127 \def\eql@equations@end@testall@parse{%
5128 \ifx\eql@parseopt@token.%
5129 \let\eql@parseopt@next\eql@parseopt@punctpass
5130 \fi
5131 \ifx\eql@parseopt@token,%
5132 \let\eql@parseopt@next\eql@parseopt@punctpass
5133 \fi
5134 \ifx\eql@parseopt@token~%
5135 \let\eql@parseopt@next\eql@parseopt@punctpass
5136 \fi
5137 \ifx\eql@parseopt@token'
5138 \let\eql@parseopt@next\eql@parseopt@punctblock
5139 \fi
5140 }

```

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:

```

5141 \def\eql@equations@processopt{%
5142   \let\eql@tags@container@block\eql@tags@container@clear
5143   \let\eql@tags@frame@cmd\@firstofone
5144   \let\eql@skip@force@above\@undefined
5145   \let\eql@skip@force@below\@undefined
5146   \let\eql@skip@force@leave\@undefined
5147   \let\eql@display@linewidth\@undefined
5148   \let\eql@display@marginleft\@undefined
5149   \let\eql@display@marginright\@undefined
5150   \eql@abovespace@\z@skip
5151   \eql@belowspace@\z@skip
5152   \eql@displaybreak@prepen@\@MM
5153   \eql@displaybreak@postpen@\@MM
5154   \eql@nextopt@process{equations}%
5155   \let\eql@punct@next\@undefined
5156   \eql@indent@\glueexpr\eql@indent@val\relax
5157   \eql@tagsepmin@\glueexpr\eql@tagsepmin@val\relax
5158 }

```

15.2 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:

```

5159 \def\eql@single@cr@error{%
5160   \eql@error{Cannot use '\string\\' within display equation.
5161     Please switch to equations environment}%
5162 }

```

`\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:

```

5163 \def\eql@single@start{%
5164   \eql@display@enter
5165   \eql@tagging@start
5166   \eql@dollar@dollar@begin
5167   \eql@display@adjust
5168   \eql@numbering@init
5169   \eql@stack@save@equations
5170   \eql@numbering@single@init
5171   \ifdefined\eql@single@cr@mode
5172     \let\\\eql@single@cr@mode
5173   \fi
5174   \ifdefined\eql@amp@mode
5175     \let&\eql@break@amp
5176   \fi
5177   \ifdefined\eql@single@native
5178     \let\eql@single@start@sel\eql@single@start@native
5179     \let\eql@single@end@sel\eql@single@end@native
5180   \else
5181     \let\eql@single@start@sel\eql@single@start@print

```

```

5182 \let\eql@single@end@sel\eql@single@end@print
5183 \fi
5184 \eql@single@start@sel
5185 }

```

`\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:

```

5186 \def\eql@single@end{%
5187 \eql@punct@apply@block
5188 \eql@hook@eqout
5189 \eql@single@end@sel
5190 \global\eql@punct@top@reset
5191 \eql@stack@restore
5192 \eql@dollar@dollar@end
5193 \eql@tagging@end
5194 \eql@display@leave
5195 }

```

`\eql@single@main` Combined opening, body and closing for pre-scanned body:

```

5196 \def\eql@single@main{%
5197 \eql@single@start
5198 \eql@scan@body
5199 \eql@single@end
5200 }

```

`equations@single@set` Configure equations macros to single-line mode:

```

5201 \def\eql@equations@single@set{%
5202 \ifdefined\eql@single@doscan
5203 \let\eql@equations@main\eql@single@main
5204 \else
5205 \let\eql@equations@main\@undefined
5206 \fi
5207 }

```

15.3 Multi-Line Main

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

```

5208 \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

```

5209 \def\eql@multi@main{%
5210 \eql@display@enter
5211 \eql@tagging@start
5212 \eql@dollar@dollar@begin
5213 \eql@display@adjust
5214 \eql@numbering@init
5215 \eql@stack@save@equations
5216 \ifdefined\eql@transpose@active
5217 \ifdefined\eql@multi@linesmode\else

```

```

5218     \eql@transpose
5219     \fi
5220 \fi
5221 \ifdefined\eql@numbering@subeq@use
5222     \eql@numbering@subeq@init
5223 \fi
5224 \eql@display@init
5225 \let\intertext\eql@intertext
5226 \let\endintertext\endeql@intertext
5227 \eql@shape@align@enable

```

Now measure the given multi-line equations body:

```

5228 \ifdefined\eql@multi@linesmode
5229     \eql@lines@measure
5230 \else
5231     \ifdefined\eql@ampproof@active
5232         \eql@ampproof
5233     \fi
5234     \eql@columns@measure
5235 \fi

```

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

```

5236 \ifdefined\eql@numbering@subeq@use
5237     \eql@numbering@subeq@test
5238 \fi
5239 \ifdefined\eql@multi@linesmode\else
5240     \ifdefined\eql@multi@linesfallback
5241         \ifnum\eql@totalcolumns@=\@ne
5242             \let\eql@multi@linesmode\eql@true
5243             \ifx\eql@multi@linesfallback\z@\else
5244                 \eql@lines@measure
5245             \fi
5246         \fi
5247     \fi
5248 \fi

```

Adjust the multi-line equations body:

```

5249 \ifdefined\eql@multi@linesmode
5250     \eql@lines@adjust
5251 \else
5252     \eql@columns@adjust
5253 \fi

```

Now print the multi-line equations body:

```

5254 \eql@display@print
5255 \eql@numbering@print@init
5256 \ifdefined\eql@multi@linesmode
5257     \eql@lines@print
5258 \else
5259     \eql@columns@print
5260 \fi
5261 \eql@display@close

```

Close numbering, restore global variables, end display math, indicate end to PDF tagging,

return to vertical mode if desired:

```

5262 \ifdefined\eql@numbering@subeq@use
5263 \eql@numbering@subeq@close
5264 \fi
5265 \global\eql@punct@top@reset
5266 \eql@stack@restore
5267 \eql@dollar@dollar@end
5268 \eql@tagging@end
5269 \eql@display@leave
5270 }
```

equations@columns@set Configure equations macros to one of the two multi-line modes:

@equations@lines@set

```

5271 \def\eql@equations@columns@set{%
5272 \let\eql@equations@main\eql@multi@main
5273 \let\eql@multi@linesmode\eql@false
5274 }
5275 \def\eql@equations@lines@set{%
5276 \let\eql@equations@main\eql@multi@main
5277 \let\eql@multi@linesmode\eql@true
5278 }
```

15.4 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`:

```

5279 \newenvironment{equations}{%
5280 \ifmmode
5281 \expandafter\eql@equations@env@cancel
5282 \else
5283 \eql@verbose@info\eql@verbose@msg@enterenv
5284 \expandafter\eql@equations@env@open
5285 \fi
5286 }{%
5287 \ifdefined\eql@equations@main\else
5288 \expandafter\eql@single@end
5289 \fi
5290 \ignorespacesafterend
5291 \eql@verbose@info\eql@verbose@msg@leaveenv
5292 }
5293 \eql@markline@amsthm@register{equations}
5294 \eql@tagging@register@luamml{equations}
```

equations@env@cancel

```

5295 \def\eql@equations@env@cancel{%
5296 \eql@error@mathmode{\string\begin{\@currentenv}}%
5297 \let\eql@scan@call\eql@scan@env@cancel
5298 \eql@scan@env
5299 }
```


`\equations@env@open`

```

5300 \def\eq@equations@env@open{%
5301   \ifdefined\eq@equations@env@modifier
5302     \eq@equations@test@setall
5303   \else
5304     \eq@equations@nomodifier
5305   \fi
5306   \eq@ampprotect\eq@equations@test\eq@equations@env@start
5307 }

```

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

```

5308 \def\eq@equations@env@start{%
5309   \eq@equations@processopt
5310   \ifdefined\eq@equations@main
5311     \eq@equations@call@set
5312     \expandafter\eq@scan@env
5313   \else
5314     \expandafter\eq@single@start
5315   \fi
5316 }

```

`\eq@equations@call`

```

5317 \def\eq@equations@call{\eq@equations@main\eq@scan@end}
5318 \def\eq@equations@call@test{%
5319   \eq@ampprotect\eq@equations@end@testall\eq@equations@call}
5320 \def\eq@equations@call@set{%
5321   \ifdefined\eq@equations@end@modifier
5322     \let\eq@scan@call\eq@equations@call@test
5323   \else
5324     \let\eq@scan@call\eq@equations@call
5325   \fi
5326 }

```

Square Brackets.

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

```

5327 \newenvironment{equations@sqr}{}{}
5328 \eq@markline@amsthm@register{equations@sqr}
5329 \eq@tagging@register@luamml{equations@sqr}

```

`\equations@sqr@open` Definition for ‘`[`’. Add the default arguments `\eq@equations@sqr@opt`, enter the pseudo-environment, scan for optional arguments, and pass on to `\eq@equations@sqr@start`:

```

5330 \def\eq@equations@sqr@open{%
5331   \expandafter\eqnaddopt\expandafter{\eq@equations@sqr@opt}%
5332   \begin{equations@sqr}%
5333   \eq@verbose@info\eq@verbose@msg@enterenv
5334   \let\]\eq@equations@sqr@close
5335   \ifdefined\eq@equations@sqr@modifier
5336     \eq@equations@test@setall

```

```

5337 \else
5338     \eqlequations@nomodifier
5339 \fi
5340 \eql@ampprotect\eqlequations@test\eqlequations@sqr@start
5341 }

```

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

```

5342 \def\eqlequations@sqr@start{%
5343     \eqlequations@processopt
5344     \ifdefined\eqlequations@main
5345         \eqlequations@call@set
5346         \expandafter\eql@scan@sqr
5347     \else
5348         \expandafter\eql@single@start
5349     \fi
5350 }

```

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

```

5351 \protected\def\eqlequations@sqr@close{%
5352     \ifdefined\eqlequations@main
5353         \let\eqlequations@end@modifier\eql@false
5354     \fi
5355     \ifdefined\eqlequations@end@modifier
5356         \expandafter\eql@ampprotect\expandafter\eqlequations@end@testall
5357     \fi
5358     \eqlequations@sqr@end
5359 }

```

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

```

5360 \def\eqlequations@sqr@end{%
5361     \ifdefined\eqlequations@main\else
5362         \expandafter\eql@single@end
5363     \fi
5364     \eql@verbose@info\eql@verbose@msg@leaveenv
5365     \end{equations@sqr}%
5366     \ignorespaces
5367 }

```

Angle Brackets.

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

```

5368 \newenvironment{equations@ang}{}{}
5369 \eql@markline@amsthm@register{equations@ang}
5370 \eql@tagging@register@luamml{equations@ang}

```

`l@equations@ang@open`

```

5371 \def\eqlequations@ang@open{%
5372     \expandafter\eqnaddopt\expandafter{\eqlequations@ang@opt}%
5373     \begin{equations@ang}%
5374     \eql@verbose@info\eql@verbose@msg@enterenv
5375     \let\>\eqlequations@ang@close

```

```

5376 \ifdefined\eqlequations@ang@modifier
5377 \eqlequations@test@setall
5378 \else
5379 \eqlequations@nomodifier
5380 \fi
5381 \eql@ampprotect\eqlequations@test\eqlequations@ang@start
5382 }

```

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

```

5383 \def\eqlequations@ang@start{%
5384 \eqlequations@processopt
5385 \ifdefined\eqlequations@main
5386 \eqlequations@call@set
5387 \expandafter\eql@scan@ang
5388 \else
5389 \expandafter\eql@single@start
5390 \fi
5391 }

```

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

```

5392 \def\eqlequations@ang@close{%
5393 \ifdefined\eqlequations@main
5394 \let\eqlequations@end@modifier\eql@false
5395 \fi
5396 \ifdefined\eqlequations@end@modifier
5397 \expandafter\eql@ampprotect\expandafter\eqlequations@end@testall
5398 \fi
5399 \eqlequations@ang@end
5400 }

```

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

```

5401 \def\eqlequations@ang@end{%
5402 \ifdefined\eqlequations@main\else
5403 \expandafter\eql@single@end
5404 \fi
5405 \eql@verbose@info\eql@verbose@msg@leaveenv
5406 \end{equations@ang}%
5407 \ignorespaces
5408 }

```

16 Options

16.1 Selection Tools

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

```

5409 \def\eql@decide@abovebelow#1#2#3#4#5{%
5410 \eql@decide@select{#1}{#2}{#3}{%
5411 {abovebelow,both,tb}{#4#5},%
5412 {above,top,t}{#4},%
5413 {below,bottom,b}{#5}}

```

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

```

5414 \def\eq@decide@situation#1#2#3#4{%
5415   \eq@decide@select{#1}{#2}{#3}{%
5416     {{\long}\def#4{0}}},%
5417     {{\short}\def#4{1}}},%
5418     {{\cont}\def#4{2}}},%
5419     {{\par}\def#4{3}}},%
5420     {{\top}\def#4{4}}},%
5421     {{\noskip}\def#4{5}}},%
5422     {{\medskip}\def#4{6}}}}

```

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

```

5423 \def\eq@decide@delim#1#2#3{%
5424   \eq@decide@select{#1}{#2}{#3}{%
5425     {{\,,\eq@decide@false}\eq@box@wrap{}{}},%
5426     {{\eq@decide@true,r,round}\eq@box@delim()}},%
5427     {{s,sqr,square}\eq@box@delim[]{}},%
5428     {{c,curly,braces}\eq@box@delim\lbrace\rbrace}},%
5429     {{a,ang,angle}\eq@box@delim\langle\rangle}},%
5430     {{v,vert}\eq@box@delim\vert\vert}},%
5431     {{d,dvert}\eq@box@delim\Vert\Vert}},%
5432     {\relax\eq@box@delim#3}}}%

