

1 *lineno.sty v3.03 2000/03/10*

2

3 A L^AT_EX style option to attach
4 Line numbers on paragraphs

5 Stephan I. Böttcher

6 stephan@nevis.columbia.edu

7 **Contents**

8	1	Introduction	2
9	2	Put the line numbers to the lines	3
10	3	Control line numbering	5
11	3.1	Display math	7
12	4	Line number references	8
13	4.1	The linelabel command	9
14	5	The appearance of the line numbers	10
15	5.1	Running line numbers	11
16	5.2	Pagewise line numbers	12
17	5.3	Numbering modulo 5	16
18	6	Package options	17
19	6.1	Package Extensions	18
20	6.1.1	<i>displaymath</i>	19
21	6.1.2	Line numbers in internal vertical mode	19
22	6.1.3	Line number references with offset	20
23	6.1.4	Numbered quotation environments	21
24	6.1.5	Frame around a paragraph	21

1	7	The final touch	22
2	8	The user commands	23
3	8.1	Customization hooks	25

4 **1 Introduction**

5 This package provides line numbers on paragraphs. After T_EX has broken
6 a paragraph into lines there will be line numbers attached to them, with
7 the possibility to make references through the L^AT_EX `\ref`, `\pageref` cross
8 reference mechanism. This includes four issues:

- 9 • attach a line number on each line,
- 10 • create references to a line number,
- 11 • control line numbering mode,
- 12 • count the lines and print the numbers.

13 The first two points are implemented through patches to the output routine.
14 The third by redefining `\par`, `\@par` and `\@@par`. The counting is easy, as
15 long as you want the line numbers run through the text. If they shall start
16 over at the top of each page, the aux-file as well as T_EXs memory have to
17 carry a load for each counted line.

18 I wrote this package for my wife Petra, who needs it for transcriptions
19 of interviews. This allows her to precisely refer to passages in the text. It
20 works well together with `\marginpars`, but not to well with `displaymath`.
21 `\footnotes` are a problem, especially when they are split, but we may get
22 there. `lineno.sty` works surprisingly well with other packages, for example,
23 `wrapfig.sty`. So please try if it works with whatever you need, and if it
24 does, please tell me, and if it does not, tell me as well, so I can try to fix it.

25 This style option is written for L^AT_EX 2_ε, later than November 1994, since
26 we need the `\protected@write` macro.

```

1 \NeedsTeXFormat{LaTeX2e}[1994/11/04]
2 \ProvidesPackage{lineno}
3 [2000/03/10 line numbers on paragraphs v3.03]

```

1 2 Put the line numbers to the lines

2 The line numbers have to be attached by the output routine. We simply set
3 the `\interlinepenalty` to -100000. The output routine will be called after
4 each line in the paragraph, except the last, where we trigger by `\par`. The
5 `\linenopenalty` is small enough to compensate a bunch of penalties (e.g.,
6 with `\samepage`).

```
4 \newcount\linenopenalty\linenopenalty=-100000
5 \mathchardef\linenopenaltypar=25000
```

7 So let's make a hook to `\output`, the direct way. The \LaTeX macro
8 `\@reinserts` puts the footnotes back on the page.

9 (New v3.01) `\@reinserts` badly screws up split footnotes. The bottom
10 part is still on the recent contributions list, and the top part will be put back
11 there after the bottom part. Thus, since `lineno.sty` does not play well with
12 `\inserts` anyway, we can safely experiment with `\holdinginserts`, without
13 making things much worse.

14 Or that's what I thought, but: Just activating `\holdinginserts` while
15 doing the `\par` will not do the trick: The `\output` routine may be called
16 for a real page break before all line numbers are done, and how can we get
17 control over `\holdinginserts` at that point?

18 Let's try this: When the `\output` routine is run with `\holdinginserts=3`
19 for a real page break, then we reset `\holdinginserts` and restart `\output`.

20 Then, again, how do we keep the remaining `\inserts` while doing further
21 line numbers?

22 If we find `\holdinginserts=-3` we activate it again after doing `\output`.
23 (/New v3.01)

24 (New v3.02) To work with `multicol.sty`, the original output routine is
25 now called indirectly, instead of being replaced. When `multicol.sty` changes
26 `\output`, it is a toks register, not the real thing. (/New v3.02)

```
6 \let\@LN@output\output
7 \newtoks\output
8 \output=\expandafter{\the\@LN@output}
9 \@LN@output={%
10     \LineNoTest
11     \if@tempswa
12     \LineNoHoldInsertsTest
13     \if@tempswa
14     \the\output
15     \ifnum\holdinginserts=-3
16     \global\holdinginserts 3
17     \fi
```

```

18         \else
19             \global\holdinginserts-3
20             \unvbox\@cclv
21             \ifnum\outputpenalty=10000\else
22                 \penalty\outputpenalty
23             \fi
24         \fi
25     \else
26         \MakeLineNo
27     \fi
28 }

```

1 The float mechanism inserts `\interlinepenalty`s during `\output`. So care-
2 fully reset it before going on. Else we get doubled line numbers on every
3 float placed in horizontal mode, e.g, from `\linelabel`.

4 Sorry, neither a `\linelabel` nor a `\marginpar` should insert a penalty,
5 else the following line number could go to the next page. Nor should any
6 other float. So let us suppress the `\interlinepenalty` altogether with the
7 `\@nobreak` switch.

