Experiments with `\parfillskip`

Udo Wermuth

Abstract

Plain TeX sets the glue parameter `\parfillskip` in such a way that any length between 1 sp and the full line width is accepted for the width of the material printed in the last line of a paragraph. In certain circumstances, typesetting tradition objects to a last line with text that has a width less than the indentation or to completely filled last lines.

This article analyzes different specifications for the `\parfillskip` glue based on experiments with twelve one-, two-, or three-line paragraphs supported by theoretical considerations. The analysis shows how TeX's line-breaking procedure acts on the last line of a paragraph and how it can run into problematic situations if `\parfillskip` has an injudicious specification. This might lead to ugly output.

1 Introduction

TeX has a handful of glue parameters that affect the typesetting of paragraphs. Two of these parameters, `\spaceskip` and `\xspaceskip`, replace the two `\fontdimen` values for the interword glue and the extra space [11, p. 76]. The spaces in the paragraph carry the glue characteristics that are specified by these two parameters if they are nonzero. Two other glue parameters change all the lines in a paragraph: `\leftskip` and `\rightskip` add glue to the left or right of every line [11, p. 100]. They are used, for example, to typeset a paragraph narrower, i.e., justified but indented on both sides, or in a shape that shows a straight margin only on one side. All four parameters have the value 0 pt in plain TeX.

The fifth glue parameter is special as it is usually applied in a single place in a paragraph: It is the `\parfillskip` glue, which has a direct effect only on the last line where it acts as an "additional `\rightskip`" [11, p. 274]. However, it can affect more than one line in a paragraph. If a paragraph is interrupted by display math mode the line before the display is treated like a last line of a paragraph, although the paragraph has not ended yet. TeX uses the value of `\parfillskip` that is current when it starts to break either a part or the whole paragraph into lines. So more than one specification of `\parfillskip` might be applied in a single paragraph. For example, the paragraph starts in a group in which `\parfillskip` is locally changed and after a displayed equation the group ends but more text follows. The last line before the display then uses a different `\parfillskip` than the end of the text.

TeX does several things when it has to build a last line: First either an infinite penalty is added or, if the paragraph has a glue item at its end, TeX changes this glue into an infinite penalty item. This penalty prevents a line break in front of the horizontal skip `\hskip\parfillskip` that is added by TeX to finish the paragraph [12, §816].

Plain TeX sets the value of `\parfillskip` to `0pt plus 1fil` [11, p. 100]. This specification gives stretchability to the last line so that it can contain text whose width is shorter than the line width. As it is a glue parameter that an author is allowed to manipulate any glue specification can be assigned to `\parfillskip`, for example, a natural width different from 0 pt, a nonzero shrinkability, or a stretchability of finite order. A change in the value has an impact on the line-breaking decisions made by TeX.

In the rest of the article the phrase "length of the last line" means "length of the material in the last line". Normally, the width of the last line is always `\hsize` (the command `\parshape` is not discussed in this article). So a "short last line" means that the width of the text in the last line can be called "short". And this word means in this article that the value of `\parindent` is larger or not much smaller than the width of that text.

Additional `\rightskip`. The glue `\parfillskip` is usually only applied to the last line of a paragraph. Therefore it can be used to get special effects for this line. For example, the assignments

```
\leftskip = 0pt plus 1fil
\rightskip = 0pt plus -1fil
\parfillskip = 0pt plus 2fil
```

sets the last line centered without affecting other lines. In all but the last line the `\rightskip` neutralizes the `\leftskip` so that there is a net contribution of 0 pt. But the last line has a `\leftskip` of `0pt plus 1fil` and at the right side the sum of `\rightskip` and `\parfillskip` which equals `0pt plus 1fil`, i.e., on both sides is the same amount of infinite stretchability and the line is centered in the output; see [4] (or [22]).

Of course, there are other ways to manipulate the last line. For example, the end of a paragraph can execute additional typesetting commands if the control sequence `\par` is redefined. Peter Wilson describes such methods in his columns *Glisterings* [22, 23, 24]. This article analyzes what happens if solely the glue specification of `\parfillskip` is changed and this analysis already fills quite a few pages.

Contents. First, the default setting of plain TeX for \parfillskip is discussed in section 2. Then in the next section the value of \parfillskip is set to 0 pt either for a complete document or a single paragraph. It presents also some effects that might occur if the input for a paragraph contains negative infinite stretchability.

In section 4, experiments with finite dimensions for the stretchability of \parfillskip are executed. First with a stretchability larger than \hsize, second with one that is a fraction of \hsize, and third with a negative finite stretchability. Section 5 adds some theoretical results. The next section checks what happens if \parfillskip has natural width besides stretchability, and section 7 presents the related theory. It also shows how to make use of the trace data written by \tracingparagraphs and how different values for the stretchability can be compared in a certain sense.

Specifications for \parfillskip that have natural width and shrinkability but no stretchability are the topic of section 8. Theoretical results about such settings are in section 9. In section 10 all three dimensions of the glue \parfillskip are changed to finite nonzero values.

Section 11 looks at a couple of specifications for \parfillskip based on the facts learned in the previous sections and compares them to some suggestions made by others. The last section provides a summary of the results.

2 Plain TeX's default 0pt plus 1fil

The default setting is useful as it works with any last line from normal text. The last line has either badness 0, i.e., it is decent, or it is tight with a badness as high as 100. Therefore, no loose or very loose lines are possible, i.e., glue never stretches. The default \parfillskip cannot be the reason for an overfull line and an underfull line can only appear if the last line is empty, e.g., if the paragraph faultily ends with a forced break entered by the author.

On the other hand, very short lines are possible; a hyphenated part of a word suffices to form a tolerated last line. (According to [6, 3.11], the last word of a paragraph should never be hyphenated.) Typesetting tradition recommends having last lines that are longer than the indentation of paragraphs if the start of a paragraph is identified by indentation (see [9, p. 142]), and that an indentation shall be at least 0.5 em. Values of 1 em and 1\baselineskip — the natural width of the \baselineskip — are recommended in [3, p. 40], and then [3, p. 42] demands at least four letters in the last line. Plain TeX indents by 20 pt, i.e., 2 em in cmr10. This is a very high value compared to the above recommendations.

A comment: A specification with the font related unit 1 em should be made after the font for the text was selected. TeX uses for the unit em the quad width of the font that is active when the specification of \parfillskip is processed. A switch to a smaller font or a different face in the text does not change \parfillskip.

Experiment 1: Description

Show that the last line of a paragraph can be shorter than the indentation if the plain TeX default values for \parindent and \parfillskip is used.

TeX input

```
1. Please answer if my topic is ``in'' or
``out''. \TeX: in
```

TeX output

• \parfillskip ← 0 pt plus 1fil (plain TeX's default):

1. Please answer if my topic is “in” or “out”. TeX: in ▯

In the section “TeX output” of an experiment a paragraph that starts with a bullet shows the specification of \parfillskip that is used in the following paragraphs either up to the end of the experiment or up to another line that starts with a bullet. The specification is written as a formula, not as a valid TeX assignment. The symbol ‘▯’ that is printed in the right margin marks the end of an experiment.

As mentioned above the spaces in the text of the last line are either perfect or they shrink.

Experiment 2: Description

Show that the last line of a paragraph can end at the right margin.

TeX input

```
2. Has the last line of this paragraph
badness~0 and has no interword space to
stretch? Do they shrink now?
```

TeX output

• \parfillskip ← 0 pt plus 1fil (plain TeX's default):

2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now? ▯

Of course, the first two experiments show simple things. They are presented mainly for comparison with later experiments that reuse the two texts. Notice that each text starts with a number that identifies the experiment where it was introduced with plain TeX's \parfillskip. The first experiment is minimal, as TeX has only one valid way to typeset the text. So no setting of \parfillskip can produce a different second line, obeying the \hsize. The last line of experiment 2 has badness 2 so its glue shrinks. It will be useful to have a variant of this experiment that has badness 100 in its last line and a tie for the last word.