```

TODO: describe

```

5433 \def\eq@keyall{equations,box,setup}

```

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:

```

5434 \eq@define@key{box}{lines,ln,gathered,gather,ga}[]{%
5435   \eq@box@lines@set}
5436 \eq@define@key{box}{columns,col,aligned,align,al}[]{%
5437   \eq@box@columns@set}
5438 \eq@define@key{box}{cases}[]{%
5439   \eq@box@cases@set\eq@box@ldelim\lbrace%
5440   \def\eq@box@colsep{\eq@box@condsep}}
5441 \eq@define@key{box}{matrix}[r]{%
5442   \eq@box@lines@set\eq@shape@set{center}%
5443   \let\eq@spread@reset\eq@true\def\eq@spread@val{\z@}%
5444   \def\eq@box@colsep{\eq@box@shortsep}%
5445   \let\eq@mathstyle\@empty
5446   \eq@punct@clear
5447   \eq@box@cr@test@setopt
5448   \eq@box@amp@test@setopt
5449   \let\eq@box@end@modifier\eq@false
5450   \eq@decide@delim{#3}{#2}{#1}}
5451 \eq@define@key{box}{top,t}[]{\let\eq@box@box\vtop}
5452 \eq@define@key{box}{center,c}[]{\let\eq@box@box\vcenter}
5453 \eq@define@key{box}{bottom,b}[]{\let\eq@box@box\vbbox}
5454 \eq@define@key{box}{intro}{%
5455   \def\eq@box@cases@condintro{#1}}
5456 \eq@define@key{box}{introtext}{%
5457   \def\eq@box@cases@condintro{%

```

```

5458 \ifmode\expandafter\hbox\else\expandafter\@firstofone\fi{#1 }}}
5459 \eqld@define@key{box}{textcond}[true]{%
5460 \eqld@decide@select{#3}{#2}{#1}{%
5461 {{\eqld@decide@true,text}}{\let\eqld@box@cases@condtext\eqld@true}},%
5462 {{\eqld@decide@false,math}}{\let\eqld@box@cases@condtext\eqld@false}}}}
5463 \eqld@define@key{setup}{scanbox}[true]{%
5464 \eqld@decide@bool{#3}{#2}{#1}\eqld@box@doscan}
5465 \eqld@define@key{box}{scan}[true]{%
5466 \eqld@decide@bool{#3}{#2}{#1}\eqld@box@doscan}
5467 \eqld@define@key{setup}{boxangopt}[]{%
5468 \def\eqld@box@ang@opt{columns,#1}}

```

Modes for Equations Environment.

```

5469 \let\eqld@box@doscan\eqn@false

```

Declare modes and switches for the equations environment:

```

5470 \eqld@define@key{equations}{single,1,equation,eq}[]{\eqld@equations@single@set}
5471 \eqld@define@key{equations}{lines,ln,gathered,gather,ga}[]{%
5472 \eqld@equations@lines@set}
5473 \eqld@define@key{equations}{columns,col,aligned,align,al}[]{%
5474 \eqld@equations@columns@set}
5475 \eqld@define@key{equations,setup}{transpose}[true]{%
5476 \eqld@decide@select{#3}{#2}{#1}{%
5477 {\eqld@decide@false{\let\eqld@transpose@active\eqld@false}},%
5478 {\noamp,plain,restricted}{\let\eqld@transpose@active\eqld@true}},%
5479 {{\eqld@decide@true,amp,cont}}{\let\eqld@transpose@active=+}}}}
5480 \eqld@define@key{setup}{nativeequation}[true]{%
5481 \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native}
5482 \eqld@define@key{equations}{native}[true]{%
5483 \eqld@decide@bool{#3}{#2}{#1}\eqld@single@native%
5484 \ifdefined\eqld@single@native\let\eqld@layoutleft\eqld@false\fi}
5485 \eqld@define@key{setup}{scanequation}[true]{%
5486 \eqld@decide@bool{#3}{#2}{#1}\eqld@single@doscan}
5487 \eqld@define@key{equations}{scan}[true]{%
5488 \eqld@decide@bool{#3}{#2}{#1}\eqld@single@doscan}
5489 \eqld@define@key{setup}{sqropt}[]{%
5490 \def\eqld@equations@sqr@opt{equation,#1}}
5491 \eqld@define@key{setup}{angopt}[]{%
5492 \def\eqld@equations@ang@opt{columns,#1}}

```

TODO: describe

```

5493 \eqld@define@key{control}{restoreexterior}[]{\eqld@display@restore}
5494 \eqld@define@key{control}{restoreexterior*}[]{\@arrayparboxrestore}

```

Modes for Modifier Scanning.

```

5495 \def\eqld@equations@nomodifier{%
5496 \eqld@equations@test@setopt
5497 \let\eqld@equations@end@modifier\eqld@false
5498 \eqld@multi@cr@test@setopt
5499 \eqld@columns@amp@test@setopt
5500 \eqld@break@cr@test@setopt
5501 \eqld@break@amp@test@setopt
5502 }
5503 \let\eqld@equations@env@modifier\eqld@false
5504 \let\eqld@equations@sqr@modifier\eqld@true

```

```

5505 \let\eqlequations@ang@modifier\eq@true
5506 \let\eqlequations@end@modifier\eq@true
5507 \eqlequations@test@setall
5508 \eql@multi@cr@test@setall
5509 \eql@columns@amp@test@setall
5510 \eql@break@cr@test@setall
5511 \eql@break@amp@test@setall
5512 \eql@break@test@setopt
5513 \let\eql@multi@cr@relnext\eq@true

5514 \def\eql@box@nomodifier{%
5515   \eql@box@test@setopt
5516   \let\eql@box@end@modifier\eq@false
5517   \eql@box@cr@test@setopt
5518   \eql@box@amp@test@setopt
5519 }
5520 \let\eql@box@env@modifier\eq@false
5521 \let\eql@box@ang@modifier\eq@true
5522 \let\eql@box@end@modifier\eq@true
5523 \eql@box@test@setall
5524 \eql@box@cr@test@setall
5525 \eql@box@amp@test@setall

5526 \eql@define@key{setup}{modifier}[true]{%
5527   \eql@decide@if{#3}{#2}{#1}%
5528   {\let\eqlequations@env@modifier\eq@true
5529     \let\eqlequations@sqr@modifier\eq@true
5530     \let\eqlequations@ang@modifier\eq@true
5531     \let\eql@box@env@modifier\eq@true
5532     \let\eql@box@ang@modifier\eq@true
5533     \eql@multi@cr@test@setall\eql@break@cr@test@setall
5534     \eql@columns@amp@test@setall\eql@break@amp@test@setall
5535     \eql@box@cr@test@setall\eql@box@amp@test@setall
5536     \eql@break@test@setall}%
5537   {\let\eqlequations@env@modifier\eq@false
5538     \let\eqlequations@sqr@modifier\eq@false
5539     \let\eqlequations@ang@modifier\eq@false
5540     \let\eql@box@env@modifier\eq@false
5541     \let\eql@box@ang@modifier\eq@false
5542     \eql@multi@cr@test@setopt\eql@break@cr@test@setopt
5543     \eql@columns@amp@test@setopt\eql@break@amp@test@setopt
5544     \eql@box@cr@test@setopt\eql@box@amp@test@setopt
5545     \eql@break@test@setopt}}
5546 \eql@define@key{setup}{modifierenv}[true]{%
5547   \eql@decide@bool{#3}{#2}{#1}\eqlequations@env@modifier
5548   \eql@decide@bool{#3}{#2}{#1}\eql@box@env@modifier}
5549 \eql@define@key{setup}{modifiersqrang}[true]{%
5550   \eql@decide@bool{#3}{#2}{#1}\eqlequations@sqr@modifier
5551   \eql@decide@bool{#3}{#2}{#1}\eqlequations@ang@modifier
5552   \eql@decide@bool{#3}{#2}{#1}\eql@box@ang@modifier}
5553 \eql@define@key{setup}{modifierend}[true]{%
5554   \eql@decide@bool{#3}{#2}{#1}\eqlequations@end@modifier
5555   \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5556 \eql@define@key{setup}{modifierreqncr}[true]{\eql@decide@if{#3}{#2}{#1}%
5557   {\eql@multi@cr@test@setall\eql@break@cr@test@setall
5558     \eql@box@cr@test@setall}%
5559   {\eql@multi@cr@test@setopt\eql@break@cr@test@setopt
5560     \eql@box@cr@test@setopt}}
5561 \eql@define@key{setup}{modifierreqnamp}[true]{\eql@decide@if{#3}{#2}{#1}%

```

```

5562     {\eqL@columns@amp@test@setall\eqL@break@amp@test@setall
5563     \eqL@box@amp@test@setall}%
5564     {\eqL@columns@amp@test@setopt\eqL@break@amp@test@setopt
5565     \eqL@box@amp@test@setopt}}
5566 \eqL@define@key{setup}{modifierbreak}[true]{\eqL@decide@if{#3}{#2}{#1}%
5567     \eqL@break@test@setall\eqL@break@test@setopt}
5568 \eqL@define@key{setup}{modifierwarning}[all]{%
5569     \eqL@decide@select{#3}{#2}{#1}{%
5570     {\eqL@decide@false{\let\eqL@parseopt@warn@env\@empty
5571     \let\eqL@parseopt@warn@cr\@empty}},%
5572     {\env,-}{\let\eqL@parseopt@warn@env\eqL@warn@parseopt
5573     \let\eqL@parseopt@warn@cr\@empty}},%
5574     {\all,\eqL@decide@true}{\let\eqL@parseopt@warn@env\eqL@warn@parseopt
5575     \let\eqL@parseopt@warn@cr\eqL@warn@parseopt}},%
5576     {\verbose,+}{\let\eqL@parseopt@warn@env\eqL@warn@parseopt@verbose
5577     \let\eqL@parseopt@warn@cr\eqL@warn@parseopt@verbose}}}}

5578 \eqL@define@key{setup}{modifiereqn}[true]{%
5579     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@env@modifier
5580     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@sqr@modifier
5581     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@ang@modifier}
5582 \eqL@define@key{setup}{modifiereqnenv}[true]{%
5583     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@env@modifier}
5584 \eqL@define@key{setup}{modifiereqnsqr}[true]{%
5585     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@sqr@modifier}
5586 \eqL@define@key{setup}{modifiereqnang}[true]{%
5587     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@ang@modifier}
5588 \eqL@define@key{setup}{modifiereqnend}[true]{%
5589     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@end@modifier}
5590 \eqL@define@key{setup}{modifiereqncr}[true]{\eqL@decide@if{#3}{#2}{#1}%
5591     {\eqL@multi@cr@test@setall\eqL@break@cr@test@setall}%
5592     {\eqL@multi@cr@test@setopt\eqL@break@cr@test@setopt}}
5593 \eqL@define@key{setup}{modifiereqnamp}[true]{\eqL@decide@if{#3}{#2}{#1}%
5594     {\eqL@columns@amp@test@setall\eqL@break@amp@test@setall}%
5595     {\eqL@columns@amp@test@setopt\eqL@break@amp@test@setopt}}

5596 \eqL@define@key{setup}{modifierbox}[true]{%
5597     \eqL@decide@bool{#3}{#2}{#1}\eqL@box@env@modifier
5598     \eqL@decide@bool{#3}{#2}{#1}\eqL@box@ang@modifier}
5599 \eqL@define@key{setup}{modifierboxenv}[true]{%
5600     \eqL@decide@bool{#3}{#2}{#1}\eqL@box@env@modifier}
5601 \eqL@define@key{setup}{modifierboxang}[true]{%
5602     \eqL@decide@bool{#3}{#2}{#1}\eqL@box@ang@modifier}
5603 \eqL@define@key{setup}{modifierboxend}[true]{%
5604     \eqL@decide@bool{#3}{#2}{#1}\eqL@box@end@modifier}
5605 \eqL@define@key{setup}{modifierboxcr}[true]{%
5606     \eqL@decide@if{#3}{#2}{#1}\eqL@box@cr@test@setall\eqL@box@cr@test@setopt}
5607 \eqL@define@key{setup}{modifierboxamp}[true]{%
5608     \eqL@decide@if{#3}{#2}{#1}\eqL@box@amp@test@setall\eqL@box@amp@test@setopt}

5609 \eqL@define@key{equations}{modifierend}[true]{%
5610     \eqL@decide@bool{#3}{#2}{#1}\eqL@equations@end@modifier}
5611 \eqL@define@key{equations}{modifiercr}[true]{\eqL@decide@if{#3}{#2}{#1}%
5612     {\eqL@multi@cr@test@setall\eqL@break@cr@test@setall}%
5613     {\eqL@multi@cr@test@setopt\eqL@break@cr@test@setopt}}
5614 \eqL@define@key{equations}{modifieramp}[true]{\eqL@decide@if{#3}{#2}{#1}%
5615     {\eqL@columns@amp@test@setall\eqL@break@amp@test@setall}%
5616     {\eqL@columns@amp@test@setopt\eqL@break@amp@test@setopt}}
5617 \eqL@define@key{setup,equations}{crrrelnext}[true]{%