8 Since (ltspace.dtx, v1.2p)[1996/07/26], the `\@nobreaktrue` does it's job
9 globally. We need to do it locally here.

```

29 \def\LineNoTest{%
30     \let\@par\@par
31     \ifnum\interlinepenalty<-\linenopenaltypar
32         \advance\interlinepenalty-\linenopenalty
33         \my@nobreaktrue
34     \fi
35     \@tempwattrue
36     \ifnum\outputpenalty>-\linenopenaltypar\else
37         \ifnum\outputpenalty>-175000\relax
38             \@tempwafalse
39         \fi
40     \fi
41 }
42
43 \def\my@nobreaktrue{\let\if@nobreak\iftrue}
44
45 \def\LineNoHoldInsertsTest{%
46     \ifnum\holdinginserts=3\relax
47         \@tempwafalse
48     \fi
49 }

```

10 We have to return all the page to the current page, and add a box with the
11 line number, without adding breakpoints, glue or space. The depth of our

1 line number should be equal to the previous depth of the page, in case the
2 page breaks here, and the box has to be moved up by that depth.
3 The `\interlinepenalty` comes after the `\adjust` from a `\linelabel`,
4 so we increment the line number *after* printing it. The macro
5 `\makeLineNumber` produces the text of the line number, see section 5. Finally
6 we put in the natural `\interlinepenalty`, except after the last line.

```

50 \def\MakeLineNo{\@tempdima\dp\@cclv \unvbox\@cclv
51   \sbox\@tempboxa{\hbox to\z@\makeLineNumber}}%
52   \stepcounter{linenumber}%
53   \dp\@tempboxa=\@tempdima\ht\@tempboxa=\z@
54   \nointerlineskip\kern-\@tempdima\box\@tempboxa
55   \ifnum\outputpenalty=-\linenopenalty\else
56     \@tempcnta\outputpenalty
57     \advance\@tempcnta -\linenopenalty
58     \penalty\@tempcnta
59   \fi
60 }

```

7 3 Control line numbering

8 The line numbering is controlled via `\par`. L^AT_EX saved the T_EX-primitive
9 `\par` in `\@@par`. We push it one level further out, and redefine `\@@par` to
10 insert the `\interlinepenalty` needed to trigger the line numbering. And
11 we need to allow pagebreaks after a paragraph.

12 New (2.05beta): the `prevgraf` test. A paragraph the ends with a displayed
13 equation, a `\noindent\par` or `wrapfig.sty` produce empty paragraphs.
14 These should not get a spurious line number via `\linenopenaltypar`.

```

61 \let\@@@par\@@par
62 \newcount\linenoprevgraf
63
64 \def\linenumberpar{\ifvmode\@@@par\else\ifinner\@@@par\else
65   \advance\interlinepenalty \linenopenalty
66   \linenoprevgraf\prevgraf
67   \global\holdinginserts3%
68   \@@@par
69   \ifnum\prevgraf>\linenoprevgraf
70     \penalty-\linenopenaltypar
71   \fi
72   \kern\z@
73   \global\holdinginserts0%
74   \advance\interlinepenalty -\linenopenalty
75   \fi\fi
76 }

```

1 The basic commands to enable and disable line numbers. `\@par` and `\par`
 2 are only touched, when they are `\let` to `\@@@par/\linenumberpar`. The line
 3 number may be reset to 1 with the star-form, or set by an optional argument
 4 [*number*].

```

77 \def\linenumbers{\let\@par\linenumberpar
78     \ifx\@par\@@@par\let\@par\linenumberpar\fi
79     \ifx\par\@@@par\let\par\linenumberpar\fi
80     \@ifnextchar[{\resetlinenumber}%
81         {\@ifstar{\resetlinenumber}{}}%
82     }
83
84 \def\nolinenumbers{\let\@par\@@@par
85     \ifx\@par\linenumberpar\let\@par\@@@par\fi
86     \ifx\par\linenumberpar\let\par\@@@par\fi
87     }

```

What happens with a display math? Since `\par` is not executed, when breaking the lines before a display, they will not get line numbers. Sorry, but I do not dare to change `\interlinepenalty` globally, nor do I want to redefine the display math environments here.

display math

5 See the subsection below, for a wrapper environment to make it work. But
 6 that requires to wrap each and every display in your LaTeX source.

7 The next two commands are provided to turn on line numbering in
 8 a specific mode. Please note the difference: for pagewise numbering,
 9 `\linenumbers` comes first to inhibit it from seeing optional arguments, since
 10 re-/presetting the counter is useless.

```

88 \def\pagewiselinenumbers{\linenumbers\setpagewiselinenumbers}
89 \def\runninglinenumbers{\setrunninglinenumbers\linenumbers}

```

11 Finally, it is a L^AT_EX style, so we provide for the use of environments, includ-
 12 ing the suppression of the following paragraphs indentation.

```

90 \@namedef{linenumbers*}{\par\linenumbers*}
91 \@namedef{runninglinenumbers*}{\par\runninglinenumbers*}
92
93 \def\endlinenumbers{\par\@endptrue}
94 \let\endrunninglinenumbers\endlinenumbers
95 \let\endpagewiselinenumbers\endlinenumbers
96 \expandafter\let\csname endlinenumbers*\endcsname\endlinenumbers
97 \expandafter\let\csname endrunninglinenumbers*\endcsname\endlinenumbers
98 \let\endnolinenumbers\endlinenumbers

```

1 3.1 Display math

2 Now we tackle the problem to get display math working. There are different
3 options.

- 4 1. Precede every display math with a `\par`. Not too good.
- 5 2. Change `\interlinepenalty` and associates globally. Unstable.
- 6 3. Wrap each display math with a `{\linenomath}` environment.

7 We'll go for option 3. See if it works:

$$display\ math \tag{1}$$

8 The star form `{\linenomath*}` should also number the lines of the display
9 itself,

$$10 \quad \quad \quad multi \quad \quad \quad line \tag{2}$$

$$11 \quad \quad \quad display \quad \quad \quad math \tag{3}$$

$$12 \quad \quad \quad \quad \quad \quad with \tag{4}$$
$$\quad \quad \quad \quad \quad \quad array$$

13 including multiline displays.