Experiment 2 continued: TeX input

```
$2'$. Has the last line of this paragraph
badness~0 and has no interword space to
stretch?\kern.557pt!
Do they shrink~now?\kern.557pt!
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

  $2'$. Has the last line of this paragraph badness 0 and has no interword space to stretch?! Do they shrink now?! ▯

Here are the lines written by `\tracingparagraphs` to show that the last line is maximally tight. (The information is written to the transcript file of the run if `\tracingparagraphs` is set to 1; see [11], pp. 98–99, or [19] for a description of this data.)

Experiment 2 continued: Log file contents

```
 1. @firstpass
 2. []$\ninerm 2[]$. Has the last line
      of this paragraph badness 0
 3. @ via @@0 b=18 p=0 d=784
 4. @@1: line 1.1 t=784 -> @@0
 5. and
 6. @ via @@0 b=64 p=0 d=5476
 7. @@2: line 1.3 t=5476 -> @@0
 8. has no interword space to stretch?!
      Do they shrink now?!
 9. @\par via @@2 b=100 p=-10000 d=12100
10. @@3: line 2.3- t=17576 -> @@2                ▯
```

If the `\parindent` is 0 pt and if the `\parskip` does not separate paragraphs with a noticeable vertical glue item a clear indication of the end of the paragraph is missing. In [9, p. 143], white space of at least 1 em is recommended at the end of the last line of a paragraph with justified margins. Ragged-right text requires much larger white space at the end of the last line, which is of course dependent on the amount that the right margin changes.

German typesetters learned the old rule never to leave white space at the end of the last line that is less than 1 em wide if paragraphs are indented. In such a situation it was considered better to keep the right margin straight. Nowadays the rule is not recommended anymore [9, p. 142] so it is ignored in this article. (According to [5], or see [22], a similar rule with `\parindent` instead of 1 em existed in Russia.)

Thus we have two observations of typographic trouble that the default setting of `\parfillskip` can create in some situations.

O1 (Short line): The last line might be shorter than the `\parindent` and such an event leaves so much white space that it can look like an empty line between paragraphs.

O2 (Completely filled): The text fills the last line completely without a visual indication that it is the last line of the paragraph, especially, if the `\parindent` is 0 pt and the `\parskip` does not separate paragraphs by an empty line. White space of width 1 em should be at the end of the last line in such a setup.

The first observation is especially bad if vertical space is used to structure the text [9, p. 142].

The two observations can be formulated more formally as recommendations that should be obeyed:

$$\texttt{\textbackslash parindent} < \text{length of last line} \qquad (1)$$

$$\text{length of last line} \le \texttt{\textbackslash hsize} - 1\,\text{em}. \qquad (2)$$

These recommendations are extracted from different books by different experts. This is usually not a good approach as each expert has a unique collection of typographical rules and two such collections might contain conflicting rules. For example, page 21 of [3] ends with a first line of a paragraph; something that [6, 3.11] forbids. On page 43 of [3] R. Bringhurst explains why he accepts such lines.

The important point for the recommendations (1) and (2) is that they are not suggested to be held simultaneously. Therefore the two recommendations should be treated independently.

In this article, sometimes the text of an experiment is indented, often it starts flush left. Nevertheless every text is used for both recommendations as only the last lines count.

Displays. Display math mode in a paragraph, i.e., material between a pair of doubled dollar signs, can create a problem if the `\parfillskip` contains only finite dimensions. TeX determines the length of the line directly above such a display and depending on its width either `\abovedisplayshortskip` or the (usually wider) `\abovedisplayskip` is put between text and formula except if it is an *alignment display* [11, p. 190]. In this case TeX always applies the latter [12, §1206]. The short skip is used if, more or less, the end of the text and the start of the formula do not overlap and if no equation number is used on the left side [11, p. 189]. After the math material is typeset TeX adds either the glue `\belowdisplayskip` or `\belowdisplayshortskip` if there was a decision based on the length of the line before the display; otherwise `\belowdisplayskip` is used.

Here is how TeX determines which skip to use. As soon as two dollar signs occur TeX computes the dimension `\predisplaysize` that contains, as one summand, the width of the line before the display. The other two summands are the width by which the line is shifted, for example, if `\leftskip` is nonzero, and a font-related constant of 2 em. But there are two exceptions: If there is no previous line, i.e., the display starts the paragraph, `\predisplaysize` is set to −\maxdimen. If the previous line contains interword glue and that glue stretches or shrinks \predisplaysize is set to \maxdimen [11, p. 188; 12, §1148]. This is necessary to guarantee that TeX does not make machine-dependent decisions.

Discussion. Ok, in some use cases the default setting of plain TeX fails to meet certain recommendations. And this “failure” is not reported by TeX as a warning or an error message. An author who sees in the output an unwanted effect can consider rewriting the paragraph, but also, sometimes a sequence of commands at the end of the paragraph can be used to avoid such a situation.

Recommendation (1) can be supported by two simple rules during the input. First, use a tie in front of the last word if it has fewer than four letters. Second, put longer last words whose last fragment after hyphenation has fewer than four letters in an \hbox. If TeX is not able to typeset this paragraph it reports an overfull line error message and the author can fix the situation. The first rule is easy to follow if the author is used to applying ties as recommended by [11, pp. 91–93]. The second rule is harder to observe and it requires more typing.

People have suggested non-default settings for \parfillskip in order to obey the two mentioned recommendations better and without new rules for the input. For example, P. Taylor [18, p. 388] suggests using 0.7\hsize instead of 1 fil as the stretchability of \parfillskip to make (1) more likely. W. Schmidt [17] discusses the use of a nonzero natural width in a very special case: if \parindent is 0 pt, always end a paragraph with some white space, i.e., to support (2). He uses the glue 2em plus 1fil. And F. Mittelbach [16, p. 344] used natural width together with shrinkability to address both recommendations. Let $x :=$ \hsize $-$ 1.5\parindent and $y := x - 1$ em and then set \parfillskip to x plus 0pt minus y. This combination leaves white space at the end of the paragraph and as the natural width does not cover the whole line width, very short lines are assumed to be unlikely. Of course, it is possible to change all three dimensions of the glue specification. P. Wilson suggests in [24, p. 340] the specification 0.75\hsize plus 0.06\hsize minus 0.75\hsize; see also [1]. And in [10, p. 1156], Donald E. Knuth and Michael F. Plass discuss the effect of the parameter \looseness and write “The penalty for adjacent lines of contrasting classes seems to work best in connection with looseness if the finishing glue at the paragraph end is set to have a normal space equal to about half the total line width, stretching to nearly the full width and shrinking to zero.” (In [14, p. 194], the text was changed to “… a normal space equal to about one-third of the total line width, stretching to the full width and shrinking to zero.” I can only guess why the text was changed: more experience with both parameters. \parfillskip became a changeable parameter only a few months before the article was written, and at the same time \looseness was added; see entries 457 and 459 in [13, Ch. 11].) As mentioned before, this article only discusses changes to \parfillskip, not the results of varying two parameters at the same time.

The length of the last line seems to be computable via the abovementioned \predisplaysize if \parfillskip has its default setting. In example 2 of [16, p. 344], a macro is presented to measure the length of the last line based on the idea of adding display math at the end of a paragraph and outputting the \predisplaysize. As noticed by the author, some aspects that the display introduces, such as skips, can be reverted, but \tracingoutput will still show them as their occurrence is not deleted from TeX’s memory.

Summary. Short last lines cannot be prevented except by rewriting the text or forcing a manual line break, and at least one ugly looking line is accepted; see experiment 1. Using ties and boxes an author can help TeX avoid typesetting a short last line and output a warning if it cannot be avoided.

Completely (or nearly so) filled last lines cannot be completely prevented either except by the author rewriting the text or adding white space at the end of each paragraph using a tie and an empty \hbox, or forcing a manual line break, and accepting at least one non-perfect line in the paragraph. Or the author makes the simplest of all changes and gives the natural width of \parfillskip a nonnegative value. This last method is discussed in section 6 together with the problems that such a change produces.

There is no general need to use finite dimensions for the stretchability or to switch to shrinkability for \parfillskip as has been suggested. Nevertheless, it seems to be an interesting topic to study, whether such settings have applications and what happens if finite dimensions are used. On the other hand it does not help to have a higher order of infinite stretchability, i.e., 1 fill [11, p. 72]. It might be useful if \leftskip, \rightskip, or the text contains infinite stretchability of the first or second order but otherwise it has no effect different from 1 fil.

3 Zero \parfillskip

The assignment of 0 pt to \parfillskip forces TeX to finish the last line of the paragraph flush right if this line contains at least one interword space. This contradicts (2) but in some situations such a value makes sense. For example, when `\parshape` is used the last line should contain a zero `\parfillskip`.

This setting is also useful to split a very long paragraph into smaller parts to avoid memory overflow as stated in the answer to exercise 14.15 of *The TeXbook* [11]: Execute `\par` to end a paragraph and use the setting `\parfillskip=0pt`. To keep the distance of lines at `\baselineskip` the glue that TeX inserts between paragraphs must be set to 0 pt too. All parameter changes should be done in a group to avoid a global change. Therefore the complete answer is given as:

```
{\parfillskip=0pt\par\parskip=0pt\noindent}
```

But that works only if it is either entered at a place where TeX has found a valid line break before, or if enough text is available to allow TeX to break at almost any interword glue. Otherwise the spaces in the line that are typeset before the break are often extremely stretched.

Experiment 3: Description

Show that the interword glue of the last line can get extremely stretched.

TeX input

```
3. A short text in 1 line. Or has the paragraph
2 or 3 lines?
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):
  3. A short text in 1 line. Or has the paragraph 2 or 3 lines?
- `\parfillskip` ← 0 pt:
  3. A short text in 1 line. Or has the paragraph 2 or 3 lines? ▯

The experiment's text is typeset in the first pass if plain TeX's default `\parfillskip` is used. But TeX reports that the last line of the last paragraph is underfull with badness 10000. It was forced to execute a second pass to check, without success, if the paragraph can be typeset with lines whose badness do not exceed the current tolerance. Plain TeX sets the tolerance for the first pass to 100 and for the second to 200 [11, p. 96]. So, more observations:

O3 (Glue stretches): Interword glue of finite order stretches in the last line if all other glue has finite stretchability and if the sum of `\parfillskip`'s natural width and the width of the material is shorter than `\hsize`.

O4 (2nd pass forced): If the glue in every possible last line of the first pass has to stretch so much that the badness of this line must be larger than `\pretolerance` then TeX is forced to execute a second pass. And depending on the value of `\emergencystretch` and the result of the second pass a third pass might be executed.

O5 (Underfull line): The output paragraph has an underfull line if the glue in the typeset last line has to stretch so much that the badness of that line becomes larger than `\tolerance`. With the plain TeX value of 1000 for `\hbadness` this is not reported for all cases.

The observation O4 about the execution of a second pass can occur with the default setting of `\parfillskip` too. But then it is the *text* that cannot obey the limit `\pretolerance`. Using the plain TeX default setting the text of experiment 3 is typeset in the first pass. It is the changed `\parfillskip` that is responsible for the text of this experiment being typeset a second time in the second pass. So in this article it is said that the default value of `\parfillskip` never *forces* a second pass but it can happen, for example, with a `\parfillskip` of 0 pt.

In contrast to the other observations, O4 states a technical point and not a property of the new last line. Of course, TeX might output the paragraph with different line breaks and the last line that is typeset in the second pass might not have a badness larger than `\pretolerance`; actually all lines of the paragraph might have badness values that are less than or equal to that limit but then at least one hyphen was inserted in the text.

A comparison of both outputs in experiment 3 shows that with plain TeX's default settings three words are placed in the last line but only two when `\parfillskip` is 0 pt. This means that the second output contains a first line with a larger badness because with the plain TeX default settings TeX must minimize the badness of the first line in this experiment. TeX considers the break as forced. It assigns the so-called *artificial demerits* [12, §854], which `\tracingparagraphs` shows as `d=*` [12, §856]. Artificial demerits occur only in the final pass, i.e., usually the second or the third pass. When TeX falls back to artificial demerits for a line this line does not contribute to the total demerits of the paragraph: TeX does not calculate the line demerits, they are set to 0 [12, §855]; compare the `t`-values in lines 10 and 13 in the following trace.

Experiment 3 continued: Log file contents

```
1. @secondpass
2. []\ninerm 3. A short text in 1 line. Or
   has the para-graph
3. @ via @@0 b=25 p=0 d=1225
4. @@1: line 1.1 t=1225 -> @@0
5. 2
6. @ via @@0 b=2 p=0 d=144
7. @@2: line 1.2 t=144 -> @@0
```

```
or
@ via @@0 b=12 p=0 d=484
@@3: line 1.2 t=484 -> @@0
3 lines?
@\par via @@3 b=10000 p=-10000 d=*
@@4: line 2.0- t=484 -> @@3
```
□

O6 (No demerits): If TeX is not able to find a valid break for the last line of a paragraph it might classify the break as forced. Then TeX avoids calculating the line demerits and assigns to this line *artificial demerits* which count as 0.

Note however that the paragraph is output differently if `\looseness` is set to −1 because of the sequence in which feasible breakpoints are listed by TeX. An explanation is given in [19], p. 372.

Experiment 3 continued: TeX definitions

```
\looseness=-1
```

TeX output

- `\parfillskip` ← 0 pt:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines? □

An underfull line without demerits also occurs when the text of experiment 1 is typeset with a zero `\parfillskip`; the output is identical to the one shown in experiment 1. But with a very short word on the last line or more shrinkability in the penultimate line a short last line might disappear.

Experiment 4: Description

Show that a short last line might be absorbed by the penultimate line.

TeX input

```
\noindent 4. My keyboArd is broken. When I
press the key for the lowercAse A the screen
repeAts it severAl times: a a a a
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a
a

- `\parfillskip` ← 0 pt:

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a □

O7 (Remove short line): To avoid infinite or high badness values TeX might dissolve the original last line and move its material into the formerly penultimate line.

The last line is also absorbed in a second pass even if `\finalhyphendemerits` get applied.

Experiment 4 continued: TeX input

```
\noindent $4'$. My keyboArd is broken; when I
press an `A' in lowercAse (only) the screen
repeAts it four times: a a a a
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

$4'$. My keyboArd is broken; when I press an 'A' in lowercAse (only) the screen repeAts it four times: a a a
a

- `\parfillskip` ← 0 pt:

$4'$. My keyboArd is broken; when I press an 'A' in lowercAse (only) the screen repeAts it four times: a a a a □

A last line may also be extended, that is, it receives some material from the penultimate line.

Experiment 5: Description

Show that a last line might be extended.

TeX input

```
\noindent 5. With enough interword glue as well
as short words at the end of the 1st line the
2nd can be extended.
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

5. With enough interword glue as well as short words at the end of the 1st line the 2nd can be extended.

- `\parfillskip` ← 0 pt:

5. With enough interword glue as well as short words at the end of the 1st line the 2nd can be extended. □

The shifting of "at" from the penultimate line into the last line allows TeX to typeset the second paragraph in the second pass. Its last line has the fitness class *very loose* and the first line stays in the class *decent*, so `\adjdemerits` are applied to the last line.

O8 (Extend last line): To avoid infinite or high badness values TeX might pack more material into the original last line to reduce its badness.

Changing `\parfillskip` once. The general solution to assign 0 pt to `\parfillskip` for a single paragraph is to enter

```
\hskip-\parfillskip\ \par
```

at its end. This technique can be used in situations where

`\hsize` − 1 em < length of last line < `\hsize`

to obey the abovementioned (outdated) tradition when (2) is violated.

Note the control space in front of the `\par` to avoid the glue from our `\hskip-\parfillskip`, that would otherwise appear at the end of the paragraph, from being removed by TeX as described above. Note also that `\par` is used here to signal the end of the paragraph but an empty line does the job too; the paragraph must only end after the control space. And finally, note that the bare minus sign is allowed but not `-1`, as any factor in front of the glue coerces it into a dimension, that is, only the natural width remains (see exercise 24.3 in [11]).

Negative infinite stretchability. TEX contains a primitive command `\hfilneg` that provides a horizontal skip with a stretchability of −1 fil. It cancels the stretchability of a skip with 1 fil when it is used in the same line. Therefore the above general solution to zeroize the default `\parfillskip` for a single paragraph can be formulated shorter as

```
\hfilneg\ \par
```

(compare to exercise 14.13 of [11]).

A digression. The use of negative infinite stretchability *in* a text — not just to cancel `\parfillskip`'s stretchability — affects the list of break candidates for the last line in probably unexpected ways.

Experiment X: Description

Show that negative infinite stretchability in the text can make TEX typeset fewer or more lines if `\parfillskip` is 0 pt.

TEX input