```

```

5618 \eql@decide@bool{#3}{#2}{#1}\eql@multi@cr@relnext}

5619 \eql@define@key{box}{modifier}[true]{\eql@decide@if{#3}{#2}{#1}%
5620   {\eql@box@cr@test@setall\eql@box@amp@test@setall
5621     \let\eql@box@end@modifier\eql@true}%
5622   {\eql@box@cr@test@setopt\eql@box@amp@test@setopt
5623     \let\eql@box@end@modifier\eql@false}}
5624 \eql@define@key{box}{modifierend}[true]{%
5625   \eql@decide@bool{#3}{#2}{#1}\eql@box@end@modifier}
5626 \eql@define@key{box}{modifiercr}[true]{%
5627   \eql@decide@if{#3}{#2}{#1}\eql@box@cr@test@setall\eql@box@cr@test@setopt}
5628 \eql@define@key{box}{modifieramp}[true]{%
5629   \eql@decide@if{#3}{#2}{#1}\eql@box@amp@test@setall\eql@box@amp@test@setopt}

```

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

```

5630 \eql@define@key\eql@keyall{spread}{%
5631   \let\eql@spread@reset\eql@false\def\eql@spread@val{#1}}
5632 \eql@define@key\eql@keyall{spread*}[Opt]{%
5633   \let\eql@spread@reset\eql@true\def\eql@spread@val{#1}}
5634 \eql@define@key\eql@keyall{strut}[true]{\eql@decide@select{#3}{#2}{#1}{%
5635   {\eql@decide@false{\let\eql@strut@cell\relax\let\eql@strut@tag\relax}},%
5636   {{cell}{\let\eql@strut@cell\eql@strut\let\eql@strut@tag\relax}},%
5637   {{tag}{\let\eql@strut@cell\relax\let\eql@strut@tag\eql@strut}},%
5638   {\eql@decide@true
5639     {\let\eql@strut@cell\eql@strut\let\eql@strut@tag\eql@strut}}}}
5640 \eql@define@key{setup}{strutdepth}{\def\eql@strut@depth{#1}}

```

Settings to specify the apparent height and depth of equations:

```

5641 \eql@define@key\eql@keyall{displayheight}[strut]{%
5642   \eql@decide@select{#3}{#2}{#1}{%
5643     {\eql@decide@false{\let\eql@display@height\@undefined}},%
5644     {{strut}{\def\eql@display@height{\ht\eql@strutbox0}}},%
5645     {\relax{\def\eql@display@height{#1}}}}}
5646 \eql@define@key\eql@keyall{displaydepth}[strut]{%
5647   \eql@decide@select{#3}{#2}{#1}{%
5648     {\eql@decide@false{\let\eql@display@depth\@undefined}},%
5649     {{strut}{\def\eql@display@depth{\dp\eql@strutbox0}}},%
5650     {\relax{\def\eql@display@depth{#1}}}}}

```

Settings concerning page breaks:

```

5651 \eql@define@key{equations}{prebreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5652   {{force,4,\eql@decide@true}{\eql@displaybreak@pre4}},%
5653   {{high,3}{\eql@displaybreak@pre3}},%
5654   {{med,medium,2}{\eql@displaybreak@pre2}},%
5655   {{low,1}{\eql@displaybreak@pre1}},%
5656   {{0,\eql@decide@false}{\eql@displaybreak@pre0}},%
5657   {{default,inherit,-1}{\eql@displaybreak@pre\m@ne}}}}
5658 \eql@define@key{equations}{postbreak}[4]{\eql@decide@select{#3}{#2}{#1}{%
5659   {{force,4,\eql@decide@true}{\eql@displaybreak@post4}},%
5660   {{high,3}{\eql@displaybreak@post3}},%
5661   {{med,medium,2}{\eql@displaybreak@post2}},%
5662   {{low,1}{\eql@displaybreak@post1}},%
5663   {{0,\eql@decide@false}{\eql@displaybreak@post0}},%
5664   {{default,inherit,-1}{\eql@displaybreak@post\m@ne}}}}
5665 \eql@define@key{equations,setup}{allowbreaks,allowdisplaybreaks}[4]{%
5666   \eql@decide@select{#3}{#2}{#1}{%

```



```

5667    {{full,4}}{\eqldisplaybreak@inter4}},%
5668    {{high,3}}{\eqldisplaybreak@inter3}},%
5669    {{med,medium,2}}{\eqldisplaybreak@inter2}},%
5670    {{low,1}}{\eqldisplaybreak@inter1}},%
5671    {{0,\eqldecide@false}}{\eqldisplaybreak@inter\z@}}}}
5672 \eqldefine@key{equations}{prepenalty}{%
5673   \eqldisplaybreak@prepen@\numexpr#1\relax}
5674 \eqldefine@key{equations}{postpenalty}{%
5675   \eqldisplaybreak@postpen@\numexpr#1\relax}
5676 \eqldefine@key{equations,setup}{interpenalty}{%
5677   \interdisplaylinepenalty\numexpr#1\relax}

```

TODO: describe

```

5678 \eqldefine@key{control}{vspace}[]{\eqlvspace@add{#1}}
5679 \eqldefine@key{control}{vspace*}[]{\eqlvspace@addfixedbefore{#1}}
5680 \eqldefine@key{control}{vspace!}[]{\eqlvspace@addfixedafter{#1}}
5681 \eqldefine@key{control}{break}[4]{\eqldisplaybreak@level[{#1}]}
5682 \eqldefine@key{control}{penalty}[]{\eqldisplaybreak@star{#1}}

```

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

```

5683 \eqldefine@key{equations}{noskip}[both]{%
5684   \eqldecide@abovebelow{#3}{#2}{#1}%
5685   {\def\eqlskip@force@above{5}}%
5686   {\def\eqlskip@force@below{5}}}
5687 \eqldefine@key{equations}{short}[above]{%
5688   \eqldecide@abovebelow{#3}{#2}{#1}%
5689   {\def\eqlskip@force@above{1}}%
5690   {\def\eqlskip@force@below{1}}}
5691 \eqldefine@key{equations}{long}[both]{%
5692   \eqldecide@abovebelow{#3}{#2}{#1}%
5693   {\def\eqlskip@force@above{0}}%
5694   {\def\eqlskip@force@below{0}}}
5695 \eqldefine@key{equations}{medskip}[both]{%
5696   \eqldecide@abovebelow{#3}{#2}{#1}%
5697   {\def\eqlskip@force@above{6}}%
5698   {\def\eqlskip@force@below{6}}}
5699 \eqldefine@key{equations}{par}[par]{%
5700   \eqldecide@select{#3}{#2}{#1}{%
5701     {\default,\eqldecide@false}{\let\eqlskip@force@leave\undefined}},%
5702     {\cont,hmode}{\let\eqlskip@force@leave\z@}},%
5703     {\par,vmode}{\let\eqlskip@force@leave\one
5704       \ifdefined\eqlskip@force@below\else
5705         \def\eqlskip@force@below{3}%
5706       \fi}},%
5707     {\top}{\let\eqlskip@force@leave\tw@
5708       \ifdefined\eqlskip@force@below\else
5709         \def\eqlskip@force@below{4}
5710       \fi}}}}

```

Specify vertical spacing explicitly:

```

5711 \eqldefine@key{equations}{skip}{%
5712   \def\eqlskip@force@above{7}%
5713   \def\eqlskip@custom@above{#1}%
5714   \let\eqlskip@force@below\eqlskip@force@above
5715   \let\eqlskip@custom@below\eqlskip@custom@above}
5716 \eqldefine@key{equations}{aboveskip}{%

```

```

5717 \def\eql@skip@force@above{7}%
5718 \def\eql@skip@custom@above{#1}%
5719 \eql@define@key{equations}{belowskip}{%
5720 \def\eql@skip@force@below{7}%
5721 \def\eql@skip@custom@below{#1}%
5722 \eql@define@key{equations}{abovespace}{%
5723 \advance\eql@abovespace@{glueexpr#1\relax}
5724 \eql@define@key{equations}{belowspace}{%
5725 \advance\eql@belowspace@{glueexpr#1\relax}

```

Vertical spacing for intertext:

```

5726 \eql@define@key{intertext}{skip}{%
5727 \def\eql@skip@force@above{7}%
5728 \def\eql@skip@custom@above{#1}%
5729 \let\eql@skip@force@below\eql@skip@force@above
5730 \let\eql@skip@custom@below\eql@skip@custom@above}
5731 \eql@define@key{intertext}{aboveskip}{%
5732 \def\eql@skip@force@below{7}%
5733 \def\eql@skip@custom@below{#1}%
5734 \eql@define@key{intertext}{belowskip}{%
5735 \def\eql@skip@force@above{7}%
5736 \def\eql@skip@custom@above{#1}%
5737 \eql@define@key{intertext}{noskip}[both]{%
5738 \eql@decide@abovebelow{#3}{#2}{#1}%
5739 {\def\eql@skip@force@below{5}}%
5740 {\def\eql@skip@force@above{5}}}%
5741 \eql@define@key{intertext}{short}[both]{%
5742 \eql@decide@abovebelow{#3}{#2}{#1}%
5743 {\def\eql@skip@force@below{1}}%
5744 {\def\eql@skip@force@above{1}}}%
5745 \eql@define@key{intertext}{long}[both]{%
5746 \eql@decide@abovebelow{#3}{#2}{#1}%
5747 {\def\eql@skip@force@below{0}}%
5748 {\def\eql@skip@force@above{0}}}%
5749 \eql@define@key{intertext}{medskip}[both]{%
5750 \eql@decide@abovebelow{#3}{#2}{#1}%
5751 {\def\eql@skip@force@below{6}}%
5752 {\def\eql@skip@force@above{6}}}%

```

Configure general vertical spacing behaviour for various situations:

```

5753 \eql@define@key{setup}{skip,longskip}{%
5754 \abovedisplayskip\glueexpr#1\relax
5755 \belowdisplayskip\abovedisplayskip
5756 \def\eql@skip@long@above{#1}%
5757 \let\eql@skip@long@below\eql@skip@long@above}
5758 \eql@define@key{setup}{aboveskip,abovelongskip}{%
5759 \abovedisplayskip\glueexpr#1\relax
5760 \def\eql@skip@long@above{#1}%
5761 \eql@define@key{setup}{belowskip,belowlongskip}{%
5762 \belowdisplayskip\glueexpr#1\relax
5763 \def\eql@skip@long@below{#1}%
5764 \eql@define@key{setup}{aboveskip,aboveskip}{%
5765 \abovedisplayshortskip\glueexpr#1\relax
5766 \def\eql@skip@short@above{#1}%
5767 \eql@define@key{setup}{belowshortskip}{%
5768 \belowdisplayshortskip\glueexpr#1\relax
5769 \def\eql@skip@short@below{#1}%
5770 \eql@define@key{setup}{tagsskip}{%

```

```

5771 \def\eq@skip@tag@above{#1}%
5772 \let\eq@skip@tag@below\eq@skip@tag@above}
5773 \eq@define@key{setup}{abovetagskip}{%
5774 \def\eq@skip@tag@above{#1}}
5775 \eq@define@key{setup}{belowtagskip}{%
5776 \def\eq@skip@tag@below{#1}}
5777 \eq@define@key{setup}{medskip}{%
5778 \def\eq@skip@med@above{#1}%
5779 \let\eq@skip@med@below\eq@skip@med@above}
5780 \eq@define@key{setup}{abovemedskip}{%
5781 \def\eq@skip@med@above{#1}}
5782 \eq@define@key{setup}{belowmedskip}{%
5783 \def\eq@skip@med@below{#1}}
5784 \eq@define@key{setup}{abovetopskip}{%
5785 \def\eq@skip@top@above{#1}}
5786 \eq@define@key{setup}{belowtopskip}{%
5787 \def\eq@skip@top@below{#1}}
5788 \eq@define@key{setup}{aboveparskip}{%
5789 \def\eq@skip@par@above{#1}}
5790 \eq@define@key{setup}{belowparskip}{%
5791 \def\eq@skip@par@below{#1}}
5792 \eq@define@key{setup}{abovecontskip}{%
5793 \eq@decide@select{#3}{#2}{#1}{%
5794 {{hide}}{\def\eq@skip@cont@above{\eq@spread@val-\eq@skip@long@below}}},%
5795 {\relax{\def\eq@skip@cont@above{#1}}}}
5796 \eq@define@key{setup}{belowcontskip}{%
5797 \def\eq@skip@cont@below{#1}}
5798 \eq@define@key{setup}{shortmode}{%
5799 \eq@decide@select{#3}{#2}{#1}{%
5800 {{\eq@decide@false,never}}{\def\eq@skip@mode@short{0}}},%
5801 {{above,neverbelow,belowoff}}{\def\eq@skip@mode@short{1}}},%
5802 {{belowone,belowsingle}}{\def\eq@skip@mode@short{2}}},%
5803 {{belowall,always,on}}{\def\eq@skip@mode@short{3}}}}
5804 \eq@define@key{setup}{abovecontmode}{%
5805 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@above}
5806 \eq@define@key{setup}{belowcontmode}{%
5807 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@cont@below}
5808 \eq@define@key{setup}{aboveparmode}{%
5809 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@above}
5810 \eq@define@key{setup}{belowparmode}{%
5811 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@par@below}
5812 \eq@define@key{setup}{abovetopmode}{%
5813 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@above}
5814 \eq@define@key{setup}{belowtopmode}{%
5815 \eq@decide@situation{#3}{#2}{#1}\eq@skip@mode@top@below}

```

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

```

5816 \eq@define@key{equations,subequations}{label}{\eq@tags@addblock@label{#1}}
5817 \eq@define@key{equations,subequations}{labelname}{\eq@tags@addblock@name{#1}}
5818 \eq@define@key{equations,subequations}{tag}{\eq@tags@addblock@tag{#1}}
5819 \eq@define@key{equations,subequations}{tag*}{%
5820 \eq@tags@addblock@tagform@off\eq@tags@addblock@tag{#1}}
5821 \eq@define@key{equations,subequations}{taglabel}{\eq@tags@addblock@ref{#1}}