14 First, here are two macros to turn on linenumbering on paragraphs pre-
15 ceeding displays, with numbering the lines of the display itself, or without.
16 The `\ifx..` tests if line numbering is turned on. It does not harm to add
17 these wrappers in sections that are not numbered. Nor does it harm to wrap
18 a display twice, e.g, in case you have some `{equation}`s wrapped explicitly,
19 and later you redefine `\equation` to do it automatically.

```
99 \newcommand\linenomathNonumbers{%  
100 \ifx\@par\@@par\else  
101 \ifnum\interlinepenalty>-\linenopenaltypar  
102 \global\holdinginserts3%  
103 \advance\interlinepenalty \linenopenalty  
104 \advance\predisplaypenalty \linenopenalty  
105 \fi  
106 \fi  
107 \ignorespaces  
108 }  
109  
110 \newcommand\linenomathWithnumbers{%  
111 \ifx\@par\@@par\else  
112 \ifnum\interlinepenalty>-\linenopenaltypar  
113 \global\holdinginserts3%
```

```

114     \advance\interlinepenalty \linenopenalty
115     \advance\predisplaypenalty \linenopenalty
116     \advance\postdisplaypenalty \linenopenalty
117     \advance\interdisplaylinepenalty \linenopenalty
118     \fi
119     \fi
120     \ignorespaces
121   }

```

1 The `{linenomath}` environment has two forms, with and without a star. The
2 following two macros define the environment, where the starred/non-starred
3 form does/doesn't number the lines of the display or vice versa.

```

122 \newcommand\linenumberdisplaymath{%
123   \def\linenomath{\linenomathWithnumbers}%
124   \@namedef{linenomath*}{\linenomathNonumbers}%
125   }
126
127 \newcommand\nolinenumberdisplaymath{%
128   \def\linenomath{\linenomathNonumbers}%
129   \@namedef{linenomath*}{\linenomathWithnumbers}%
130   }
131
132 \def\endlinenomath{%
133   \global\holdinginserts0
134   \@ignoretrue
135 }
136 \expandafter\let\csname endlinenomath*\endcsname\endlinenomath

```

4 The default is not to number the lines of a display. But the package option
5 `mathlines` may be used to switch that behavior.

```

137 \nolinenumberdisplaymath

```

6 4 Line number references

7 The only way to get a label to a line number in a paragraph is to ask the
8 output routine to mark it.

9 We use the marginpar mechanism to hook to `\output` for a second time.
10 Marginpars are floats with number -1 , we fake marginpars with No -2 .
11 Originally, every negative numbered float was considered to be a marginpar.

12 The float box number `\@currbox` is used to transfer the label name in a
13 macro called `\@LNL@{box-number}`.

14 A `\newlabel` is written to the aux-file. The reference is to
15 `\theLineNumber`, *not* `\thelinenumber`. This allows to hook in, as done
16 below for pagewise line numbering.

1 (New v3.03) The `\@LN@ExtraLabelItems` are added for a hook to keep
2 packages like `{hyperref}` happy. (/New v3.03)

```
138 \let\@LN@addmarginpar\@addmarginpar
139 \def\@addmarginpar{%
140   \ifnum\count\@currbox>-2\relax
141     \expandafter\@LN@addmarginpar
142   \else
143     \@cons\@freelist\@currbox
144     \protected@write\@auxout{}\{%
145       \string\newlabel
146         {\csname @LNL@the\@currbox\endcsname}%
147         {\theLineNumber}\thepage}\@LN@ExtraLabelItems}}%
148   \fi}
149
150 \let\@LN@ExtraLabelItems\@empty
```

3 4.1 The `linelabel` command

4 To refer to a place in line `\ref{<foo>}` at page `\pageref{<foo>}` you place a
5 `\linelabel{<foo>}` at that place.

6 If you use this command outside a `\linenumbers` paragraph, you will
7 get references to some bogus line numbers, sorry. But we don't disable the
8 command, because only the `\par` at the end of a paragraph may decide
9 whether to print line numbers on this paragraph or not. A `\linelabel` may
10 legally appear earlier than `\linenumbers`.

See if it
works:
This
paragraph
starts on
page 9,
line 6.

11 `\linelabel`, via a fake float number `-2`, puts a `\penalty` into a
12 `\vadjust`, which triggers the pagebuilder after putting the current line to
13 the main vertical list. A `\write` is placed on the main vertical list, which
14 prints a reference to the current value of `\thelinenumbers` and `\thepage` at
15 the time of the `\shipout`.

16 A `\linelabel` is allowed only in outer horizontal mode. In outer ver-
17 tical mode we start a paragraph, and ignore trailing spaces (by fooling
18 `\@esphack`).

19 The argument of `\linelabel` is put into a macro with a name derived
20 from the number of the allocated float box. Much of the rest is dummy float
21 setup.

```
151 \def\linelabel#1{%
152   \ifvmode
153     \ifinner \else
154       \leavevmode \@bsphack \@savsk\p@
155     \fi
156   \else
```

```

157     \bsphack
158 \fi
159 \ifhmode
160   \ifinner
161     \@parmoderr
162   \else
163     \@floatpenalty -\@Mii
164     \@next\@currbox\@freelist
165     {\global\count\@currbox-2%
166      \expandafter\gdef\csname @LNL@\the\@currbox\endcsname{#1}}%
167     {\@floatpenalty\z@ \@fltovf \def\@currbox{\@tempboxa}}%
168   \begingroup
169     \setbox\@currbox \color@vbox \vbox \bgroup \end@float
170   \endgroup
171   \@ignorefalse \@esphack
172 \fi
173 \else
174   \@parmoderr
175 \fi
176 }