```
Xa. There’s no need to set interword space by
hand: a \penalty10000\hskip3.33333pt plus -1fil
b. And be careful with negative stretchability!
```

TEX output

• `\parfillskip` ← 0 pt plus 1fil (plain TEX's default):

Xa. There's no need to set interword space by hand:
a   b. And be careful with negative
stretchability!

• `\parfillskip` ← 0 pt:

Xa. There's no need to set interword space by hand:
a   b. And be careful with negative stretchability! ▯

TEX reports for the three-line paragraph 9804 demerits; 100 demerits more than for the two-line solution. A line with positive or negative infinite stretchability receives badness 0 [12, §852] if the glue does not need to shrink.

It is possible to construct an experiment that produces a longer paragraph when the stretchability of `\parfillskip` is set to 0 pt. But as in experiment Xa it contains some tricks.

Experiment X continued: TEX definitions

```
\def\X#1(#2)[#3+#4-#5]{% repeat a letter
 \penalty10000
 \xleaders\hbox to #1pt{\hss #2\hss}%
   \hskip#3\hsize plus #4fil minus #5\hsize
 \penalty10000 }
```

TEX input

```
\noindent Xb. \hbox{Myy}\X4.9(y)[0.16+1-0.15]\
     \hbox{old electrical}\penalty5\
\hbox{typewriter see}\X3.9(e)[0.14+0-0.1]%
     \hbox spread 1.5pt{ms to}
\hbox{hav}\X5(v)[0.13+1-0.13]\hbox{e a bad}
\hbox{bug. It repeats letters in mh}%
     \X4.7(h)[0.185+1-0]\null\penalty5\
\vphantom{h}\X4.7(h)[0.18+{-1}-0.5]!
\hbox{an unpredictable way \dots\ !}
```

TEX output

• `\parfillskip` ← 0 pt plus 1fil (plain TEX's default):

Xb. Myyyyyyy old electrical typewriter seeeeeeeeems to
havvvve a bad bug. It repeats letters in mhhhhhhhhh!
an unpredictable way . . . !

• `\parfillskip` ← 0 pt:

Xb. Myyyyyyyyyyyyyyyyyyyyyyyyyyyyyy old electrical
typewriter seeeeeeeeeeems to havvvvvvvvvvvvvvvve a bad
bug. It repeats letters in mhhhhhhhhhhhhhhhhhhhhhhhhh
hhhhhhhhhhhhhhhhhhhhhhhh! an unpredictable way . . . ! ▯

Although not the topic of this section a positive natural width can be used with experiment Xb to show a typographical disaster. The text sticks out into the left margin!

Experiment X continued: TEX output

• `\parfillskip` ← 120 pt:

Xb. Myyyyyyyyyyyyyyyyyyyyyyyyyyyyyy old electrical
typewriter seeeeeeeeeeems to havvvvvvvvvvvvvvvve a bad
bug. It repeats letters in mhhhhhhhhhhhhhhhhhhhhhhhhh
! an unpredictable way . . . ! ▯

And here is another anomaly: The paragraph might have the same last line as with the default setting but the number of lines is larger.

Experiment X continued: TEX definitions

```
\looseness=2
```

TEX output

• `\parfillskip` ← 0 pt plus 1fil (plain TEX's default):

Xb. Myyyyyyyyyyyyyyyyyyyyyyyyyyyyyy old electrical
typewriter seeeeeeeeeeems to havvvvvvvvvvvvvvvve a bad
bug. It repeats letters in mhhhhhhhhhhhhhhhhhhhhhhhhh
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh!
an unpredictable way . . . ! ▯

This new paragraph has 550 total demerits instead of 525. No setting of `\parfillskip` alone will force TEX to typeset these five lines.

In the rest of this article only "normal" texts are considered. No text will contain spaces with infinite stretchability, negative dimensions, or a shrinkability larger than the natural width. Of course, in the specification of `\parfillskip` such settings will still be considered.

Discussion. A couple of observations were made in this section: Some are bad, some neutral, some are typographical, other just technical. An underfull line with extremely stretched glue is typographically bad. A forced second pass is primarily technical, although now hyphenation and wider spaces are possible. Normally, a second pass should only be triggered either by content or by a command entered by the author, not unpredictably because of the setting of a parameter. But in this article the influence of `\parfillskip` is studied more or less independently of other aspects. Therefore settings that force a second pass are considered too.

Table 1: Observations on zero glue

	Specification of `\parfillskip`		
	natural width	0pt	0pt
	stretch	1fil	0pt
	shrink	0pt	0pt
Observation			
1	Short line	1	(1)
2	Completely filled	2	+ if the line contains glue
3	Glue stretches	−	3
4	2nd pass forced	−	3
5	Underfull line	−	3
6	No demerits	−	3
7	Remove short line	−	4
8	Extend last line	−	5

Legend: −/ /+: never/don't care/always
n/(*n*): (implicitly) shown in example number *n*

Table 1 summarizes the observations and lists an experiment in which the observation occurs. The reason for the observation must be the setting of `\parfillskip`. The header lines state the specification of `\parfillskip` that was used in the experiments. If the specification in an experiment differs from those given in the headlines but the observation is the same, the number of the experiment is placed in parentheses. For instance, this is done for experiment 1 with the observation "Short line". The output is identical to the result shown in experiment 1 but the second line has badness 10000 and artificial demerits. For experiment 3 these facts have made a completely different and ugly output, for experiment 1 there is no visible indication of a problem for TEX. Nevertheless because of such cases the *always* in the last column of Table 1 needs a comment.

A zero `\parfillskip` can be used to achieve an aesthetic effect for a whole document, as in [7]: All paragraphs end flush right and they are separated by an empty line. This layout is part of a general page design approach [8]. Such a strict requirement must have created a lot of work and the willingness to rephrase the text so that bad things do not happen.

One of the bad things with `\parfillskip=0pt` is the occurrence of artificial demerits. TEX does not show them as an error message, only the underfull line is reported as a warning. The specification of `\parfillskip` should not be the reason for their appearance. Again, artificial demerits can also occur with plain TEX's default settings. But in such a case it is not the specification of `\parfillskip` that causes the problem; it is the content itself. But then the output might not have a visual problem.

The all-zero specification might leave too much white space at the beginning of math displays if a line in which the glue stretches precedes a short centered formula without an equation label at its left.

If the problem occurs it can be fixed by setting `\abovedisplayskip` and `\belowdisplayskip` to their `\...shortskip` variants directly after the two dollar signs. This change is done inside a group so the old values are restored after the closing `$$`. Or all paragraphs that contain displayed material start preemptively within a group, in which the default `\parfillskip` is active and which ends after the last display. But the simplest input rule is to enter always, for example, `\hfil\ $$` (with a control space) instead of the opening `$$`.

It should be noted that `\parfillskip` is relevant for line breaking and therefore the number of lines is affected by its specification. Of course, it is not directly a problem that paragraphs get shorter or longer if `\parfillskip` is changed. But in the first case recommendation (2) might be violated and in the second case it is recommendation (1) that might not be obeyed.

Summary. The value 0 pt for `\parfillskip` supports (1) if possible but might fail spectacularly if the last lines contains interword glue. Recommendation (2) is only obeyed with underfull last lines that contain no glue. In general, this setting might be used in certain circumstances to show a special effect but it needs too much care and work to be applied for a longer document.

4 Using finite stretchability

Of course, the stretchability of 1 fil in the default setting of `\parfillskip` is not needed to fill the last line; simply `\hsize` has enough stretchability. But this setting does not have the same properties as infinite stretchability. One remarkable difference was already mentioned above: the spacing around display math.

Finite stretchability: $\nu\times$ `\hsize`. As mentioned earlier: With the default setting of `\parfillskip`, TEX creates, under normal conditions (e.g., a last line that is not empty), either a last line with badness 0 or a tight last line with badness up to 100. When the input

```
\noindent\hbox to 1sp{\hss}          (*)
```

is used with finite stretchability the maximum difference for the badness of a non-tight line is reported for *TUGboat*'s column width, namely, 225 pt. With the definition `\parfillskip=0pt plus` ν`\hsize` the following values are found:

$\nu =$	1	2	3	4	5	6
badness for input $(*) =$	100	12	4	2	1	0

Thus, there is no difference between a stretchability of 1 fil and a stretchability of 6`\hsize` in the line breaks if `\leftskip`, `\rightskip`, and the input for the paragraph do not contain infinite glue. Note, a zero badness does not imply that the interword glue has its natural width. The setting 2`\hsize` still creates a last line of fitness class *decent*, that is, no `\adjdemerits` can be charged to the last line if it is not already charged using infinite stretchability.

One argument for using a finite stretchability is to make (1) more likely; that is, for a short last line help O8 "Extend last line" to occur. The idea is: With finite stretchability a higher badness is assigned to short lines as a lot of stretchability must be used and TeX has a reason to select line breaks putting more material into the last line in order to reduce its badness value. So short lines should be less likely.

The texts of experiments 1 and 2 produce the identical output as before with TeX's default setting for all ν so they are not shown here. The biggest change of badness values is seen for the step from 2`\hsize` to 1`\hsize`. With these values it should be easiest to find an experiment that produces a longer last line.

Experiment 6: Description

Show a noticeable difference between the finite stretchabilities of 2`\hsize` and 1`\hsize`.

TeX input