```

TODO: describe

```

5822 \eq@define@key{control}{label}{\eq@tags@add@label{#1}}
5823 \eq@define@key{control}{labelname}{\eq@tags@add@name{#1}}

```

```

5824 \eqld@define@key{control}{tag}{\eqld@tags@add@tag{#1}}
5825 \eqld@define@key{control}{tag*}{\eqld@tags@add@tagform@off\eqld@tags@add@tag{#1}}
5826 \eqld@define@key{control}{taglabel}{\eqld@tags@add@ref{#1}}
5827 \eqld@define@key{control}{shifftag}{\eqld@tags@add@raiseshift{#1}}
5828 \eqld@define@key{control}{smashtag}{\eqld@tags@add@raisesmash{#1}}
5829 \eqld@define@key{control}{pushtag}{\eqld@tags@add@forceraise}

```

TODO: describe

```

5830 \eqld@define@key{setup}{labelname}{\protected@edef\eqld@tags@name@generic{#1}}
5831 \eqld@define@key{setup}{autolabel}[true]{%
5832   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autolabel}
5833 \eqld@define@key{setup}{autotag}[true]{%
5834   \eqld@decide@bool{#3}{#2}{#1}\eqld@tags@autotag}

```

Tag Spacing. Configure horizontal spacing for equation tags:

```

5835 \eqld@define@key{equations,setup}{tagmargin}[auto]{%
5836   \eqld@decide@select{#3}{#2}{#1}{%
5837     {auto,\eqld@decide@false}{\let\eqld@tagmargin@val\undefined}},%
5838     {\relax{\def\eqld@tagmargin@val{#1}}}}}%
5839 \eqld@define@key{equations,setup}{tagmargin*}{%
5840   \settowidth\dimen@{#1}\edef\eqld@tagmargin@val{\the\dimen@}}
5841 \eqld@define@key{equations,setup}{tagmarginratio}{%
5842   \eqld@tagmargin@ratio@dimexpr#1pt\relax}
5843 \eqld@define@key{equations,setup}{tagmarginthreshold}{%
5844   \def\eqld@tagmargin@threshold{#1}}
5845 \eqld@define@key{equations,setup}{mintagsep}{\def\eqld@tagsepmin@val{#1}}
5846 \eqld@define@key{equations,setup}{mintagwidth}{%
5847   \settowidth\dimen@{#1}\edef\eqld@tagsepmin@val{\the\dimen@}}
5848 \eqld@define@key{equations,setup}{mintagwidth*}{%
5849   \settowidth\eqld@tagwidthmin@{#1}}
5850 \eqld@define@key{equations,setup}{tagsnap}{%
5851   \eqld@decide@select{#3}{#2}{#1}{%
5852     {\eqld@decide@false{\let\eqld@tagpos@snap\z@}},%
5853     {\relax{\def\eqld@tagpos@snap{#1}}}}}%

```

Tag Layout. Configure methods to declare equation tag layout:

```

5854 \eqld@define@key{equations,setup}{tagbox,taglayout}{%
5855   \eqld@tags@taglayout@set{#1}}
5856 \eqld@define@key{equations,setup}{tagbox*,taglayout*}{%
5857   \eqld@tags@taglayout@set@direct{#1}}
5858 \eqld@define@key{equations,setup}{tagform}{%
5859   \eqld@tags@tagform@set{#1}}
5860 \eqld@define@key{equations,setup}{tagform*}{%
5861   \eqld@tags@tagform@set@direct{#1}}
5862 \eqld@define@key{equations,setup}{subeqtemplate}{%
5863   \def\eqld@subequations@template####1####2{#1}%
5864   \eqld@append\eqld@subequations@template{\theparentequation{equation}}}}
5865 \eqld@define@key{control}{tagbox,taglayout}{%
5866   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set{#1}}}
5867 \eqld@define@key{control}{tagbox*,taglayout*}{%
5868   \global\eqld@append\eqld@tags@container{\eqld@tags@taglayout@set@direct{#1}}}
5869 \eqld@define@key{control}{tagform}{%
5870   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set{#1}}}
5871 \eqld@define@key{control}{tagform*}{\####1}{%
5872   \global\eqld@append\eqld@tags@container{\eqld@tags@tagform@set@direct{#1}}}

```

Equation Numbering. Configure equation numbering schemes:

```

5873 \eqld@define@key{equations,setup}{numberline,number,num,numline,n}[all]{%
5874   \eqld@decide@select{#3}{#2}{#1}{%
5875     {\eqld@decide@false,0,*}{\let\eqld@numbering@active\eqld@false}},%
5876     {\eqld@decide@true,!}{\let\eqld@numbering@active\eqld@true}},%
5877     {none,n,-}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5878       \let\eqld@numbering@active\eqld@false}},%
5879     {single,1}{\let\eqld@numbering@mode\eqld@numbering@mode@single
5880       \let\eqld@numbering@active\eqld@true}},%
5881     {multi,@}{\let\eqld@numbering@mode\eqld@numbering@mode@multi
5882       \let\eqld@numbering@active\eqld@true}},%
5883     {\relax{\eqld@numbering@set{#1}}}}
5884 \eqld@define@key{equations,setup}{nonumber,nn,*}[]{%
5885   \let\eqld@numbering@active\eqld@false}
5886 \eqld@define@key{equations,setup}{donumber,dn,!}[]{%
5887   \let\eqld@numbering@active\eqld@true}
5888 \eqld@define@key{equations,setup}{tagsleft,leqno}[]{%
5889   \let\eqld@tagsleft\eqld@true}
5890 \eqld@define@key{equations,setup}{tagsright,reqno}[]{%
5891   \let\eqld@tagsleft\eqld@false}
5892 \eqld@define@key{equations,setup}{tags,eqno}{%
5893   \eqld@decide@select{#3}{#2}{#1}{%
5894     {right,r}{\let\eqld@tagsleft\eqld@false}},%
5895     {left,l}{\let\eqld@tagsleft\eqld@true}}}}
5896 \eqld@define@key{equations,setup}{evadetag,avoidtag}[true]{%
5897   \eqld@decide@bool{#3}{#2}{#1}\eqld@numbering@best@auto}
5898 \eqld@define@key{equations,setup}{tagbetween}[true]{%
5899   \eqld@decide@bool{#3}{#2}{#1}\eqld@tagpos@doconvert}

```

TODO: describe

```

5900 \eqld@define@key{control}{nonumber,nn,*}[]{\global\eqnswfalse}
5901 \eqld@define@key{control}{donumber,dn,!}[]{\global\eqnswtrue}
5902 \eqld@define@key{control}{numberhere}[]{\eqld@numberhere}
5903 \eqld@define@key{control}{numbernext}[]{\eqld@numbernext}

```

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

```

5904 \eqld@define@key{equations,setup}{layout}{\eqld@decide@select{#3}{#2}{#1}{%
5905   {\center,c}{\let\eqld@layoutleft\eqld@false}},%
5906   {\left,l}{\let\eqld@layoutleft\eqld@true}}}}
5907 \eqld@define@key{equations,setup}{center}[]{%
5908   \let\eqld@layoutleft\eqld@false}
5909 \eqld@define@key{equations,setup}{flushleft,left}[]{%
5910   \let\eqld@layoutleft\eqld@true}
5911 \eqld@define@key{equations,setup}{leftmargin}{\def\eqld@layoutleftmargin{#1}}
5912 \eqld@define@key{equations,setup}{leftmargin*}{%
5913   \settowidth\dimen{#1}\edef\eqld@layoutleftmargin{\the\dimen}}
5914 \eqld@define@key{equations,setup}{minleftmargin}{%
5915   \def\eqld@layoutleftmarginmin{#1}}
5916 \eqld@define@key{equations,setup}{maxleftmargin}{%
5917   \eqld@decide@select{#3}{#2}{#1}{%
5918     {\eqld@decide@false{\def\eqld@layoutleftmarginmax{.5\maxdimen}}},%
5919     {\relax{\def\eqld@layoutleftmarginmax{#1}}}}}
5920 \eqld@define@key{equations,box}{margin}{%
5921   \def\eqld@display@marginleft{#1}\def\eqld@display@marginright{#1}}

```

```

5922 \eqld@define@key{equations,box}{marginleft}{\def\eqld@display@marginleft{#1}}
5923 \eqld@define@key{equations,box}{marginright}{\def\eqld@display@marginright{#1}}
5924 \eqld@define@key{equations}{linewidth,width}{\def\eqld@display@linewidth{#1}}

```

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

```

5925 \eqld@define@key{equations,setup}{alignshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5926     {{max,full,4}{\eqld@alignbadness@inf@bad}},%
5927     {{high,3}{\eqld@alignbadness@54\relax}},%
5928     {{med,medium,2}{\eqld@alignbadness@18\relax}},%
5929     {{low,1}{\eqld@alignbadness@6\relax}},%
5930     {{0,\eqld@decide@false}{\eqld@alignbadness@z@}}}}
5931 \eqld@define@key{equations,setup}{tagshrink}{\eqld@decide@select{#3}{#2}{#1}{%
5932     {{max,full,4}{\eqld@tagbadness@inf@bad}},%
5933     {{high,3}{\eqld@tagbadness@54\relax}},%
5934     {{med,medium,2}{\eqld@tagbadness@18\relax}},%
5935     {{low,1}{\eqld@tagbadness@6\relax}},%
5936     {{0,\eqld@decide@false}{\eqld@tagbadness@z@}}}}
5937 \eqld@define@key{equations,setup}{alignbadness}{%
5938     \eqld@alignbadness@numexpr#1\relax}
5939 \eqld@define@key{equations,setup}{tagbadness}{%
5940     \eqld@tagbadness@numexpr#1\relax}
5941 \eqld@define@key{equations,setup}{mincolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5942     {{0,\eqld@decide@false}{\def\eqld@colsepmin@val{0pt}}},%
5943     {\relax{\def\eqld@colsepmin@val{#1}}}}}
5944 \eqld@define@key{equations,setup}{maxcolsep}{\eqld@decide@select{#3}{#2}{#1}{%
5945     {\eqld@decide@false{\def\eqld@colsepmax@val{.5\maxdimen}}},%
5946     {\relax{\def\eqld@colsepmax@val{#1}}}}}
5947 \eqld@define@key{equations,setup}{fulllength}[true]{%
5948     \eqld@decide@bool{#3}{#2}{#1}\eqld@columns@fulllength}

```

TODO: is boxcolsep vs breakcolsep okay??!

```

5949 \eqld@define@key{equations,setup}{linesep}{\eqld@decide@select{#3}{#2}{#1}{%
5950     {{0,\eqld@decide@false}{\def\eqld@break@line@sep{0pt}}},%
5951     {\relax{\def\eqld@break@line@sep{#1}}}}}
5952 \eqld@define@key{equations,setup}{linesep*}{\eqld@decide@select{#3}{#2}{#1}{%
5953     {{0,\eqld@decide@false}{\def\eqld@break@line@shortsep{0pt}}},%
5954     {\relax{\def\eqld@break@line@shortsep{#1}}}}}
5955 \eqld@define@key{box,setup}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5956     {{0,\eqld@decide@false}{\def\eqld@box@colsep{0pt}}},%
5957     {{short}{\def\eqld@box@colsep{\eqld@box@shortsep}}},%
5958     {\relax{\def\eqld@box@colsep{#1}}}}}
5959 \let\eqld@break@col@sep\eqld@box@colsep
5960 \eqld@define@key{equations}{colsep}{\eqld@decide@select{#3}{#2}{#1}{%
5961     {{0,\eqld@decide@false}{\def\eqld@break@col@sep{0pt}}},%
5962     {\relax{\def\eqld@break@col@sep{#1}}}}}
5963 \let\eqld@colsepmin@val\eqld@box@colsep
5964 \let\eqld@colsepmax@val\eqld@box@colsep
5965 \let\eqld@box@colsep\eqld@break@col@sep
5966 \eqld@define@key{equations,setup}{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5967     {{0,\eqld@decide@false}{\def\eqld@break@col@shortsep{0pt}}},%
5968     {\relax{\def\eqld@break@col@shortsep{#1}}}}}
5969 \eqld@define@key{box,setup}{colsep*}{\eqld@decide@select{#3}{#2}{#1}{%
5970     {{0,\eqld@decide@false}{\def\eqld@box@shortsep{0pt}}},%
5971     {\relax{\def\eqld@box@shortsep{#1}}}}}
5972 \eqld@define@key{box,setup}{condsep}{\eqld@decide@select{#3}{#2}{#1}{%
5973     {{0,\eqld@decide@false}{\def\eqld@box@condsep{0pt}}},%

```



```
5974 \relax{\def\eql@box@condsep{#1}}}
```

Horizontal Shape. Configure horizontal alignment schemes:

```
5975 \eql@define@key\eql@keyall{shape}[default]{\eql@shape@set{#1}}
5976 \eql@define@key\eql@keyall{padding,pad}[indent]{%
5977 \eql@decide@select{#3}{#2}{#1}{%
5978   {{max}}{\let\eql@paddingleft@val\@undefined}},%
5979   {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5980   {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5981   {\relax{\def\eql@paddingleft@val{#1}}}}%
5982 \let\eql@paddingright@val\eql@paddingleft@val}
5983 \eql@define@key\eql@keyall{padleft}[indent]{%
5984 \eql@decide@select{#3}{#2}{#1}{%
5985   {{max}}{\let\eql@paddingleft@val\@undefined}},%
5986   {{indent}}{\def\eql@paddingleft@val{\eql@indent@val}}},%
5987   {{0,\eql@decide@false}}{\def\eql@paddingleft@val{0pt}}},%
5988   {\relax{\def\eql@paddingleft@val{#1}}}}%
5989 \eql@define@key\eql@keyall{padright}[indent]{%
5990 \eql@decide@select{#3}{#2}{#1}{%
5991   {{max}}{\let\eql@paddingright@val\@undefined}},%
5992   {{indent}}{\def\eql@paddingright@val{\eql@indent@val}}},%
5993   {{0,\eql@decide@false}}{\def\eql@paddingright@val{0pt}}},%
5994   {\relax{\def\eql@paddingright@val{#1}}}}%
5995 \eql@define@key\eql@keyall{indent}[2em]{%
5996 \def\eql@indent@val{#1}}
```