```

5 The appearance of the line numbers

The line numbers are set as `\tiny\sffamily\arabic{linenumber}`, 10pt
3 left of the text. With options to place it right of the text, or . . .
. . . here are the hooks:

```

177 \def\makeLineNumberLeft{\hss\linenumberfont\LineNumber\hskip\linenumbersep}
178
179 \def\makeLineNumberRight{\linenumberfont\hskip\linenumbersep\hskip\textwidth
180                          \hbox to\linenumberwidth{\hss\LineNumber}\hss}
181
182 \def\linenumberfont{\normalfont\tiny\sffamily}
183
184 \newdimen\linenumbersep
185 \newdimen\linenumberwidth
186
187 \linenumberwidth=10pt
188 \linenumbersep=10pt

```

Margin switching requires `pagewise` numbering mode, but choosing the left
6 or right margin for the numbers always works.

```

189 \def\switchlinenumbers{\@ifstar
190   {\let\makeLineNumberOdd\makeLineNumberRight
191    \let\makeLineNumberEven\makeLineNumberLeft}%
192   {\let\makeLineNumberOdd\makeLineNumberLeft

```

```

193     \let\makeLineNumberEven\makeLineNumberRight}%
194   }
195
196 \def\setmakelinenumbers#1{\@ifstar
197   {\let\makeLineNumberRunning#1%
198     \let\makeLineNumberOdd#1%
199     \let\makeLineNumberEven#1}%
200   {\ifx\c@linenumber\c@runninglinenumber
201     \let\makeLineNumberRunning#1%
202     \else
203       \let\makeLineNumberOdd#1%
204       \let\makeLineNumberEven#1%
205     \fi}%
206   }
207
208 \def\leftlinenumbers{\setmakelinenumbers\makeLineNumberLeft}
209 \def\rightlinenumbers{\setmakelinenumbers\makeLineNumberRight}
210
211 \leftlinenumbers*

```

`\LineNumber` is a hook which is used for the modulo stuff. It is the command to use for the line number, when you customize `\makeLineNumber`. Use `\thelinenumber` to change the outfit of the digits.

We will implement two modes of operation:

- numbers `running` through (parts of) the text
- `pagewise` numbers starting over with one on top of each page.

Both modes have their own count register, but only one is allocated as a \LaTeX counter, with the attached facilities serving both.

```

212 \newcounter{linenumber}
213 \newcount\c@pagewiselinenumber
214 \let\c@runninglinenumber\c@linenumber

```

- Only the running mode counter may be reset, or preset, for individual paragraphs. The pagewise counter must give a unique anonymous number for each line.

```

215 \newcommand\resetlinenumber[1][1]{\c@runninglinenumber#1}

```

5.1 Running line numbers

Running mode is easy, `\LineNumber` and `\theLineNumber` produce `\thelinenumber`, which defaults to `\arabic{linenumber}`, using the `\c@runninglinenumber` counter. This is the default mode of operation.

```

216 \def\makeRunningLineNumber{\makeLineNumberRunning}
217
218 \def\setrunninglinenumbers{%
219   \def\theLineNumber{\thelinenumber}%
220   \let\c@linenumber\c@runninglinenumber
221   \let\makeLineNumber\makeRunningLineNumber
222 }
223
224 \setrunninglinenumbers\resetlinenumber

```

5.2 Pagewise line numbers

Difficult, if you think about it. The number has to be printed when there is
3 no means to know on which page it will end up, except through the aux-file.
My solution is really expensive, but quite robust.

With version v2.00 the hashsize requirements are reduced, because we
6 do not need one controlsequence for each line any more. But this costs some
computation time to find out on which page we are.

\makeLineNumber gets a hook to log the line and page number to the
9 aux-file. Another hook tries to find out what the page offset is, and
subtracts it from the counter \c@linenumber. Additionally, the switch
\ifoddNumberedPage is set true for odd numbered pages, false otherwise.

```

225 \def\setpagewiselinenumbers{%
226   \let\theLineNumber\thePagewiseLineNumber
227   \let\c@linenumber\c@pagewiselinenumber
228   \let\makeLineNumber\makePagewiseLineNumber
229 }
230
231 \def\makePagewiseLineNumber{\logtheLineNumber\getLineNumber
232   \ifoddNumberedPage
233     \makeLineNumberOdd
234   \else
235     \makeLineNumberEven
236   \fi
237 }

```

12 Each numbered line gives a line to the aux file

$$\backslash@LN\{\langle line \rangle\}\{\langle page \rangle\}$$

very similar to the \newlabel business, except that we need an arabic rep-
15 resentation of the page number, not what there might else be in \thepage.

```

238 \def\logtheLineNumber{\protected@write\@auxout{}{\%
239   \string\@LN{\the\c@linenumber}\noexpand\the\c@page}}

```

From the aux-file we get one macro `\LN@P⟨page⟩` for each page with line numbers on it. This macro calls four other macros with one argument each.

3 These macros are dynamically defined to do tests and actions, to find out on which page the current line number is located.

We need sort of a pointer to the first page with line numbers, initialized to point to nothing:

```
240 \def\LastNumberedPage{first}
241 \def\LN@Pfirst{\nextLN\relax}
```

The four dynamic macros are initialized to reproduce themselves in an `\xdef`

```
242 \let\lastLN\relax % compare to last line on this page
243 \let\firstLN\relax % compare to first line on this page
244 \let\pageLN\relax % get the page number, compute the linenumber
245 \let\nextLN\relax % move to the next page
```

During the end-document run through the aux-files, we disable `\@LN`. I may put in a check here later, to give a rerun recommendation.

```
246 \AtEndDocument{\let\@LN\@gobbletwo}
```

Now, this is the tricky part. First of all, the whole definition of `\@LN` is grouped, to avoid accumulation on the save stack. Somehow `\csname⟨cs⟩\endcsname` pushes an entry, which stays after an `\xdef` to that `⟨cs⟩`.