```
\noindent 6. One line or two for this text?
That is the question, or?
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

```
6. One line or two for this text? That is the question,
or?
```

- `\parfillskip` ← 0 pt plus 2`\hsize`:

```
6. One line or two for this text? That is the question,
or?
```

- `\parfillskip` ← 0 pt plus 1`\hsize`:

```
6. One line or two for this text? That is the question, or?
```
□

The single line, a tight line, has badness 65 and 5625 demerits. A second line with the stretchability of `\hsize` would receive badness 84, demerits 8836. Finite stretchability might avoid a short last line and produce a paragraph that has one line less. That line ends flush right violating (2). This effect was seen before in O7 "Remove short line".

For this observation the badness of the last line might even be smaller. As seen above, the badness of the last line might become 1 if the stretchability of `\parfillskip` is set to 5`\hsize`. This difference is important enough to produce other line breaks than the default setting.

Experiment 7: Description

Show that the badness 1 for the last line can be a reason to make a paragraph shorter.

TeX output

- `\parfillskip` ← 0 pt plus 6`\hsize`:

```
4. My keyboArd is broken. When I press the key for the
lowercAse A the screen repeAts it severAl times: a a a
a
```

- `\parfillskip` ← 0 pt plus 5`\hsize`:

```
4. My keyboArd is broken. When I press the key for the
lowercAse A the screen repeAts it severAl times: a a a a
```
□

The badness and the demerits for the three lines are: 10/400, 1/121, and 0/100; the *path demerits* [20], i.e., the sum of the line demerits, is therefore $400 + 121 + 100 = 621$. In the paragraph with two lines the values are 10/400 and 5/225 with path demerits of 625. In this case a third line would get 1/121 instead of 0/100. Then the path demerits are 642, so this path is less attractive for TeX.

But of course, the original idea for longer lines can be shown too. In the next experiment the two-lines paragraph's first line is decent and with finite stretchability it stays decent but the treatment of glue changes.

Experiment 8: Description

Show that `\parfillskip=0pt plus \hsize` might create a longer last line.

TeX input

```
\noindent 8. One line or two for this text?
That's the question, or not?
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

```
8. One line or two for this text? That's the question, or
not?
```

- `\parfillskip` ← 0 pt plus `\hsize`:

```
8. One line or two for this text? That's the question,
or not?
```
□

The badness values of the two first lines are 1 and 8, resp. This increase is compensated for in the case with a stretchability of `\hsize` by the decrease of the badness in the second line from 78 for the single word to 64 with two words. Of course, the interword glue in the last line of the second paragraph has to stretch. And again the badness value 1 is sufficient to make a difference.

Experiment 9: Description

Show that the badness 1 for the last line can be a reason to extend it.

TeX input

```
\noindent 9.\ Give me 5! As a factor for
width; to stretch! I need a five (5)!
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

```
9. Give me 5! As a factor for width; to stretch! I need a
five (5)!
```

- `\parfillskip` ← 0 pt plus 5`\hsize`:

```
9. Give me 5! As a factor for width; to stretch! I need
a five (5)!
```
□

Finite stretchability: $(\nu/10)\times$`\hsize`. The finite stretchability might be less than `\hsize` to support (1) even more: Now the white space provided by the stretchability of `\parfillskip` cannot cover the whole line width. That means that a short last line represents an underfull line and TeX will create such a line only if there is no other way to break the text.

In this case a successful first pass cannot be guaranteed. Let's check what badness β is produced with input $(*)$ if the stretchability is reduced to $\nu/10$ of `\hsize`:

$\nu =$	3	4	5	6	7	8	9
$\beta =$	3701	1558	800	463	291	195	137

The values for $\nu = 1$ and $\nu = 2$ result in an infinite badness of 10000. Stretchability lower than `\hsize` might force TeX to execute a second pass; see O4. Values greater than or equal to 0.8`\hsize` obey plain TeX's `\tolerance`. Otherwise underfull lines as described in O5 might be produced. With the plain TeX value of 1000 for `\hbadness` [11, p. 29] this is only reported for a stretchability somewhat less than 0.5`\hsize`; for exact numbers see section 5.

But underfull lines are not the only problem with small ν; artificial demerits might occur.

Experiment 10: Description
Show that a smaller fraction for the stretchability can make the last line of a paragraph shorter again.

TeX output

- `\parfillskip` ← 0 pt plus 0.7`\hsize`:

8. One line or two for this text? That's the question, or not?

- `\parfillskip` ← 0 pt plus 0.5`\hsize`:

8. One line or two for this text? That's the question, or not?

A stretchability of 0.7`\hsize` can make the last line of a paragraph longer as it was shown with a stretchability of `\hsize`, but the stretchability of 0.5`\hsize` fails in this experiment as artificial demerits are reported by TeX. The last line must become nearly 40% filled with such a small stretchability to get a badness value less than 200 (and then the change is unnecessary to avoid a short last line).

Experiment 11: Description
Show that a smaller fraction for the stretchability can make the last line of a paragraph longer.

TeX input

```
11. Sure, this text needs always two lines with
the current line width.
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

11. Sure, this text needs always two lines with the current line width.

- `\parfillskip` ← 0 pt plus 0.7`\hsize`:

11. Sure, this text needs always two lines with the current line width.

- `\parfillskip` ← 0 pt plus 0.5`\hsize`:

11. Sure, this text needs always two lines with the current line width.

TeX needs a second pass only in the third case. To extend the last line a high price in demerits must be paid; the total demerits are, in sequence: 200, 8200, and 51410.

Negative finite stretchability. There is an important difference if the negative stretchability is infinite or finite [15]. The first case produces last lines of badness 0 as discussed in section 3. But if the stretchability for the calculation of the badness is finite and negative the badness of the last line is set to 10000 [12, §852, §108] and artificial demerits are applied. As the line demerits for the last line are not calculated they do not influence TeX's line breaking decisions.

Experiment 12: Description
Show the difference between positive and negative finite stretchability.

TeX output

- `\parfillskip` ← 0 pt plus 0.7`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?

- `\parfillskip` ← 0 pt plus 0.5`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?

- `\parfillskip` ← 0 pt plus −0.7`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3lines?

The finite stretchability of 0.7`\hsize` makes the last line longer, but the negative stretchability shortens it. The first paragraph has 19994 demerits, the third only 484 as the last line does not count because of artificial demerits. As seen before, TeX picks a line break that does not even minimize the line demerits of the first line and it produces an underfull last line. The paragraph with stretchability 0.5`\hsize` has the same problem. When the interword spaces of the last lines for these two paragraphs are closely inspected then the space in the last one seems to be quite small.

Let's execute a test similar to a situation of an answer to a multiple choice question.

Experiment 13: Description
Show that a space vanishes with negative stretchability.

TeX input

```
\noindent 13. a
```

TeX output

- `\parfillskip` ← 0 pt plus 1fil (plain TeX's default):

13. a

- `\parfillskip` ← 0 pt plus −0.7`\hsize`:

13a ▯

The space between “13.” and “a” disappears, overwriting the period. The line is underfull and the log file shows that the stretchability of the space erodes the natural width.

Experiment 13 continued: Log file contents