TODO: describe

```
5997 \def\eql@shape@rel{\eqncontrol{align=left}}
5998 \def\eql@shape@cont{\eqncontrol{align=left,shift=*}}
5999 \eql@define@key\eql@keyall{shaperel}[]{\def\eql@shape@rel{#1}}
6000 \eql@define@key\eql@keyall{shapecont}[]{\def\eql@shape@cont{#1}}
```

TODO: describe

```
6001 \eql@define@key{control}{align}[]{%
6002 \eql@decide@select{#3}{#2}{#1}{%
6003   {{l,left}}{\global\eql@append\eql@cell@container{\eql@shape@pos@z@}}},%
6004   {{c,center}}{\global\eql@append\eql@cell@container{\eql@shape@pos@one}}},%
6005   {{r,right}}{\global\eql@append\eql@cell@container{\eql@shape@pos@two}}}}%
6006 \eql@define@key{control}{shift,shiftto}[]{%
6007 \eql@decide@select{#3}{#2}{#1}{%
6008   {{*,indent}}{\eql@shape@alignamount@set{\eql@indent@}}},%
6009   {{!,outdent}}{\eql@shape@alignamount@set{-\eql@indent@}}},%
6010   {\relax{\eql@shape@alignamount@set{#1}}}}%
6011 \eql@define@key{control}{shift*,shiftby}[]{\eql@shape@alignamount@add{#1}}
```

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

```
6012 \eql@define@key\eql@keyall{classout}{\eql@class@innerleft@set{#1}}
6013 \eql@define@key\eql@keyall{classin}{\eql@class@innerright@set{#1}}
6014 \eql@define@key\eql@keyall{classlead,classin*}{\eql@class@innerlead@set{#1}}
6015 \eql@define@key\eql@keyall{rel}{\def\eql@class@rel@symb{#1}}
6016 \eql@define@key\eql@keyall{classcont}{\def\eql@class@rel@cont##1{#1}}
6017 \eql@define@key\eql@keyall{classstart}{\def\eql@class@rel@start##1{#1}}
6018 \eql@define@key{control}{rel}[\eql@class@rel@symb]{\eql@class@rel@make{#1}}
6019 \eql@define@key{control}{rel;}[]{\eql@class@rel@make{}}
6020 \eql@define@key{control}{rel*}[]{\eql@class@rel@make{}}
```

```

6021 \eqld@define@key\eqld@keyall{ampeq}[]{\eqld@class@ampeq}
6022 \eqld@define@key\eqld@keyall{eqamp}[]{\eqld@class@eqamp}
6023 \eqld@define@key\eqld@keyall{class}{\eqld@decide@select{#3}{#2}{#1}{%
6024   {{ampeq,amprel,eqafter,beforerel}\eqld@class@ampeq},%
6025   {{eqamp,relamp,eqbefore,afterrel}\eqld@class@eqamp}}}%

```

Math Styles. Configure math classes at alignment marker:

```

6026 \eqld@define@key\eqld@keyall{style}[display]{%
6027   \eqld@decide@select{#3}{#2}{#1}{%
6028     {{text}{\let\eqld@mathstyle\@empty}},%
6029     {{display}{\let\eqld@mathstyle\displaystyle}}}%
6030 \eqld@define@key{setup}{casesstyle}[display]{%
6031   \eqld@decide@select{#3}{#2}{#1}{%
6032     {\eqld@decide@false{\let\eqld@cases@mathstyle\eqld@false}},%
6033     {{text}{\let\eqld@cases@mathstyle\@empty}},%
6034     {{display}{\let\eqld@cases@mathstyle\displaystyle}}}%

```

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

```

6035 \def\eqld@punct@all#1#2#3#4#5\eqld@punct@end{%
6036   \def\eqld@tmp{#4}\def\eqld@tmpa{1}%
6037   \ifx\eqld@tmp\eqld@tmpa
6038     \ifnum#5=1111\relax
6039       \eqld@punct@set\eqld@punct@col{#1}%
6040       \eqld@punct@set\eqld@punct@line{#2}%
6041       \eqld@punct@set\eqld@punct@block{#3}%
6042     \else\ifnum#5=111\relax
6043       \eqld@punct@set\eqld@punct@line{#1}%
6044       \eqld@punct@set\eqld@punct@block{#2}%
6045     \else\ifnum#5=11\relax
6046       \eqld@punct@set\eqld@punct@block{#1}%
6047     \else
6048       \eqld@punct@clear
6049     \fi\fi\fi
6050   \else
6051     \eqld@error{Too many arguments to punctall}%
6052   \fi
6053 }

```

TODO: describe

```

6054 \eqld@define@key\eqld@keyall{punctsep}[\,]{\def\eqld@punct@sep{#1}}
6055 \eqld@define@key\eqld@keyall{punctclass}[\mathclose{}]{\def\eqld@punct@class{#1}}
6056 \eqld@define@key\eqld@keyall{punct}[\,]{\eqld@punct@set\eqld@punct@block{#1}}
6057 \eqld@define@key\eqld@keyall{punct*}[\,]{\eqld@punct@set\eqld@punct@block\relax}
6058 \eqld@define@key\eqld@keyall{punctline}[\,]{\eqld@punct@set\eqld@punct@line{#1}}
6059 \eqld@define@key\eqld@keyall{punctline*}[\,]{\eqld@punct@set\eqld@punct@line\relax}
6060 \eqld@define@key\eqld@keyall{punctcol}[\,]{\eqld@punct@set\eqld@punct@col{#1}}
6061 \eqld@define@key\eqld@keyall{punctcol*}[\,]{\eqld@punct@set\eqld@punct@col\relax}
6062 \eqld@define@key\eqld@keyall{punctall}[\,]{\eqld@punct@all#111111\eqld@punct@end}
6063 \eqld@define@key{box}{punctterm}[true]{%
6064   \eqld@decide@bool{#3}{#2}{#1}\eqld@box@punct@term}

6065 \eqld@define@key{control}{punctsep}[\,]{\def\eqld@punct@sep{#1}}
6066 \eqld@define@key{control}{setpunct}[\,]{\eqld@punct@set\eqld@punct@next{#1}}
6067 \eqld@define@key{control}{setpunct}[\,]{\eqld@punct@set\eqld@punct@next{#1}}
6068 \eqld@define@key{control}{setpunct*}[\,]{\let\eqld@punct@next\relax}
6069 \eqld@define@key{control}{punct,punctapply}[\relax]{%

```



```

6070 \eql@punct@set\eql@punct@next{#1}\eql@punct@apply@top}
6071 \eql@define@key{control}{punctline}[]{\eql@punct@print@line}
6072 \eql@define@key{control}{punctcol}[]{\eql@punct@print@col}

```

Frames. **TODO:** describe

```

6073 \eql@define@key{box}{frame}[\fbox]{%
6074 \def\eql@box@frame{#1}%
6075 \ifx\eql@box@frame\empty\let\eql@box@frame\@firstofone\fi}
6076 \eql@define@key{box}{wrap}{\eql@box@wrap#1}
6077 \eql@define@key{box}{delim}[r]{\eql@decide@delim{#3}{#2}{#1}}
6078 \eql@define@key{box}{ldelim}{\eql@box@ldelim#1}
6079 \eql@define@key{box}{rdelim}{\eql@box@rdelim#1}
6080 \eql@define@key{box}{lbrace}[]{\eql@box@ldelim\lbrace}
6081 \eql@define@key{box}{rbrace}[]{\eql@box@rdelim\rbrace}
6082 \eql@define@key{box}{lrbace,lrbaces}[]{\eql@box@delim\lbrace\rbrace}
6083 \eql@define@key{box}{braces}[lr]{%
6084 \eql@decide@select{#3}{#2}{#1}{%
6085 {\eql@decide@false}{\eql@box@wrap}{}}},%
6086 {\l,left}{\eql@box@ldelim\lbrace}},%
6087 {\r,right}{\eql@box@rdelim\rbrace}},%
6088 {\eql@decide@true,lr,both}{\eql@box@delim\lbrace\rbrace}}}}

```

TODO: describe

```

6089 \eql@define@key{control}{framecell}[\fbox]{%
6090 \global\eql@append\eql@cell@container{\def\eql@frame@cmd{#1}}}
6091 \eql@define@key{control}{frametag}[\fbox]{%
6092 \global\eql@append\eql@tags@container{\def\eql@tags@frame@cmd{#1}}}

```

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

```

6093 \eql@define@key{equations,box}{alt}{}

```

Injectons.

```

6094 \eql@define@key{control}{inject}{%
6095 \global\eql@append\eql@interline@container{%
6096 \eql@append\eql@display@injectbefore{#1}}}
6097 \eql@define@key{control}{inject*}{%
6098 \global\eql@append\eql@interline@container{%
6099 \eql@append\eql@display@injectafter{#1}}}
6100 \eql@define@key{control}{markline}[]{\eql@markline@inject{#1}}
6101 \eql@define@key{control}{markline*}[]{\eql@markline@inject{push,#1}}
6102 \eql@define@key{control}{qed}[]{\eql@markline@inject{qed,#1}}
6103 \eql@define@key{control}{qed*}[]{\eql@markline@inject{qed,push,#1}}

```

TODO: describe

```

6104 \eql@define@key{markline}{pos}{%
6105 \eql@decide@select{#3}{#2}{#1}{%
6106 {\below,push}{\let\eql@markline@pos\eql@markline@pos@below}},%
6107 {\baseline}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
6108 {\bottom}{\let\eql@markline@pos\eql@markline@pos@bottom}}}}
6109 \eql@define@key{markline}{below,push}[]{%
6110 \let\eql@markline@pos\eql@markline@pos@below}
6111 \eql@define@key{markline}{baseline}[]{%

```

```

6112 \let\eql@markline@pos\eql@markline@pos@baseline}
6113 \eql@define@key{markline}{bottom}[]{%
6114 \let\eql@markline@pos\eql@markline@pos@bottom}
6115 \eql@define@key{markline}{shift}{\def\eql@markline@shift{#1}}
6116 \eql@define@key{markline}{symbol}{\def\eql@markline@symbol{#1}}
6117 \eql@define@key{markline}{qed}[]{\let\eql@markline@symbol\eql@markline@qed}
6118 \eql@define@key{setup}{marksymbol}{\def\eql@markline@symbol{#1}}
6119 \eql@define@key{setup}{qedsymbol}{\def\eql@markline@qed{#1}}
6120 \eql@define@key{setup}{markpos}{%
6121 \eql@decide@select{#3}{#2}{#1}{%
6122 {{below}}{\let\eql@markline@pos\eql@markline@pos@below}},%
6123 {{baseline}}{\let\eql@markline@pos\eql@markline@pos@baseline}},%
6124 {{bottom}}{\let\eql@markline@pos\eql@markline@pos@bottom}}}%

```

Global Switches. Set global switches:

```

6125 \let\eql@multi@linesfallback\eql@false
6126 \let\eql@scan@par\eql@false
6127 \let\eql@single@cr@mode\eql@false
6128 \let\eql@amp@mode\eql@true
6129 \let\eql@ampproof@active\eql@false
6130 \let\eql@parseopt@warn@env\eql@warn@parseopt
6131 \let\eql@parseopt@warn@cr\@empty

6132 \eql@define@key{equations,setup}{linesfallback}[true]{%
6133 \eql@decide@select{#3}{#2}{#1}{%
6134 {{\eql@decide@false{\let\eql@multi@linesfallback\eql@false}}},%
6135 {{reuse,lean}}{\let\eql@multi@linesfallback\z@}},%
6136 {{measure,full,\eql@decide@true}}{\let\eql@multi@linesfallback\eql@true}}}%
6137 \eql@define@key{setup}{ampproof}[true]{%
6138 \eql@decide@bool{#3}{#2}{#1}\eql@ampproof@active}
6139 \eql@define@key{equations,setup}{equationcr}[true]{%
6140 \eql@decide@select{#3}{#2}{#1}{%
6141 {{\eql@decide@false{\let\eql@single@cr@mode\eql@false}}},%
6142 {{\eql@decide@true,break}}{\let\eql@single@cr@mode\eql@break@cr}}},%
6143 {{error,verbose}}{\let\eql@single@cr@mode\eql@single@cr@error}}}%
6144 \eql@define@key\eql@keyall{amp}[true]{%
6145 \eql@decide@bool{#3}{#2}{#1}{\eql@amp@mode}}
6146 \eql@define@key\eql@keyall{rescan}[true]{%
6147 \eql@decide@if{#3}{#2}{#1}%
6148 {{\let\eql@scan@body\eql@scan@body@rescan}}%
6149 {{\let\eql@scan@body\eql@scan@body@dump}}}
6150 \eql@define@key\eql@keyall{scanpar}[true]{%
6151 \eql@decide@bool{#3}{#2}{#1}\eql@scan@par}
6152 \eql@define@key{setup}{defaults}{%
6153 \eql@decide@select{#3}{#2}{#1}{%
6154 {{classic}}{\eql@defaults@classic}},%
6155 {{eqnlines}}{\eql@defaults@eqnlines}}}%
6156 \eql@define@key\eql@keyall{verbose}[true]{%
6157 \eql@decide@if{#3}{#2}{#1}\eql@verbose@on\eql@verbose@off}