If `\LN@P⟨page⟩` is undefined, initialize it with the current page and line number, with the *pointer-to-the-next-page* pointing to nothing. And the macro for the previous page will be redefined to point to the current one.

If the macro for the current page already exists, just redefine the *last-line-number* entry.

Finally, save the current page number, to get the pointer to the following page later.

```
247 \def\@LN#1#2{\expandafter\@LN\csname LN@P#2\endcsname{#1}{#2}}
248 \def\@LN#1#2#3{\ifx#1\relax
249   \expandafter\@@LN\csname LN@P\LastNumberedPage\endcsname#1
250   \xdef#1{\lastLN{#2}\firstLN{#2}\pageLN{#3}\nextLN\relax}%
251 \else
252   \def\lastLN##1{\noexpand\lastLN{#2}}%
253   \xdef#1{#1}%
254 \fi
255 \gdef\LastNumberedPage{#3}}
```

The previous page macro gets its pointer to the current one, replacing the `\relax` with the cs-token `\LN@P<page>`. In case of page number mismatch, `TeX` will trip here, because the argument string for `\nextLN` is not `\relax`. I think it's difficult to do a reasonable intercept here, because this is running in an `\xdef`. Does `\PackageError{}` work in there?

```
256 \def\@@@LN#1#2{{\def\nextLN\relax{\noexpand\nextLN\noexpand#2}%
257 \xdef#1{#1}}}
```

Now, to print a line number, we need to find the page, where it resides. This will most probably be the one where the last one came from or maybe the next. However, it can be a completely different one. We maintain a cache, which is let to the last accessed pages macro. But for now it is initialized to expand `\LN@first`, where the pointer to the first numbered page has been stored in.

```
258 \def\NumberedPageCache{\LN@Pfirst}
```

To find out on which page the current `\c@linenumber` is, we define the four dynamic macros to do something useful and execute the current cache macro. `\lastLN` is run first, testing if the line number in question may be on a later page. If so, disable `\firstLN`, and go on to the next page via `\nextLN`.

```
259 \def\testLastNumberedPage#1{\ifnum#1<\c@linenumber
260 \let\firstLN@gobble
261 \fi}
```

Else, if `\firstLN` finds out that we need an earlier page, we start over from the beginning. Else, `\nextLN` will be disabled, and `\pageLN` will run `\gotNumberedPage` with two arguments: the first line number on this page, and the page number.

```
262 \def\testFirstNumberedPage#1{\ifnum#1>\c@linenumber
263 \def\nextLN##1{\testNextNumberedPage\LN@Pfirst}%
264 \else
265 \let\nextLN@gobble
266 \def\pageLN{\gotNumberedPage{#1}}%
267 \fi}
```

We start with `\pageLN` disabled and `\nextLN` defined to continue the search with the next page.

```

268 \def\testNumberedPage{%
269   \let\lastLN\testLastNumberedPage
270   \let\firstLN\testFirstNumberedPage
271   \let\pageLN@gobble
272   \let\nextLN\testNextNumberedPage
273   \NumberedPageCache
274 }

```

When we switch to another page, we first have to make sure that it is there. If we are done with the last page, we probably need to run T_EX again, but for the rest of this run, the cache macro will just return two zeros. This saves a lot of time, for example if you have half of an aux-file from an aborted run, in the next run the whole page-list would be searched in vain again and again for the second half of the document.

If there is another page, we iterate the search.

```

275 \def\testNextNumberedPage#1{\ifx#1\relax
276   \global\def\NumberedPageCache{\gotNumberedPage00}%
277   \PackageWarningNoLine{lineno}%
278     {Linenummer reference failed,
279     \MessageBreak  rerun to get it right}%
280   \else
281     \global\let\NumberedPageCache#1%
282   \fi
283   \testNumberedPage
284 }

```

To separate the official hooks from the internals there is this equivalence, to hook in later for whatever purpose:

```

285 \let\getLineNumber\testNumberedPage

```

Let's see if it finds the label on page 9, line 6, and back here on page 15, line 8.

So, now we got the page where the number is on. We establish if we are on an odd or even page, and calculate the final line number to be printed.

```

286 \newif\ifoddNumberedPage
287
288 \def\gotNumberedPage#1#2{\oddNumberedPagefalse
289   \ifodd#2\relax\oddNumberedPagetrue\fi
290   \advance\c@linenummer 1\relax
291   \subtractlinenummeroffset{#1}%
292 }

```

You might want to run the pagewise mode with running line numbers, or you might not. It's your choice:

```

293 \def\runningpagewiselinenumbers{%
294   \let\subtractlinenumberoffset\@gobble
295 }
296
297 \def\realpagewiselinenumbers{%
298   \def\subtractlinenumberoffset##1{\advance\c@linenumber-##1\relax}%
299 }
300
301 \realpagewiselinenumbers

```

For line number references, we need a protected call to the whole procedure, with the requested line number stored in the `\c@linenumber` counter. This is what gets printed to the aux-file to make a label:

```

302 \def\thePagewiseLineNumber{\protect
303   \getpagewiselinenumbers{\the\c@linenumber}}%

```

And here is what happens when the label is referred to:

```

304 \def\getpagewiselinenumbers#1{%
305   \c@linenumber #1\relax\testNumberedPage
306   \thelinenumber
307 }%

```

A summary of all per line expenses:

6 **CPU:** The `\output` routine is called for each line, and the page-search is done.

DISK: One line of output to the aux-file for each numbered line

9 **MEM:** One macro per page. Great improvement over v1.02, which had one control sequence per line in addition. It blew the hash table after some five thousand lines.