```
\hbox(5.79999+0.0)x225.0, glue set -1.33734
.\ninerm 1
.\ninerm 3
.\ninerm .
.\glue 4.11108 plus 4.62497 minus 0.34259
.\ninerm a
.\penalty 10000
.\glue(\parfillskip) 0.0 plus -157.49931
.\glue(\rightskip) 0.0
```
▯

Note the value after `glue set` is negative. It is not a minus separated by a space from the number which is used to signal shrinkability [11, p. 79]. Therefore the “a” is moved $1.33734 \times 4.62497\,\text{pt} > 4.11108\,\text{pt}$ to the left.

O9 (Backspaces): Negative finite stretchability in `\parfillskip` induces negative stretchability for finite interword glue and might create spaces with negative width in the last line.

Discussion. Table 2 shows which experiments document the observations. The stretchability 6`\hsize` cannot be distinguished from 1 fil if no infinite glue appears in the paragraph. But the interword glue in a line with spaces in front of a display stretches so that the vertical spacing is wrong.

The reason for parentheses in the last column of Table 2 is the non-specific stretch value; therefore experiment 13, which uses a concrete value, appears in parentheses. The output is so bad that other observations are not analyzed for these specifications so a lot of entries are empty.

Table 2: Observations for finite stretchability

Specification of `\parfillskip` ($h \equiv$ `\hsize`)

	natural width	0pt	0pt	0pt	0pt	0pt	0pt	0pt	0pt
	stretch	1fil	6h	5h	h	.8h	.7h	.5h	<0pt
	shrink	0pt	0pt	0pt	0pt	0pt	0pt	0pt	0pt
Observation									
1	Short line	1	7	(1)	(1)	(1)	(1)	(1)	
2	Completely filled	2	(2)	7	6	(2)	(2)	(2)	
3	Glue stretches	–	–	(13)	8	(8)	10	11	
4	2nd pass forced	–	–	–	–	(1)	(1)	11	+
5	Underfull line	–	–	–	–	–	(1)	12	+
6	No demerits	–	–	–	–	–	(1)	12	+
7	Remove short line	–	–	7	6	(6)	(6)	(6)	
8	Extend last line	–	–	9	8	(8)	12	11	
9	Backspaces	–	–	–	–	–	–	–	(13)

Legend: −/ /+: never/don’t care/always
n/(n): (implicitly) shown in example number n

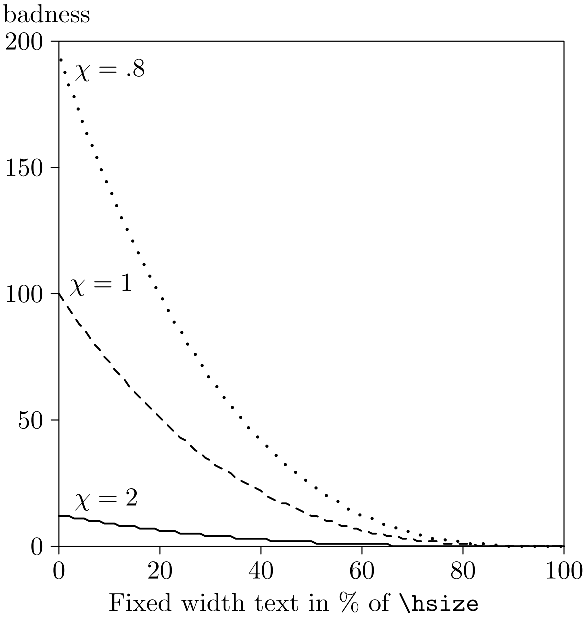

Figure 1: Graphs for $\chi = 0.8$, $\chi = 1$, and $\chi = 2$ when `\parfillskip` $\equiv$ `0pt plus` χ`\hsize`

One reason to select a finite stretchability for `\parfillskip` is to avoid a short last line. Experiments show that a smaller value for the stretchability might create longer last lines. Nevertheless, TeX might also respond with a paragraph that is not changed at all or becomes one line shorter compared to the number of lines with the default setting for `\parfillskip`. Such a paragraph then contains a completely filled last line, i.e., the line is not short.

Figure 1 shows how much badness a last line gets if the text has no additional stretchability. With a stretchability of 0.8`\hsize` a line must cover at least 60% of the line width to become decent (so that spaces do not stretch more than 50% [11, p. 97]).

The smaller the stretchability, the more likely that longer last lines are produced. But too small means artificial demerits can occur and then the last line is not improved at all. A stretchability like `\hsize` − `\parindent` seems to be an ideal choice as it forces the second pass only if the last line with the default setting of `\parfillskip` is shorter than the indent. But a smaller value might pull more strongly, so 0.8`\hsize` seems to be a better value in this respect. A second pass is avoided if the stretchability is at least `\hsize`.

Summary. To support (1) a second pass might be acceptable if the length of the last line is shorter than the indent. A stretchability of `\hsize` or a large fraction of `\hsize` seems to be the best choice although they also tend to remove the original last line and create paragraphs with one line less. Recommendation (2) is violated more often as very short lines are sometimes absorbed by the former penultimate line which then gets completely filled.

5 Theory: Finite stretchability

First, a few words about the notation: Lowercase math Latin letters stand for dimensions; especially, h is used for `\hsize` and p for `\parindent`; bold face Latin represents glue that consists of three dimensions: the natural width, the stretchability, and the shrinkability. For example, $\mathbf{p}$ is the `\parfillskip` and its three dimensions are written with a Latin p and a superscript: p°, p^+, and p^-. Lowercase Greek letters denote numbers, integers, or rational numbers. Several of them are reserved for often used integers: β is a badness, π a penalty, δ additional demerits, λ the `\linepenalty`, and τ the `\tolerance`. The uppercase Greek letter Λ stands for line demerits and Λ_p for path demerits, i.e., the sum of all line demerits of a set of valid line breaks.

The badness of input $(*)$ was found by runs with *TUGboat*'s column width. This must not be determined by an experiment though. The badness of lines in which the text occupies only 1 sp is computed as an approximation by the following formula (see [11], p. 97), in which the numerator represents the stretchability used to typeset the line and the denominator stands for the available stretchability in the line:

$$\beta \approx 100\Bigl(\frac{\text{used stretchability}}{\text{available stretchability}}\Bigr)^3.$$

Therefore

$$\beta \approx 100\Bigl(\frac{h - 1\,\mathrm{sp}}{\nu h}\Bigr)^3 \le 100\Bigl(\frac{h}{\nu h}\Bigr)^3 = \frac{100}{\nu^3}.$$

This computes the following values rounded to one decimal place, or two if that digit is 5 or would be rounded to 5:

$\nu =$	1	2	3	4	5	6
$100/\nu^3 \approx$	100	12.50	3.7	1.56	0.8	0.46

so that the above found badness values for $(*)$ seem to be the result of rounding the value $100/\nu^3$ to the nearest integer (and a tie-break rule to get 12).

This proves also that the results about 6`\hsize`, i.e., equality to the default line breaks if no infinite stretchability is in the paragraph, and 2`\hsize`, i.e., only decent last lines occur, are valid for all text widths.

The approximation can be used to find the factor ν that will never produce an underfull line. So the question is: For which ν does the badness reach $\tau = 200$? That is, find ν with

$$100\Bigl(\frac{h - 1\,\mathrm{sp}}{\nu h}\Bigr)^3 = \tau.$$

The 1 sp is very small compared to h; so it can be dropped. (Actually it is sufficient to find a value that *rounds down* to τ, but this difference is small.) Thus

$$1/\nu^3 \approx 2 \quad\Longrightarrow\quad \nu \approx 1/\sqrt[3]{2} \approx 0.79.$$

Well, that is not a big surprise as it was shown above that $0.8h$ creates a last line with badness 195. If `\hbadness` instead of `\tolerance` is used in the computation, ν must be larger than $1/\sqrt[3]{10} \approx 0.46$. This is the value after which plain TeX reports an underfull line; again an expected result.

Okay, something more complicated: Is it possible to identify a short last line through its badness value? Figure 1 shows that the badness falls very fast when the line is not filled with much text. So it should be possible to distinguish the case if the width of the text is less than the `\parindent` p; maybe not exactly but close.

The problem is to find ν such that, say,

$$100\Bigl(\frac{h - p}{\nu h}\Bigr)^3 = \beta + 1 \quad\text{and}\quad 100\Bigl(\frac{h - 1.1p}{\nu h}\Bigr)^3 = \beta.$$

The badness β is not important and gets replaced in the first formula by the second:

$$100\Bigl(\frac{h - p}{\nu h}\Bigr)^3 - 100\Bigl(\frac{h - 1.1p}{\nu h}\Bigr)^3 = 1$$

$$\Longleftrightarrow \qquad \frac{100}{h^3}\Bigl((h-p)^3 - (h-1.1p)^3\Bigr) = \nu^3$$

$$\Longleftrightarrow \qquad \frac{100}{h^3}\Bigl(0.3h^2p - .63hp^2 + 0.331p^3\Bigr) = \nu^3$$

$$\Longrightarrow \qquad \sqrt[3]{30\frac{p}{h} - 63\frac{p^2}{h^2} + 33.1\frac{p^3}{h^3}} = \nu.$$

For the *TUGboat* format $p = 20\,\mathrm{pt}$ and $h = 225\,\mathrm{pt}$ so that $p/h = 20/225 \approx 0.0889$, and thus

$$\nu = \sqrt[3]{2.667 - 0.4979 + 0.0233} \approx 1.29.$$

With $\nu = 1.29$ the available stretchability gets $\nu h = 290.25\,\mathrm{pt}$ and the badness values are 35 and 34 as

$$100\Bigl(\frac{205}{290.25}\Bigr)^3 = 35.23 \quad\text{and}\quad 100\Bigl(\frac{203}{290.25}\Bigr)^3 = 34.21.$$

So a short last line with a badness higher than 34 is not longer than the indentation. Of course, the phrase "short last line" is important as completely filled lines can have badness values larger than 34 when they have to shrink a lot. Values larger than 47 are assigned only to tight lines, as $100/1.29^3 \approx 47$.

The result $\nu = 1.29$ is based on the values of h, p, the factor 1.1, and the difference of 1 for the badness values. But both the factor and the difference can be changed and other specifications for p^+ are possible.

The specification with a stretchability using the factor 5 for `\hsize` has badness 1 up to 14% of the `\hsize`, so it is an ideal base for a division of short last lines and longer last lines. For longer last lines the badness is 0 so there is minimal influence on the glue. Smaller last lines have badness 1 and the glue does not stretch very much. Nevertheless the important point is that a short last line for a given length can be identified by the badness if the stretchability is a little bit arranged. For example, using the *TUGboat* format short last lines are distinguished from those that are wider than ≈ 25.250527 pt, which is sufficiently larger than 20 pt, i.e., the `\parindent`, if the stretchability of `\parfillskip` is 5.17`\hsize`.

Figure 2 presents a complete set of macros to check the badness of last lines for the *TUGboat* format; `\parfillskip` is only changed at the end of paragraphs so displays are not affected. False hits for tight lines (or short lines above a heading) are possible. The author has then the option to enter `\lastlineaccepted` at the end of the paragraph to suppress the error message in the subsequent runs.

6 Adding natural width to `\parfillskip`

Recommendation (2) is always obeyed if the natural width of `\parfillskip` is at least 1 em or 10 pt. The text of experiment 1 is unaffected, but experiment 2 changes with or without stretchability.

Experiment 14: Description
Show that natural width for `\parfillskip` might produce short last lines.

TEX output

- `\parfillskip` ← 10 pt plus 1fil:

  2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?

- `\parfillskip` ← 10 pt:

  2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now? □

Of course, the new last line in the last paragraph is underfull and has artificial demerits.

O10 (Add short line): The `\parfillskip` value might create a situation in which TEX prefers or is forced to typeset a new last line. This line might be short.

The natural width must be less than `\hsize` for normal text, or all last lines will be wider than `\hsize` and the shrinkability of the material in the last line must be large enough to avoid an overfull line. Otherwise only last lines that contain material of width greater than the difference of `\hsize` and the natural width shrink or become overfull.