```

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

```

6158 \let\eql@provide@opt@env\tw@
6159 \let\eql@provide@opt@amsmathpatch\eql@false
6160 \let\eql@provide@opt@backup\eql@false
6161 \let\eql@provide@opt@ang\eql@true
6162 \let\eql@provide@opt@eqref\eql@true

```

```

6163 \let\eql@provide@opt@matrix\eql@true

6164 \eql@define@key{setup}{amsmathends,amsmathpatch}[true]{%
6165   \eql@error@packageoption{#2}%
6166   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@amsmathpatch}
6167 \eql@define@key{setup}{backup}[true]{%
6168   \eql@error@packageoption{#2}%
6169   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@backup}
6170 \eql@define@key{setup}{env}[equation]{%
6171   \eql@error@packageoption{#2}%
6172   \eql@decide@select{#3}{#2}{#1}{%
6173     {none,\eql@decide@false}{\let\eql@provide@opt@env\z@}},%
6174     {equation,latex}{\let\eql@provide@opt@env\ne}},%
6175     {amsmath,all,\eql@decide@true}{\let\eql@provide@opt@env\tw@}}}%
6176 \eql@define@key{setup}{ang}[true]{%
6177   \eql@error@packageoption{#2}%
6178   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@ang}
6179 \eql@define@key{setup}{eqref}[true]{%
6180   \eql@error@packageoption{#2}%
6181   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@eqref}
6182 \eql@define@key{setup}{matrix}[true]{%
6183   \eql@error@packageoption{#2}%
6184   \eql@decide@bool{#3}{#2}{#1}\eql@provide@opt@matrix}

```

Shortcut Options. **TODO:** describe

```

6185 \def\eql@parseopt@nonumber#1{\eqnaddopt{nonumber}\eql@parseopt@peek}
6186 \def\eql@parseopt@donumber#1{\eqnaddopt{donumber}\eql@parseopt@peek}
6187 \def\eql@parseopt@single#1{\eqnaddopt{single}\eql@parseopt@peek}
6188 \def\eql@parseopt@lines#1{\eqnaddopt{lines}\eql@parseopt@peek}
6189 \def\eql@parseopt@eqamp#1{\eqnaddopt{eqamp}\eql@parseopt@peek}
6190 \def\eql@parseopt@ampeq#1{\eqnaddopt{ampeq}\eql@parseopt@peek}
6191 \def\eql@parseopt@columns#1{\eqnaddopt{columns}\eql@parseopt@peek}
6192 \def\eql@parseopt@transpose#1{\eqnaddopt{columns,transpose}\eql@parseopt@peek}
6193 \def\eql@parseopt@opt[#1]{\eqnaddopt{#1}\eql@parseopt@peek}
6194 \def\eql@parseopt@label#1#2{\eqnaddopt{label={#2}}\eql@parseopt@peek}
6195 \def\eql@parseopt@punctpass{\eql@parseopt@peek'}
6196 \def\eql@parseopt@punctclear#1{\eql@parseopt@peek'\sim}
6197 \def\eql@parseopt@punctopt#1#2{\eqnaddopt{punctall={#2}}\eql@parseopt@peek}
6198 \def\eql@parseopt@punctnext#1#2{%
6199   \eql@punct@set\eql@punct@next{#2}\eql@parseopt@peek}
6200 \def\eql@parseopt@punctblock#1#2{%
6201   \eql@punct@set\eql@punct@block{#2}\eql@parseopt@peek}
6202 \def\eql@parseopt@punctterm#1{\let\eql@punct@term\eql@true\eql@parseopt@peek}
6203 \def\eql@parseopt@relsyms#1{\eql@parseopt@peek?\eql@class@rel@symb}
6204 \def\eql@parseopt@relcont#1{\eql@parseopt@peek?{}}
6205 \def\eql@parseopt@relstart#1{\eql@parseopt@peek?\relax}
6206 \def\eql@parseopt@relord#1{\eql@parseopt@peek?{}}
6207 \def\eql@parseopt@vspace[#1]{%
6208   \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.

`eqn@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:

```

6209 \def\eqn@defaults@classic{%
6210   \eqnlineset{numberline=all}%
6211   \eqnlineset{mintagsep={.5\fontdimen6\textfont2}}%
6212   \eqnlineset{maxcolsep=off}%
6213   \eqnlineset{spread={\jot}}%
6214   \eqnlineset{tagmargin}%
6215   \eqnlineset{tagmarginratio=1}%
6216   \eqnlineset{tagmarginthreshold=0.5}%
6217   \eqnlineset{leftmargin={\leftmargini}}%
6218   \eqnlineset{padding=max}%
6219   \eqnlineset{evadetag=off}%
6220   \eqnlineset{displayheight=off}%
6221   \eqnlineset{displaydepth=off}%
6222   \eqnlineset{shortmode=belowsingle}%
6223   \eqnlineset{abovecontmode=short}%
6224   \eqnlineset{belowcontmode=short}%
6225   \eqnlineset{aboveparmode=long}%
6226   \eqnlineset{belowparmode=long}%
6227   \eqnlineset{abovetopmode=long}%
6228   \eqnlineset{belowtopmode=long}%
6229   \eqnlineset{abovelongskip={\abovedisplayskip}}%
6230   \eqnlineset{belowlongskip={\belowdisplayskip}}%
6231   \eqnlineset{aboveshortskip={\abovedisplayshortskip}}%
6232   \eqnlineset{belowshortskip={\belowdisplayshortskip}}%
6233   \eqnlineset{abovemedskip={.5\abovedisplayskip}}%
6234   \eqnlineset{belowmedskip={.5\belowdisplayskip}}%
6235   \eqnlineset{abovecontskip=0pt}%
6236   \eqnlineset{belowcontskip=0pt}%
6237   \eqnlineset{aboveparskip=0pt}%
6238   \eqnlineset{belowparskip=0pt}%
6239   \eqnlineset{abovetopskip=0pt}%
6240   \eqnlineset{belowtopskip=0pt}%
6241   \eqnlineset{abovetagskip=0pt}%
6242   \eqnlineset{belowtagskip=0pt}%
6243   \eqnlineset{allowbreaks=0}%
6244   \eqnlineset{equationcr=off}%
6245   \eqnlineset{amp=off}%
6246   \eqnlineset{modifier=off}%
6247   \eqnlineset{linesfallback=false}%
6248   \eqnlineset{casesstyle=text}%
6249   \eqnlineset{sqropt=nonumber}%
6250   \eqnlineset{angopt=nonumber}%
6251 }
```

values based on 10pt vs 12pt

`eqn@defaults@eqnlines` The (default) preset `eqnlines` implements a layout that scales with the font size by using the units `em` and `\normalbaselineskip` for horizontal and vertical spacing, respectively. It aims to approximately reproduce the `classic` spacing for a 12 pt computer modern font such that 10 pt fonts will lead to slightly reduced spacing. Apart from that, the `eqnlines` setting makes some deliberate layout choices that deviate significantly from `classic` (maximum column separation, no shortening below equations):

```

6252 \def\eqn@defaults@eqnlines{%
6253   \eqnlineset{numberline=all}%
6254   \eqnlineset{mintagsep=.5em}%

```

```

6255 \eqnlineset{maxcolsep=2em}%
6256 \eqnlineset{spread={0.2\normalbaselineskip}}%
6257 \eqnlineset{tagmargin}%
6258 \eqnlineset{tagmarginratio=.334}%
6259 \eqnlineset{tagmarginthreshold=0.5}%
6260 \eqnlineset{leftmargin={\leftmargini}}%
6261 \eqnlineset{padding=0pt}%
6262 \eqnlineset{evadetag}%
6263 \eqnlineset{displayheight=strut}%
6264 \eqnlineset{displaydepth=strut}%
6265 \eqnlineset{shortmode=above}%
6266 \eqnlineset{abovecontmode=noskip}%
6267 \eqnlineset{belowcontmode=long}%
6268 \eqnlineset{aboveparmode=long}%
6269 \eqnlineset{belowparmode=long}%
6270 \eqnlineset{abovetopmode=noskip}%
6271 \eqnlineset{belowtopmode=long}%
6272 \eqnlineset{longskip={0.75\normalbaselineskip
6273   plus 0.25\normalbaselineskip minus 0.4\normalbaselineskip}}%
6274 \eqnlineset{aboveshortskip={0.0\normalbaselineskip
6275   plus 0.25\normalbaselineskip}}%
6276 \eqnlineset{belowshortskip={0.0\normalbaselineskip
6277   plus 0.25\normalbaselineskip}}%
6278 \eqnlineset{medskip={0.4\normalbaselineskip
6279   plus 0.2\normalbaselineskip minus 0.2\normalbaselineskip}}%
6280 \eqnlineset{abovecontskip=0pt}%
6281 \eqnlineset{belowcontskip=0pt}%
6282 \eqnlineset{aboveparskip=0pt}%
6283 \eqnlineset{belowparskip=0pt}%
6284 \eqnlineset{abovetopskip=0pt}%
6285 \eqnlineset{belowtopskip=0pt}%
6286 \eqnlineset{abovetagskip={0.25\normalbaselineskip
6287   minus 0.25\normalbaselineskip}}%
6288 \eqnlineset{belowtagskip={0.25\normalbaselineskip
6289   minus 0.25\normalbaselineskip}}%
6290 \eqnlineset{allowbreaks=3}%
6291 \eqnlineset{equationcr=break}%
6292 \eqnlineset{amp=on}%
6293 \eqnlineset{modifier=on,modifierenv=off}%
6294 \eqnlineset{linesfallback=true}%
6295 \eqnlineset{casesstyle=false}%
6296 \eqnlineset{sqropt}%
6297 \eqnlineset{angopt}%
6298 }

```

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.

```

\eq@provide@movecmd We introduce a couple of tools to rename and undefine commands and environments:
\eq@provide@moveenv
@provide@undefinecmd
@provide@undefineenv
6299 \def\eq@provide@movecmd#1#2{%
6300   \eq@letcs{#1\expandafter}\cname#2\endcsname
6301 }

```

```

6302 \def\eql@provide@moveenv#1#2{%
6303   \eql@provide@movecmd{#1}{#2}%
6304   \ifcsname end#2\endcsname
6305     \eql@provide@movecmd{end#1}{end#2}%
6306   \fi
6307 }
6308 \def\eql@provide@undefinecmd#1{%
6309   \eql@letcs{#1}\@undefined
6310 }
6311 \def\eql@provide@undefineenv#1{%
6312   \eql@provide@undefinecmd{#1}%
6313   \eql@provide@undefinecmd{end#1}%
6314 }

```

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:

```

6315 \def\eql@amsmath@endfix#1#2{%
6316   \long\edef\eql@tmpa{\expandafter\noexpand\csname end#2\endcsname}%
6317   \expandafter\ifx\csname end#1\endcsname\eql@tmpa
6318     \eql@provide@movecmd{end#1}{end#2}%
6319   \fi
6320 }

```

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

```

6321 \def\eql@amsmath@fixmatrix#1{%
6322   \expandafter\let\expandafter\eql@tmp\csname#1\endcsname
6323   \begingroup
6324     \let\matrix@check\@gobble
6325     \def\env@matrix{\noexpand\env@matrix}%
6326     \def\env@cases{\noexpand\env@cases}%
6327     \global\edef\eql@tmp{\eql@tmp}%
6328   \endgroup
6329   \eql@letcs{#1}\eql@tmp
6330 }
6331 \def\eql@amsmath@fixmatrixend#1{%
6332   \expandafter\let\expandafter\eql@tmp\csname end#1\endcsname
6333   \begingroup
6334     \expandafter\def\expandafter\endmatrix\expandafter{%
6335       \expandafter\unexpanded\expandafter{\endmatrix}}%
6336     \global\long\edef\eql@tmp{\eql@tmp}%
6337   \endgroup
6338   \eql@letcs{end#1}\eql@tmp
6339 }

```

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

```

6340 \def\eql@amsmath@fixends{%

```

```

6341 \eq@amsmath@after{%
6342 \eq@amsmath@endfix{flalign}{align}%
6343 \eq@amsmath@endfix{alignat}{align}%
6344 \eq@amsmath@endfix{xalignat}{align}%
6345 \eq@amsmath@endfix{xxalignat}{align}%
6346 \eq@amsmath@endfix{gather*}{gather}%
6347 \eq@amsmath@endfix{multline*}{multline}%
6348 \eq@amsmath@endfix{align*}{align}%
6349 \eq@amsmath@endfix{flalign*}{align}%
6350 \eq@amsmath@endfix{alignat*}{align}%
6351 \eq@amsmath@endfix{xalignat*}{align}%
6352 \eq@amsmath@endfix{gathered}{aligned}%
6353 \eq@amsmath@endfix{alignedat}{aligned}%
6354 }
6355 }

```

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

```

6356 \def\eq@amsmath@fixmatrices{%
6357 \eq@amsmath@after{%
6358 \eq@amsmath@fixmatrix{cases}%
6359 \eq@amsmath@fixmatrix{matrix}%
6360 \eq@amsmath@fixmatrix{pmatrix}%
6361 \eq@amsmath@fixmatrixend{pmatrix}%
6362 \eq@amsmath@fixmatrixend{bmatrix}%
6363 \eq@amsmath@fixmatrixend{Bmatrix}%
6364 \eq@amsmath@fixmatrixend{vmatrix}%
6365 \eq@amsmath@fixmatrixend{Vmatrix}%
6366 }
6367 }

```

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