12 5.3 Numbering modulo 5

Most users want to have only one in five lines numbered. `\LineNumber` is supposed to produce the outfit of the line number attached to the line, while `\thelinenumber` is used also for references, which should appear even if they are not multiples of five.

```

308 \newcount\c@linenumbermodulo
309
310 \def\themodulolinenumber{\@tempcnta\c@linenumber
311   \divide\@tempcnta\c@linenumbermodulo
312   \multiply\@tempcnta\c@linenumbermodulo
313   \ifnum\@tempcnta=\c@linenumber\thelinenumber\fi
314 }%

```

The user command to set the modulo counter:

```
315 \newcommand\modulolinenumbers[1][0]{%
316 \let\LineNumber\thomodulolinenumber
317 \ifnum#1>1\relax
318   \c@linenumbermodulo#1\relax
319 \else\ifnum#1=1\relax
320   \def\LineNumber{\thelinenumber}%
321 \fi\fi
322 }
323
324 \setcounter{linenumbermodulo}{5}
325 \modulolinenumbers[1]
```

2 6 Package options

3 There is a bunch of package options, all of them executing only user com-
4 mands (see below).

5 Options `left` (`right`) put the line numbers on the left (right) margin.

6 This works in all modes. `left` is the default.

```
326 \DeclareOption{left}{\leftlinenumbers*}
327
328 \DeclareOption{right}{\rightlinenumbers*}
```

7 Option `switch` (`switch*`) puts the line numbers on the outer (inner) margin
8 of the text. This requires running the pagewise mode, but we turn off the
9 page offset subtraction, getting sort of running numbers again. The `pagewise`
10 option may restore true pagewise mode later.

```
329 \DeclareOption{switch}{\setpagewiselinenumbers
330   \switchlinenumbers
331   \runningpagewiselinenumbers}
332
333 \DeclareOption{switch*}{\setpagewiselinenumbers
334   \switchlinenumbers*%
335   \runningpagewiselinenumbers}
```

11 The options `pagewise` and `running` select the major linenumber mechanism.
12 `running` line numbers refer to a real counter value, which can be reset for
13 any paragraph, even getting multiple paragraphs on one page starting with
14 line number one. `pagewise` line numbers get a unique hidden number within
15 the document, but with the opportunity to establish the page on which they
16 finally come to rest. This allows the subtraction of the page offset, getting

the numbers starting with 1 on top of each page, and margin switching in twoside formats becomes possible. The default mode is `running`.

The order of declaration of the options is important here `pagewise` must come after `switch`, to override running pagewise mode. `running` comes last, to reset the running line number mode, e.g, after selecting margin switch mode for `pagewise` running. Once more, if you specify all three of the options [`switch,pagewise,running`], the result is almost nothing, but if you later say `\pagewiselinenumbers`, you get margin switching, with real pagewise line numbers.

```
336 \DeclareOption{pagewise}{\setpagewiselinenumbers
337                          \realpagewiselinenumbers}
338
339 \DeclareOption{running}{\setrunninglinenumbers}
```

The option `modulo` causes only those linenumbers to be printed which are multiples of five.

```
340 \DeclareOption{modulo}{\modulolinenumbers\relax}
```

The package option `mathlines` switches the behavior of the `{linenomath}` environment with its star-form. Without this option, the `{linenomath}` environment does not number the lines of the display, while the star-form does. With this option, its just the opposite.

```
341 \DeclareOption{mathlines}{\linenumberdisplaymath}
```

`displaymath` now calls for wrappers of the standard LaTeX display math environment. This was previously done by `mlineno.sty`.

```
342 \let\do@mlineno\relax
343 \DeclareOption{displaymath}{\let\do@mlineno\@empty}
```

The `hyperref` package, via `nameref`, requires three more groups in the second argument of a `\newlabel`. Well, why shouldn't it get them? TODO: This should be automagically detected `\AtBeginDocument`.

```
344 \DeclareOption{hyperref}{\def\@LN@ExtraLabelItems{ }{ }{ }}
345
346 \ProcessOptions
```

6.1 Package Extensions

The extensions in this section were previously supplied in separate `.sty` files.

1 6.1.1 *displaymath*

2 The standard L^AT_EX display math environments are wrapped in a
3 `{\linenomath}` environment.

```
347 \ifx\do@mlineno\@empty
348
349 \renewenvironment{displaymath}
350   {\linenomath\[]
351   {\}\endlinenomath}
352
353 \renewenvironment{equation}
354   {\linenomath$$\refstepcounter{equation}}
355   {\eqno\hbox{\@eqnnum}$$\endlinenomath}
356
357 \let\LN@eqnarray\eqnarray
358 \let\LN@endeqnarray\endeqnarray
359
360 \renewenvironment{eqnarray}
361   {\linenomath\LN@eqnarray}
362   {\LN@endeqnarray\endlinenomath}
363
364 \fi
```