```
% ASSUMPTION: \par has the TeX default
% NOTE: if active a par's end resets \parfillskip
\newif\ifCSLLtrace      % true: show message about
 \CSLLtracetrue   % start and stop on the terminal
\newif\ifCSLLactive      % true: checks are active
\newif\ifCSLLaccepted % true: user said line is OK
\newskip\CSLLpfs    % register to save parfillskip
\newlinechar=`\^^J%        newline in help message
\newhelp\CSLLhelp{checkshortlastline reports a
 warning message: The length^^Jof the line that
 ends at the line number shown in the^^Jlast line
 of the message might be shorter or not
 much^^Jlonger than the indentation or the last
 line is completely^^Jfilled and a little bit
 tight. Then it is a false hit.^^JUse
 \string\lastlineaccepted\space if you can accept
 the line.}
% output message (depending on \CSLLtrace setting)
\def\CSLLmsg(#1){% #1=0/1/2/3: off/on/is off/is on
 \immediate\write\ifCSLLtrace 16 \else -1 \fi
 {>>> \string\checkshortlastline\space
 \ifcase#1 de\or\or not \or already \fi activated;
 \string\parfillskip=\the\parfillskip}}
% execute badness test with customized parfillskip
\def\checkshortlastline{%
 \parfillskip=0pt plus 5.17\hsize  % TUGboat value
 \endgraf\parfillskip=0pt plus 1fil
 \ifnum\badness=1 \errhelp=\CSLLhelp         % HIT
  \errmessage{Notice: Short (or tight) last line}%
 \fi}
% start the check but don't do it twice
\def\activatecheckshortlastline{%
 \ifCSLLactive\CSLLmsg(3)\else\CSLLactivetrue
  % 1st: new def of \par: in hmode do the check
  \def\par{\ifhmode\ifinner\else % horizontal mode
     \ifCSLLaccepted \CSLLacceptedfalse % user: OK
     \else\checkshortlastline\fi\fi   % else check
     \fi\endgraf}%    reset \badness with \endgraf
  % 2nd: save \parfillskip and use default value
  \CSLLpfs=\parfillskip \parfillskip=0pt plus 1fil
  \CSLLmsg(1)\fi}%  write message about activation
\def\stopcheckshortlastline{%  stop check but only
 \ifCSLLactive\CSLLactivefalse  % if it is running
  \let\par=\endgraf % undo changes: reset \par and
  \parfillskip=\CSLLpfs \CSLLmsg(0)%  \parfillskip
 \else\CSLLmsg(2)\fi}
% accept a line that creates a false hit
\let\lastlineaccepted=\CSLLacceptedtrue
```

Figure 2: Macros to check badness of last lines

Experiment 15: Description
Show that a positive natural width for `\parfillskip` might produce overfull lines.

TEX input

```
\noindent 15. Each interword glue in this
single-line text shrinks.
```

TeX output

- \parfillskip ← 0 pt plus 1fil (plain TeX's default):

15. Each interword glue in this single-line text shrinks.

- \parfillskip ← 10 pt plus 1fil:

15. Each interword glue in this single-line text shrinks. |

- \parfillskip ← 10 pt:

15. Each interword glue in this single-line text shrinks. |▯

The last two paragraphs have a badness larger than 10000 as the shrinkability of the line is not large enough to fit the line width [12, §851]. In this case \tracingparagraphs reports no value but outputs the badness as *. Without a valid badness the demerits are set to * too.

Experiment 15 continued: Log file contents