```

6368 \def\eq@provide@backup@amsenv#1{%
6369 \eq@amsmath@after{%
6370 \eq@provide@moveenv{ams#1}{#1}%
6371 \eq@tagging@register@luamml{ams#1}%
6372 \eq@markline@amsthm@move{ams#1}{#1}%
6373 }%
6374 }

```

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

```

6375 \def\eq@provide@backup@amsbox#1{%
6376 \eq@amsmath@after{%
6377 \eq@provide@moveenv{ams#1}{#1}%
6378 }%
6379 }

```

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

```

6380 \def\eq@provide@backup@eqref{%
6381 \eq@amsmath@after{%
6382 \eq@provide@movecmd{amseqref}{eqref}%

```

```

6383 }%
6384 }

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

6385 \def\eql@provide@backup@multlined{%
6386   \AddToHook{package/mathtools/after}{%
6387     \eql@provide@moveenv{amsmultlined}{multlined}}%
6388 }

vide@backup@equation The LATEX environment equation is overwritten by several packages to implement their
adjustments. Here we cater for adjustments through amsmath, hyperref and the PDF
tagging mechanism. Copy equation and equation* whenever amsmath is loaded.
Whenever hyperref is loaded, and amsmath is not yet present, backup the original LATEX
and hyperref versions of equation. If neither hyperref nor amsmath are present, just
backup the original LATEX 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:

6389 \def\eql@provide@backup@equation{%
6390   \eql@amsmath@after{%
6391     \eql@provide@moveenv{amsequation}{equation}%
6392     \eql@provide@moveenv{amsequation*}{equation*}%
6393     \eql@tagging@register@env{amsequation}%
6394     \eql@tagging@register@env{amsequation*}%
6395     \eql@tagging@register@luamml{amsequation}%
6396     \eql@tagging@register@luamml{amsequation*}%
6397     \eql@markline@amsthm@move{amsequation}{equation}%
6398     \eql@markline@amsthm@move{amsequation*}{equation*}%
6399   }%
6400   \AddToHook{package/hyperref/after}{%
6401     \ifpackageloaded{amsmath}{}%
6402       \eql@provide@moveenv{hyperrefequation}{equation}%
6403       \eql@tagging@register@env{hyperrefequation}%
6404       \eql@tagging@register@luamml{hyperrefequation}%
6405       \eql@markline@amsthm@move{hyperrefequation}{equation}%
6406     }%
6407   }%
6408   \ifpackageloaded{amsmath}{}%
6409     \ifpackageloaded{hyperref}{%
6410       \let\latexequation\H@equation
6411       \let\endlatexequation\H@endequation
6412     }{\eql@provide@moveenv{latexequation}{equation}}%
6413     \eql@tagging@register@env{latexequation}%
6414     \eql@tagging@register@luamml{latexequation}%
6415     \eql@markline@amsthm@move{latexequation}{equation}%
6416   }%
6417 }

e@backup@displaymath TODO: describe

6418 \def\eql@provide@backup@displaymath{%
6419   \eql@provide@moveenv{latexdisplaymath}{displaymath}%
6420   \eql@markline@amsthm@move{latexdisplaymath}{displaymath}%
6421 }

```

o@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:

```

6422 \def\eql@provide@backup@subequations{%
6423   \eql@amsmath@after{%
6424     \eql@provide@moveenv{amssubequations}{subequations}%
6425   }%
6426   \AddToHook{package/hyperref/after}{%
6427     \AddToHook{cmd/amssubequations/before}{%
6428       {%
6429         \stepcounter{equation}%
6430         \protected@edef\theHparentequation{\theHequation}%
6431         \addtocounter{equation}{-1}%
6432       }%
6433       \AddToHook{cmd/amssubequations/after}{%
6434         {%
6435           \def\theHequation{\theHparentequation\alph{equation}}%
6436           \ignorespaces
6437         }%
6438       }%
6439 }

```

`\eql@provide@backup` Backup all amsmath environments:

```

6440 \def\eql@provide@backup{%
6441   \eql@provide@backup@eqref
6442   \eql@provide@backup@equation
6443   \eql@provide@backup@displaymath
6444   \eql@provide@backup@amsenv{gather}%
6445   \eql@provide@backup@amsenv{multline}%
6446   \eql@provide@backup@amsenv{align}%
6447   \eql@provide@backup@amsenv{flalign}%
6448   \eql@provide@backup@amsenv{alignat}%
6449   \eql@provide@backup@amsenv{xalignat}%
6450   \eql@provide@backup@amsenv{xxalignat}%
6451   \eql@provide@backup@amsenv{gather*}%
6452   \eql@provide@backup@amsenv{multline*}%
6453   \eql@provide@backup@amsenv{align*}%
6454   \eql@provide@backup@amsenv{flalign*}%
6455   \eql@provide@backup@amsenv{alignat*}%
6456   \eql@provide@backup@amsenv{xalignat*}%
6457   \eql@provide@backup@amsbox{gathered}%
6458   \eql@provide@backup@multlined
6459   \eql@provide@backup@amsbox{aligned}%
6460   \eql@provide@backup@amsbox{alignedat}%
6461   \eql@provide@backup@amsbox{cases}%
6462   \eql@provide@backup@amsbox{matrix}%
6463   \eql@provide@backup@amsbox{pmatrix}%
6464   \eql@provide@backup@amsbox{bmatrix}%
6465   \eql@provide@backup@amsbox{Bmatrix}%
6466   \eql@provide@backup@amsbox{vmatrix}%
6467   \eql@provide@backup@amsbox{Vmatrix}%
6468   \eql@provide@backup@subequations
6469 }

```

Replacement amsmath Environments. **TODO:** describe

```

6470 \def\eql@alignat@gobblecol#1{%
6471   \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.)
6472 \newenvironment{eql@gathered}
6473   {\eqnaddopt{lines}\equationsbox}{\endequationsbox}
6474 \newenvironment{eql@multlined}
6475   {\eqnaddopt{lines,padding,shape=steps}\equationsbox}{\endequationsbox}
6476 \newenvironment{eql@aligned}
6477   {\eqnaddopt{columns}\equationsbox}{\endequationsbox}
6478 \newenvironment{eql@alignedat}
6479   {\eqnaddopt{columns,colsep=off}\eql@alignat@gobblecol\equationsbox}
6480   {\endequationsbox}
6481 \newenvironment{eql@cases}
6482   {\eqnaddopt{cases}\equationsbox}{\endequationsbox}
6483 \newenvironment{eql@matrix}
6484   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=.}\equationsbox}
6485   {\endequationsbox}
6486 \newenvironment{eql@pmatrix}
6487   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=r}\equationsbox}
6488   {\endequationsbox}
6489 \newenvironment{eql@bmatrix}
6490   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=s}\equationsbox}
6491   {\endequationsbox}
6492 \newenvironment{eql@Bmatrix}
6493   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=c}\equationsbox}
6494   {\endequationsbox}
6495 \newenvironment{eql@vmatrix}
6496   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=v}\equationsbox}
6497   {\endequationsbox}
6498 \newenvironment{eql@Vmatrix}
6499   {\eqnlineset{modifierboxenv=false}\eqnaddopt{matrix=d}\equationsbox}
6500   {\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.)
6501 \newenvironment{eql@equation}
6502   {\eqnaddopt{equation,donumber}\equations}{\endequations}
6503 \newenvironment{eql@equation*}
6504   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
6505 \newenvironment{eql@displaymath}
6506   {\eqnaddopt{equation,nonumber}\equations}{\endequations}
6507 \newenvironment{eql@gather}
6508   {\eqnaddopt{lines,donumber}\equations}{\endequations}
6509 \newenvironment{eql@gather*}
6510   {\eqnaddopt{lines,nonumber}\equations}{\endequations}
6511 \newenvironment{eql@multline}
6512   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,donumber}%
6513     \equations}{\endequations}
6514 \newenvironment{eql@multline*}
6515   {\eqnaddopt{lines,padding=max,shape=steps,numberline=out,nonumber}%
6516     \equations}{\endequations}
6517 \newenvironment{eql@align}
6518   {\eqnaddopt{columns,donumber}\equations}{\endequations}
6519 \newenvironment{eql@align*}
6520   {\eqnaddopt{columns,nonumber}\equations}{\endequations}
6521 \newenvironment{eql@flalign}
6522   {\eqnaddopt{columns,fulllength,donumber}\equations}{\endequations}
```

```

6523 \newenvironment{eql@flalign*}
6524   {\eqnaddopt{columns,fulllength,nonumber}\equations}\endequations}
6525 \newenvironment{eql@alignat}
6526   {\eqnaddopt{columns,colsep=off,donumber}%
6527    \eql@alignat@gobblecol\equations}\endequations}
6528 \newenvironment{eql@xalignat}
6529   {\eqnaddopt{columns,donumber}%
6530    \eql@alignat@gobblecol\equations}\endequations}
6531 \newenvironment{eql@xxalignat}
6532   {\eqnaddopt{columns,fulllength,donumber}%
6533    \eql@alignat@gobblecol\equations}\endequations}
6534 \newenvironment{eql@alignat*}
6535   {\eqnaddopt{columns,colsep=off,nonumber}%
6536    \eql@alignat@gobblecol\equations}\endequations}
6537 \newenvironment{eql@xalignat*}
6538   {\eqnaddopt{columns,nonumber}%
6539    \eql@alignat@gobblecol\equations}\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):

```

6540 \def\eql@provide@onlyonce#1#2{%
6541   \def\eql@tmp{#2}\def\eql@tmpa{#1}%
6542   \ifx\eql@tmp\eql@tmpa
6543     \let\eql@tmp\@empty
6544   \fi
6545   \ifx\eql@tmp\@empty
6546     \let\eql@tmp\@undefined
6547     \ifx\@nodocument\relax
6548       \def\eql@tmp{#1}%
6549     \fi
6550     \ifcsname eql@provided@#1\endcsname
6551       \def\eql@tmp{#1}%
6552     \fi
6553     \eql@letcs{eql@provided@#1}\eql@true
6554   \else
6555     \def\eql@tmpa{*}%
6556     \ifx\eql@tmp\eql@tmpa
6557       \def\eql@tmp{#1}%
6558     \fi
6559   \fi
6560 }

```

\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:

```

6561 \def\eql@provide@eqref#1{%
6562   \def\eql@tmp{#1}\def\eql@tmpa{\eqref}%
6563   \ifx\eql@tmp\eql@tmpa

```

```

6564 \let\eql@tmp\@empty
6565 \fi
6566 \ifx\eql@tmp\@empty
6567 \eql@provide@onlyonce{eqref}{}%
6568 \else
6569 \def\eql@tmpa{*}%
6570 \ifx\eql@tmp\eql@tmpa
6571 \def\eql@tmp{eqref}%
6572 \else
6573 \edef\eql@tmp{\expandafter\@gobble\string#1}%
6574 \fi
6575 \fi
6576 \ifdefined\eql@tmp
6577 \expandafter\eql@provide@movecmd\expandafter{\eql@tmp}{eqleqref}%
6578 \else
6579 \eql@amsmath@after{%
6580 \eql@provide@movecmd{eqref}{eqleqref}%
6581 }%
6582 \AddToHook{package/mathtools/after}{%
6583 \eql@provide@movecmd{eqref}{eqleqref}%
6584 }%
6585 \eql@provide@movecmd{eqref}{eqleqref}%
6586 \eql@amsmath@undefine\eqref
6587 \fi
6588 }

```

\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:

```

6589 \def\eql@provide@amsenv#1#2{%
6590 \eql@provide@onlyonce{#1}{#2}%
6591 \ifdefined\eql@tmp
6592 \eql@provide@moveenv{\eql@tmp}{eqleq#1}%
6593 \eql@tagging@register@luamml{\eql@tmp}%
6594 \eql@markline@amsthm@register{\eql@tmp}%
6595 \else
6596 \eql@amsmath@after{%
6597 \eql@provide@moveenv{#1}{eqleq#1}%
6598 \eql@markline@amsthm@register{#1}%
6599 }%
6600 \AddToHook{package/mathtools/after}{%
6601 \eql@provide@moveenv{#1}{eqleq#1}%
6602 \eql@markline@amsthm@register{#1}%
6603 }%
6604 \eql@provide@moveenv{#1}{eqleq#1}%
6605 \eql@markline@amsthm@register{#1}%
6606 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6607 \fi
6608 }

```

\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:

```

6609 \def\eql@provide@amsbox#1#2{%
6610 \eql@provide@onlyonce{#1}{#2}%
6611 \ifdefined\eql@tmp
6612 \eql@provide@moveenv{\eql@tmp}{eqleq#1}%
6613 \else

```

```

6614 \eql@amsmath@after{%
6615 \eql@provide@moveenv{#1}{eql@#1}}%
6616 \AddToHook{package/mathtools/after}{%
6617 \eql@provide@moveenv{#1}{eql@#1}}%
6618 \eql@provide@moveenv{#1}{eql@#1}%
6619 \eql@amsmath@before{\eql@provide@undefineenv{#1}}%
6620 \fi
6621 }

```

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:

```

6622 \def\eql@provide@multlined#1{%
6623 \eql@provide@onlyonce{multlined}{#1}%
6624 \ifdefined\eql@tmp
6625 \eql@provide@moveenv{\eql@tmp}{eql@multlined}%
6626 \else
6627 \AddToHook{package/mathtools/after}{%
6628 \eql@provide@moveenv{multlined}{eql@multlined}}%
6629 \eql@provide@moveenv{multlined}{eql@multlined}%
6630 \ifpackageloaded{mathtools}{\AddToHook{package/mathtools/before}{%
6631 \eql@provide@undefineenv{multlined}}}%
6632 \fi
6633 }