4 6.1.2 Line numbers in internal vertical mode

5 The command `\internallinenumbers` adds line numbers in internal vertical
6 mode, but with limitations: we assume fixed baseline skip.

```
365 \def\internallinenumbers{\setrunninglinenumbers
366   \let\@@par\internallinenumberspar
367   \ifx\@par\@@par\let\@par\internallinenumberspar\fi
368   \ifx\par\@@@par\let\par\internallinenumberspar\fi
369   \ifx\@par\linenumberspar\let\@par\internallinenumberspar\fi
370   \ifx\par\linenumberspar\let\par\internallinenumberspar\fi
371   \@ifnextchar[{\resetlinenumbers}]%
372     {\@ifstar{\let\c@linenumber\c@internallinenumbers
373               \c@linenumber\@ne}{}}%
374   }
375
376 \let\endinternallinenumbers\endlinenumbers
377 \@namedef{internallinenumbers*}{\internallinenumbers*}
378 \expandafter\let\csname endinternallinenumbers*\endcsname\endlinenumbers
379
380 \newcount\c@internallinenumbers
381 \newcount\c@internallinenumbers
382
383 \def\internallinenumberspar{\ifvmode\@@@par\else\ifinner\@@@par\else\@@@par
```

```

384 \begingroup
385 \c@internallinenumbers\prevgraf
386 \setbox\@tempboxa\hbox{\vbox{\makeinternalLinenumbers}}%
387 \dp\@tempboxa\prevdepth
388 \ht\@tempboxa\z@
389 \nobreak\vskip-\prevdepth
390 \nointerlineskip\box\@tempboxa
391 \endgroup
392 \fi\fi
393 }
394
395 \def\makeinternalLinenumbers{\ifnum\c@internallinenumbers>0\relax
396 \hbox to\z@{\makeLineNumber}\global\advance\c@linenumber\@ne
397 \advance\c@internallinenumbers\m@ne
398 \expandafter\makeinternalLinenumbers\fi
399 }

```

6.1.3 Line number references with offset

1

This extension to defines macros to refer to line numbers with a offset, e.g.,
to refer to a line which cannot be labeled directly (display math). This was
formerly knows as `rlineno.sty`.

2

3

4

To refer to a pagewise line number with offset:

5

```
\linerefp[⟨OFFSET⟩]{⟨LABEL⟩}
```

6

To refer to a running line number with offset:

7

```
\linerefr[⟨OFFSET⟩]{⟨LABEL⟩}
```

8

To refer to a line number labeled in the same mode as currently selected:

9

```
\lineref[⟨OFFSET⟩]{⟨LABEL⟩}
```

10

```

400 \newcommand\lineref{%
401 \ifx\c@linenumber\c@runninglinenumber
402 \expandafter\linerefr
403 \else
404 \expandafter\linerefp
405 \fi
406 }
407
408 \newcommand\linerefp[2][\z@]{%
409 \let\@thelinenumber\thelinenumber
410 \edef\thelinenumber{\advance\c@linenumber#1\relax\noexpand\@thelinenumber}%
411 \ref{#2}%
412 }}

```

1 This goes deep into L^AT_EX's internals.

```
413 \newcommand\linerefr[2][\z@]{%  
414   \def\@linerefadd{\advance\c@linenumber#1}%  
415   \expandafter\@setref\csname r@#2\endcsname  
416   \@linerefadd{#2}%  
417 }}  
418  
419 \newcommand\@linerefadd[2]{\c@linenumber=#1\@linerefadd\relax  
420                               \thelinenumber}
```

2 6.1.4 Numbered quotation environments

3 The `{numquote}` and `{numquotation}` environments are like `{quote}` and
4 `{quotation}`, except there will be line numbers.

5 An optional argument gives the number to count from. A star `*` (inside
6 or outside the closing `}`) prevent the reset of the line numbers. Default is to
7 count from one.

```
421 \newcommand\quotelinenumbers  
422   {\@ifstar\linenumbers{\ifnextchar[\linenumbers{\linenumbers*}}}  
423  
424 \newdimen\quotelinenumbersep  
425 \quotelinenumbersep=\linenumbersep  
426 \let\quotelinenumberfont\linenumberfont  
427  
428 \newcommand\numquotelist  
429   {\leftlinenumbers  
430     \linenumbersep\quotelinenumbersep  
431     \let\linenumberfont\quotelinenumberfont  
432     \addtolength{\linenumbersep}{-\@totalleftmargin}%  
433     \quotelinenumbers  
434   }  
435  
436 \newenvironment{numquote}   {\quote\numquotelist}\endquote}  
437 \newenvironment{numquotation} {\quotation\numquotelist}\enquotation}  
438 \newenvironment{numquote*}  {\quote\numquotelist*}\endquote}  
439 \newenvironment{numquotation*}{\quotation\numquotelist*}\enquotation}
```

8 6.1.5 Frame around a paragraph

9 The `{bframe}` environment draws a frame around some text, across page
10 breaks, if necessary.

11 This works only for plain text paragraphs, without special height lines.
12 All lines must be `\baselineskip` apart, no display math.

```

440 \newenvironment{bframe}
441   {\par
442     \@tempdima\textwidth
443     \advance\@tempdima 2\bframesep
444     \setbox\bframebox\hbox to\textwidth{%
445       \hskip-\bframesep
446       \vrule\@width\bframerule\@height\baselineskip\@depth\bframesep
447       \advance\@tempdima-2\bframerule
448       \hskip\@tempdima
449       \vrule\@width\bframerule\@height\baselineskip\@depth\bframesep
450       \hskip-\bframesep
451     }%
452     \hbox{\hskip-\bframesep
453           \vrule\@width\@tempdima\@height\bframerule\@depth\z@}%
454     \nointerlineskip
455     \copy\bframebox
456     \nobreak
457     \kern-\baselineskip
458     \runninglinenumbers
459     \def\makeLineNumber{\copy\bframebox\hss}%
460   }
461   {\par
462     \kern-\prevdepth
463     \kern\bframesep
464     \nointerlineskip
465     \@tempdima\textwidth
466     \advance\@tempdima 2\bframesep
467     \hbox{\hskip-\bframesep
468           \vrule\@width\@tempdima\@height\bframerule\@depth\z@}%
469   }
470
471 \newdimen\bframerule
472 \bframerule=\fboxrule
473
474 \newdimen\bframesep
475 \bframesep=\fboxsep
476
477 \newbox\bframebox

```

7 The final touch

1

There is one deadcycle for each line number.

2

```

478 \advance\maxdeadcycles 100
479
480 \endinput

```

1 8 The user commands

2 The user command to turn on and off line numbering are

3 `\linenumbers`

4 Turn on line numbering in the current mode.

5 `\linenumbers*`

6 and reset the line number to 1.

7 `\linenumbers[<number>]`

8 and start with *<number>*.

9 `\nolinenumbers`

10 Turn off line numbering.

11 `\runninglinenumbers*[<number>]`

12 Turn on *running* line numbers, with the same optional arguments as
13 `\linenumbers`. The numbers are running through the text over page-
14 breaks. When you turn numbering off and on again, the numbers will
15 continue, except, of course, if you ask to reset or preset the counter.

16 `\pagewiselinenumbers`

17 Turn on *pagewise* line numbers. The lines on each page are numbered
18 beginning with one at the first *pagewise* numbered line.

19 `\resetlinenumber[<number>]`

20 Reset [Set] the line number to 1 [*<number>*].

21 `\setrunninglinenumbers`

22 Switch to *running* line number mode. Do *not* turn it on or off.

23 `\setpagewiselinenumbers`

24 Switch to *pagewise* line number mode. Do *not* turn it on or off.

25 `\switchlinenumbers*`

26 Causes margin switching in *pagewise* modes. With the star, put the
27 line numbers on the inner margin.

28 `\leftlinenumbers*`

<code>\rightlinenumbers*</code>	1
Set the line numbers in the left/right margin. With the star this works for both modes of operation, without the star only for the currently selected mode.	2 3 4
<code>\runningpagewiselinenumbers</code>	5
When using the pagewise line number mode, do not subtract the page offset. This results in running line numbers again, but with the possibility to switch margins. Be careful when doing line number referencing, this mode status must be the same while setting the paragraph and during references.	6 7 8 9 10
<code>\realpagewiselinenumbers</code>	11
Reverses the effect of <code>\runningpagewiselinenumbers</code> .	12
<code>\modulolinenumbers[<i>number</i>]</code>	13
Give a number only to lines which are multiples of [<i>number</i>]. If <i>number</i> is not specified, the current value in the counter <code>linenumbermodulo</code> is retained. <i>number</i> =1 turns this off without changing <code>linenumbermodulo</code> . The counter is initialized to 5.	14 15 16 17
<code>\linenumberdisplaymath</code>	18
Number the lines of a display math in a <code>{linenomath}</code> environment, but do not in a <code>{linenomath*}</code> environment. This is used by the package option <code>[displaymath]</code> .	19 20 21
<code>\nolinenumberdisplaymath</code>	22
Do not Number the lines of a display math in a <code>{linenomath}</code> environment, but do in a <code>{linenomath*}</code> environment. This is the default.	23 24
<code>\linelabel</code>	25
Set a <code>\linelabel{foo}</code> to the line number where this commands is in. Refer to it with the L ^A T _E X referencing commands <code>\ref{foo}</code> and <code>\pageref{foo}</code> .	26 27 28
The commands can be used globally, locally within groups or as environments. It is important to know that they take action only when the <code>\par</code> is executed. The <code>\end{mode}linenumbers</code> commands provide a <code>\par</code> . Examples:	29 30 31 32

```

1      {\linenumbers <text> \par}
2
3      \begin{linenumbers}
4      <text>
5      \end{linenumbers}
6
7      <paragraph> {\linenumbers\par}
8
9      \linenumbers
10     <text> \par
11     \nolinenumbers
12
13     \linenumbers
14     <paragraph> {\nolinenumbers\par}

```

15 8.1 Customization hooks

16 There are several hooks to customize the appearance of the line numbers,
17 and some low level hooks for special effects.

18 `\thelinenumber`

19 This macro should give the representation of the line number in the
20 L^AT_EX-counter `linenumber`. The default is provided by L^AT_EX:

```
21     \arabic{linenumber}
```

22 `\makeLineNumberLeft`

23 This macro is used to attach a line number to the left of the text page.
24 This macro should fill an `\hbox` to `0pt` which will be placed at the
25 left margin of the page, with the reference point aligned to the line to
26 which it should give a number. Please use the macro `\LineNumber` to
27 refer to the line number.

28 The default definition is

```
29     \hss\linenumberfont\LineNumber\hskip\linenumbersep
```

30 `\makeLineNumberRight`

31 Like `\makeLineNumberLeft`, but for line numbers on the right margin.

32 The default definition is

```
33     \linenumberfont\hskip\linenumbersep\hskip\textwidth
34     \hbox to\linenumberwidth{\hss\LineNumber}\hss
```

<code>\linenumberfont</code>	1
This macro is initialized to	2
<code>\normalfont\tiny\sffamily</code>	3
<code>\linenumbersep</code>	4
This dimension register sets the separation of the linenumber to the text. Default value is 10pt.	5 6
<code>\linenumberwidth</code>	7
This dimension register sets the width of the line number box on the right margin. The distance of the right edge of the text to the right edge of the line number is <code>\linenumbersep + \linenumberwidth</code> . The default value is 10pt.	8 9 10 11
<code>\theLineNumber</code> (for wizards)	12
This macro is called for printing a <code>\newlabel</code> entry to the aux-file. Its definition depends on the mode. For running line numbers it's just <code>\thelinenumber</code> , while in pagewise mode, the page offset subtraction is done in here.	13 14 15 16
<code>\makeLineNumber</code> (for wizards)	17
This macro produces the line numbers. The definition depends on the mode. In the running line numbers mode it just expands <code>\makeLineNumberLeft</code> .	18 19 20
<code>\LineNumber</code> (for wizards)	21
This macro is called by <code>\makeLineNumber</code> to typeset the line number. This hook is changed by the modulo mechanism.	22 23