```

\eql@provide@matrix Provide the cases and matrix environments. Copy into place, and copy again when amsmath is loaded:

```

6634 \def\eql@provide@matrix#1#2#3{%
6635 \eql@provide@onlyonce{#1}{#3}%
6636 \ifdefined\eql@tmp
6637 \eql@provide@moveenv{\eql@tmp}{eql@#1}%
6638 \eql@tagging@register@luamml{\eql@tmp}%
6639 \else
6640 \eql@amsmath@after{%
6641 \eql@provide@moveenv{#1}{eql@#1}%
6642 }%
6643 \eql@provide@moveenv{#1}{eql@#1}%
6644 \ifdefined#2\eql@amsmath@before{\eql@provide@undefineenv{#1}}\fi%
6645 \fi
6646 }

```

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:

```

6647 \def\eql@provide@equation#1{%
6648 \eql@provide@onlyonce{equation}{#1}%
6649 \ifdefined\eql@tmp
6650 \eql@provide@moveenv{\eql@tmp}{eql@equation}%
6651 \eql@tagging@register@luamml{\eql@tmp}%
6652 \eql@markline@amsthm@register{\eql@tmp}%
6653 \else
6654 \eql@amsmath@after{%
6655 \eql@provide@moveenv{equation}{eql@equation}%
6656 \eql@markline@amsthm@register{equation}%
6657 }%

```

```

6658 \AddToHook{package/hyperref/after}{%
6659 \ifpackage{amsmath}{%
6660 \eql@provide@moveenv{equation}{eql@equation}%
6661 \eql@markline@amsthm@register{equation}%
6662 }%
6663 }%
6664 \eql@provide@moveenv{equation}{eql@equation}%
6665 \eql@markline@amsthm@register{equation}%
6666 \ifdefined\eql@tagging@on
6667 \AddToHook{begindocument/end}{%
6668 \eql@provide@moveenv{equation}{eql@equation}%
6669 \eql@markline@amsthm@register{equation}%
6670 }%
6671 \fi
6672 \fi
6673 }

```

`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:

```

6674 \def\eql@provide@equationstar#1{%
6675 \eql@provide@onlyonce{equation*}{#1}%
6676 \ifdefined\eql@tmp
6677 \eql@provide@moveenv{\eql@tmp}{eql@equation*}%
6678 \eql@tagging@register@luamml{\eql@tmp}%
6679 \eql@markline@amsthm@register{\eql@tmp}%
6680 \else
6681 \eql@amsmath@after{%
6682 \eql@provide@moveenv{equation*}{eql@equation*}%
6683 \eql@markline@amsthm@register{equation*}%
6684 }%
6685 \eql@provide@moveenv{equation*}{eql@equation*}%
6686 \eql@markline@amsthm@register{equation*}%
6687 \eql@amsmath@before{\eql@provide@undefineenv{equation*}}%
6688 \ifdefined\eql@tagging@on
6689 \AddToHook{begindocument/end}{%
6690 \eql@provide@moveenv{equation*}{eql@equation*}%
6691 \eql@markline@amsthm@register{equation*}%
6692 }%
6693 \fi
6694 \fi
6695 }

```

`@provide@displaymath` **TODO:** describe

```

6696 \def\eql@provide@displaymath#1{%
6697 \eql@provide@onlyonce{displaymath}{#1}%
6698 \ifdefined\eql@tmp
6699 \eql@provide@moveenv{\eql@tmp}{eql@displaymath}%
6700 \eql@markline@amsthm@register{\eql@tmp}%
6701 \eql@tagging@register@luamml{\eql@tmp}%
6702 \else
6703 \eql@provide@moveenv{displaymath}{eql@displaymath}%
6704 \eql@markline@amsthm@register{displaymath}%
6705 \ifdefined\eql@tagging@on
6706 \AddToHook{begindocument/end}{%
6707 \eql@provide@moveenv{displaymath}{eql@displaymath}}%

```

```

6708     \fi
6709     \fi
6710 }

```

`\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:

```

6711 \def\eql@provide@subequations#1{%
6712   \eql@provide@onlyonce{subequations}{#1}%
6713   \ifdefined\eql@tmp
6714     \eql@provide@moveenv{\eql@tmp}{eql@subequations}%
6715   \else
6716     \eql@amsmath@after{%
6717       \eql@provide@moveenv{subequations}{eql@subequations}%
6718     }%
6719     \AddToHook{package/hyperref/after}{%
6720       \AddToHook{cmd/subequations/before}[hyperref]{}%
6721       \AddToHook{cmd/subequations/after}[hyperref]{}%
6722       \RemoveFromHook{cmd/subequations/before}[hyperref]%
6723       \RemoveFromHook{cmd/subequations/after}[hyperref]%
6724       \AddToHook{begindocument/end}{%
6725         \eql@provide@moveenv{subequations}{eql@subequations}}%
6726     }%
6727     \eql@provide@moveenv{subequations}{eql@subequations}%
6728     \eql@amsmath@before{\eql@provide@undefineenv{subequations}}%
6729   \fi
6730 }

```

`\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:

```

6731 \def\eql@provide@sqr{%
6732   \let\[\eql@sqr@open
6733   \let\]\eql@sqr@close
6734   \eql@amsmath@after{%
6735     \let\[\eql@sqr@open
6736     \let\]\eql@sqr@close
6737   }%
6738   \ifdefined\eql@tagging@on
6739     \AddToHook{begindocument/end}{%
6740       \let\[\eql@sqr@open
6741       \let\]\eql@sqr@close
6742     }%
6743   \fi
6744 }

```

`\eql@provide@ang` Provide the symbolic environment `\<...\>`. This is easy because none of the other packages uses this structure:

```

6745 \def\eql@provide@ang{%
6746   \let\<\eql@ang@open
6747   \let\>\eql@ang@close
6748 }

```

Interface.

`provide (key)` We provide the additional environments via key-value pairs, where the value specifies the intended name:

```
6749 \eql@define@key{provide}{equation}[]{\eql@provide@equation{#1}}
6750 \eql@define@key{provide}{equation*}[]{\eql@provide@equationstar{#1}}
6751 \eql@define@key{provide}{displaymath}[]{\eql@provide@displaymath{#1}}
6752 \eql@define@key{provide}{gather}[]{\eql@provide@amsenv{gather}{#1}}
6753 \eql@define@key{provide}{multline}[]{\eql@provide@amsenv{multline}{#1}}
6754 \eql@define@key{provide}{align}[]{\eql@provide@amsenv{align}{#1}}
6755 \eql@define@key{provide}{flalign}[]{\eql@provide@amsenv{flalign}{#1}}
6756 \eql@define@key{provide}{alignat}[]{\eql@provide@amsenv{alignat}{#1}}
6757 \eql@define@key{provide}{xalignat}[]{\eql@provide@amsenv{xalignat}{#1}}
6758 \eql@define@key{provide}{xxalignat}[]{\eql@provide@amsenv{xxalignat}{#1}}
6759 \eql@define@key{provide}{gather*}[]{\eql@provide@amsenv{gather*}{#1}}
6760 \eql@define@key{provide}{multline*}[]{\eql@provide@amsenv{multline*}{#1}}
6761 \eql@define@key{provide}{align*}[]{\eql@provide@amsenv{align*}{#1}}
6762 \eql@define@key{provide}{flalign*}[]{\eql@provide@amsenv{flalign*}{#1}}
6763 \eql@define@key{provide}{alignat*}[]{\eql@provide@amsenv{alignat*}{#1}}
6764 \eql@define@key{provide}{xalignat*}[]{\eql@provide@amsenv{xalignat*}{#1}}
6765 \eql@define@key{provide}{gathered}[]{\eql@provide@amsbox{gathered}{#1}}
6766 \eql@define@key{provide}{multlined}[]{\eql@provide@multlined{#1}}
6767 \eql@define@key{provide}{aligned}[]{\eql@provide@amsbox{aligned}{#1}}
6768 \eql@define@key{provide}{alignedat}[]{\eql@provide@amsbox{alignedat}{#1}}
6769 \eql@define@key{provide}{cases}[]{\eql@provide@matrix{cases}\eql@false{#1}}
6770 \eql@define@key{provide}{matrix}[]{\eql@provide@matrix{matrix}\eql@false{#1}}
6771 \eql@define@key{provide}{pmatrix}[]{\eql@provide@matrix{pmatrix}\eql@false{#1}}
6772 \eql@define@key{provide}{bmatrix}[]{\eql@provide@matrix{bmatrix}\eql@true{#1}}
6773 \eql@define@key{provide}{Bmatrix}[]{\eql@provide@matrix{Bmatrix}\eql@true{#1}}
6774 \eql@define@key{provide}{vmatrix}[]{\eql@provide@matrix{vmatrix}\eql@true{#1}}
6775 \eql@define@key{provide}{Vmatrix}[]{\eql@provide@matrix{Vmatrix}\eql@true{#1}}
6776 \eql@define@key{provide}{subequations}[]{\eql@provide@subequations{#1}}
6777 \eql@define@key{provide}{sqr}[]{\eql@provide@sqr}
6778 \eql@define@key{provide}{ang}[]{\eql@provide@ang}
6779 \eql@define@key{provide}{eqref}[]{\eql@provide@eqref{#1}}
6780 \eql@define@key{provide}{tagform}[]{%
6781   \def\tagform##1{\maketag@@@{\eql@tags@tagform{#1}}}%
6782 \eql@define@key{provide}{maketag}[]{%
6783   \def\maketag@@@##1{\eql@tags@taglayout{##1}}}
```

`\eqnlinesprovide` Provide an additional environment or macro via key-value interface:

```
6784 \newcommand{\eqnlinesprovide}[1]{%
6785   \eql@setkeys{provide}{#1}%
6786   \ignorespaces
6787 }
```

16.5 Global and Package Options

Handle global and package options:

Disable error message for exclusive package options:

```
6788 \let\eql@error@packageoption\@gobble
```

Declare math layout options `leqno` and `fleqn` for common L^AT_EX classes:

```
6789 \DeclareOption{leqno}{\eqnlineset{tagsleft}}
```



```
6790 \DeclareOption{fleqn}{\eqnlineset{left}}
```

Pass undeclared options on to keyval processing:

```
6791 \DeclareOption*{\expandafter\eqnlineset\expandafter{\CurrentOption}}
```

Set defaults for package:

```
6792 \eqldefaults@eqnlines
6793 \eqlequations@columns@set
6794 \eqlbox@columns@set
```

Make sure that the `amsmath` conditionals `\iftagsleft@` and `\if@fleqn` are declared without spelling out their name which may upset the \TeX conditional parsing mechanism:

```
6795 \ifdefined\tagsleft@true\else
6796   \expandafter\newif\csname iftagsleft@\endcsname
6797 \fi
6798 \ifdefined\@fleqntrue\else
6799   \expandafter\newif\csname if@fleqn\endcsname
6800 \fi
```

Import `amsmath` switches `leqno` as `tagsleft` and `fleqn` as `left`:

```
6801 \eql@amsmath@after{%
6802   \ifnum\eql@provide@opt@env=\tw@
6803     \iftagsleft@
6804       \eqnlineset{tags=left}%
6805     \else
6806       \eqnlineset{tags=right}%
6807     \fi
6808     \if@fleqn
6809       \eqnlineset{layout=left}%
6810     \else
6811       \eqnlineset{layout=center}%
6812     \fi
6813   \fi
6814 }
```

Process package options:

```
6815 \ProcessOptions
```

`\error@packageoption` Enable error message for exclusive package options:

```
6816 \def\eql@error@packageoption#1{%
6817   \eql@error{may only use '#1' as a package option}%
6818 }
```

Make the ending statements for `amsmath` environments independent if desired, so that they may be overwritten individually:

```
6819 \ifnum\eql@provide@opt@env=\tw@
6820 \ifdefined\eql@provide@opt@matrix
6821   \let\eql@provide@opt@amsmathpatch\eql@false
6822 \fi\fi
6823 \ifdefined\eql@provide@opt@backup
6824   \let\eql@provide@opt@amsmathpatch\eql@true
6825 \fi
6826 \ifdefined\eql@provide@opt@amsmathpatch
6827   \eql@amsmath@fixends
6828   \eql@amsmath@fixmatrices
6829 \fi
```

Backup all amsmath environments that may be overwritten to `ams...`. This will happen before any replacements:

```
6830 \ifdefined\eqL@provide@opt@backup\eqL@provide@backup\fi
```

Provide native L^AT_EX environment `equation` and symbolic shortcut `\[...\]` if desired:

```
6831 \ifnum\eqL@provide@opt@env>\z@
6832   \eqnlinesprovide{equation,equation*,sqr,displaymath}
6833 \fi
```

Provide amsmath equation environments if desired:

```
6834 \ifnum\eqL@provide@opt@env=\tw@
6835   \eqnlinesprovide{%
6836     multiline,gather,align,flalign,alignat,xalignat,xxalignat,%
6837     multiline*,gather*,align*,flalign*,alignat*,xalignat*,%
6838     multlined,gathered,aligned,alignedat,%
6839     subequations}
6840 \fi
```

Provide symbolic shortcut `\<...\>` if desired:

```
6841 \ifdefined\eqL@provide@opt@ang\eqnlinesprovide{ang}\fi
```

Provide equation reference `\eqref` if desired:

```
6842 \ifdefined\eqL@provide@opt@eqref\eqnlinesprovide{eqref}\fi
```

Provide `cases` and `matrix` environments if desired:

```
6843 \ifdefined\eqL@provide@opt@matrix
6844   \eqnlinesprovide{cases,matrix,pmatrix,bmatrix,Bmatrix,vmatrix,Vmatrix}
6845 \fi
```