```
@secondpass
\ninerm 15. The in-ter-word glue in this
  one-line para-graph shrinks.
@\par via @@0 b=* p=-10000 d=*
@@1: line 1.3- t=0 -> @@0
```

▯

O11 (Glue shrinks early): Interword glue shrinks in last lines early if \parfillskip has a net contribution > 0 pt. When the distance of the natural width of the material in the last line to the right margin is smaller than the natural width of \parfillskip the interword glue starts to shrink.

O12 (Overfull line): If \parfillskip's net contribution to the last line is always greater than 0 pt TeX might typeset an overfull last line. As the total shrinkability in the last line cannot make the text fit into the line width, TeX computes neither the badness nor the demerits for the line.

Overfull lines are signaled by plain TeX as a warning in the log file, on the terminal, and with a black rectangle in the output. (For the experiments in this article a very thin, 1 pt wide, rule is used.) But there is another way to produce an "overfull line" without notification by TeX.

Experiment 16: Description

Show that negative natural width for \parfillskip can create last lines that are wider than \hsize.

TeX output

- \parfillskip ← −15 pt plus 1fil:

1. Please answer if my topic is "in" or "out". TeX:▯in

O13 (Stick out right): If the glue specification of \parfillskip has the effect of extending the \hsize then the last line might stick out into the right margin without marking this line as overfull.

Negative natural width can be useful in certain applications. For example, in [2, p. 112], a macro is presented that uses \rightskip to provide stretchability in all lines and a negative natural width for \parfillskip to put some material flush right into the white space that the ragged-right setting leaves. The macro is used to break long headers in a table of contents; the material that must be placed in the last line is the page number. If its width is called x, \rightskip gets x plus 2em and $-x$ is assigned to \parfillskip. But again, this article experiments only with changes to \parfillskip alone.

Using finite stretchability. The combination of finite stretchability with a natural width is in some respect a contradiction: The finite stretchability is used to make (1) more likely, i.e., to avoid a short last line, but the natural width adds white space so the line does not appear to be short or rather its badness becomes smaller as less stretchability is needed. Nevertheless earlier experiments show that a badness as small as 1 is enough to change TeX's line-breaking decisions. So it should be possible to find cases where the last line is absorbed or extended with a specification having finite stretchability.

Experiment 17: Description

Show that the combination of natural width and finite stretchability for \parfillskip might produce a longer last line.

TeX output

- \parfillskip ← 10 pt plus \hsize:

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a

- \parfillskip ← 10 pt plus 0.7\hsize:

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a ▯

With a small variation of the text the paragraph gets shorter.

Experiment 18: Description

Show that the combination of natural width and finite stretchability for \parfillskip might remove a short last line.

TeX output

- \parfillskip ← 10 pt plus \hsize:

4′. My keyboArd is broken; when I press an 'A' in lowercAse (only) the screen repeAts it four times: a a a a ▯

A pure finite stretchability of width p^+ is not equivalent to a specification where the sum of natural width and stretchability equals p^+. This is obvious as the stretchability is reduced when a part of it becomes fixed width.

With natural width $\nu' = (1/10)$\hsize and a stretchability of $(\nu''/10)$\hsize for \parfillskip the input (∗) of section 4 has the following badness values, β.

$\nu'' =$	4	5	6	7	8	9	19	29	39
$\beta =$	1137	581	336	211	142	100	10	3	1

Experiment 19: Description
Show that a specification with natural width and finite stretchability might not output a single line.

TEX output

- `\parfillskip` ← 0 pt plus 0.7`\hsize`:

6. One line or two for this text? That is the question, or?

- `\parfillskip` ← 0.1`\hsize` plus 0.6`\hsize`:

6. One line or two for this text? That is the question, or? ▯

Compare the results with experiment 6: The natural width of `\parfillskip` prevents the absorption of the last line in the second paragraph. TEX executes a second pass which makes the short last line longer. A smaller value for the natural width of `\parfillskip` is sufficient in this case though.

Discussion. The use of a small value for the natural width of `\parfillskip` is ideal to assure (2), although overfull lines can occur and interword spaces shrink early. And short last lines might be produced when TEX has to add a line to the paragraph, i.e., (1) is violated in more cases than before.

With the default setting of `\parfillskip` an author has to check all last lines to ensure that neither (1) nor (2) is violated. With a natural width and infinite stretchability all last lines must be checked at most for a violation of (1) as (2) is either guaranteed or it fails with an error message. In some sense overfull lines are not bad as they point out a significant problem. Either an author rewrites the text or some other action is taken.

An overfull last line for a single paragraph can be solved by the technique mentioned in section 3 but now *with* a factor of −1 (or a fraction of −1) and not just a minus sign:

```
\hskip-1\parfillskip\ \par
```

This cancels only the natural width not the stretchability of `\parfillskip`. The line might be filled completely, violating (2).

Table 3 summarizes all the observations made in the experiments of this section. The entries show that problems are possible with infinite stretchability, and the situation does not improve if the stretchability is reduced to a finite value. Although few experiments were shown with a small variation of values for finite stretchability the findings in section 4 also apply here in many aspects. For example, a too small stretchability produces underfull lines and even lines with artificial demerits.

Summary. Using a nonnegative natural width in `\parfillskip`, for example, 10 pt, is an effective way to have all last lines end before they reach the right margin so that recommendation (2) is always obeyed. Overfull lines might occur, which can be fixed either by rewriting the text or canceling the natural width of `\parfillskip`. But the setting has a side effect: Short last lines might be produced even with finite stretchability.

Table 3: Observations for natural width & stretchability

Specification of `\parfillskip` ($h \equiv$ `\hsize`)							
	natural width	0pt	<0pt	10pt	10pt	10pt	10pt
	stretch	1fil	1fil	1fil	h	.7h	0pt
	shrink	0pt	0pt	0pt	0pt	0pt	0pt
Observation							
1	Short line	1		(1)	(1)	(1)	(1)
2	Completely filled	2		–	–	–	–
3	Glue stretches	–		–	17	17	(3)
4	2nd pass forced	–		15	(15)	17	15
5	Underfull line	–		–	–	(1)	(3)
6	No demerits	–		15	(15)	(1)	15
7	Remove short line	–		–	18	(18)	(18)
8	Extend last line	–		–	17	17	(5)
9	Backspaces	–		–	–	–	–
10	Add short line	–		14	(14)	(14)	14
11	Glue shrinks early	–		15	(15)	(15)	15
12	Overfull line	–		15	(15)	(15)	15
13	Stick out right	–	(16)	–	–	–	–

Legend: –/ /+: never/don't care/always
n/(n): (implicitly) shown in example number n

7 Theory: Natural width and stretchability

The formula to compute the badness introduced in section 5 can be used if the sum of the width of the text and p° is not greater than h. The numerator, which contains the used stretchability, has to apply not only the width of the text but also p°. But if the sum of these two values is greater than h then the glue in the last line must shrink and a different calculation is required. Well, the formula stays the same but now the used shrinkability and the available shrinkability form the quotient.

If the length of the text in the last line is named t then the used stretchability is $h - t - p^\circ$ if $t + p^\circ \le h$. Otherwise, i.e., $h < t + p^\circ < 2h$, the required shrinkability is $t + p^\circ - h$. This amount must come from the shrinkability of the text as $p^- = 0\,\mathrm{pt}$. If $t + p^\circ$ is too large then the line gets overfull. Therefore, in most cases the value of p° should not be very large and less than h, unless the text has unusual shrinkability.

In experiment 19 TEX uses different line breaks for a specification with pure stretchability and for one in which the sum of a nonnegative natural width and a smaller stretchability equals the stretchability of the first specification. In the experiment different passes occur. But if the last lines are identical and all glue stretches, both specifications typeset the paragraph in the same pass. The specification with

$p^\circ = 0\,\mathrm{pt}$ has an equal or greater badness in the first pass if the badness values are less than 100; in a second pass, its badness is equal or smaller. Compare the badness values shown before experiment 19, for input $(*)$, with the corresponding values in section 4.

For the proof of this statement the following shortcut is defined. The sum of all horizontal glue in the last line, i.e., interword spaces and skips, except the `\parfillskip`, is named **g**. It has natural width g°, stretchability g^+, and shrinkability g^-, as with other glue parameters.

If $\nu = \nu' + \nu''$ and the last line contains text of width t with stretchability g^+ the two badness values for the specifications `0pt plus` ν`\hsize` and ν'`\hsize plus` ν''`\hsize` are approximately

$$100\Bigl(\frac{h-t}{\nu h + g^+}\Bigr)^3 \quad\text{and}\quad 100\Bigl(\frac{h-t-\nu' h}{\nu'' h + g^+}\Bigr)^3.$$

To make sure that the badness values are not larger than the tolerance for the second pass, $\tau = 200$, the relation $\sqrt[3]{2}\nu'' \geq 1 - \nu' - (t + \sqrt[3]{2}g^+)/h$ or, better, $\nu'' \geq (1-\nu')/\sqrt[3]{2}$ must hold.

In a first pass with badness < 100 the quotient of the second specification that is cubed obeys

$$0 \leq \frac{h-t-\nu' h}{\nu'' h + g^+} < 1 = \frac{\nu' h}{\nu' h}.$$

The quotient of the first specification, $(h-t)/(\nu h + g^+)$, is the *mediant*, i.e., the quotient of the sum of the numerators and the denominators of the two quotients on the left and the right of 1, and therefore

$$\frac{h-t-\nu' h}{\nu'' h + g^+} < \frac{h-t}{\nu h + g^+} < \frac{\nu' h}{\nu' h} = 1.$$

A similar argument shows

$$\frac{h-t-\nu' h}{\nu'' h + g^+} > 1 \Longrightarrow \frac{h-t-\nu' h}{\nu'' h + g^+} > \frac{h-t}{\nu h + g^+} > 1.$$

The relations are not changed when they are cubed and multiplied by 100 to get the badness. TEX's computation [12, §108] is monotone but only an approximation: Both relations "less than" and "greater than" should also be "or equal to".

Using the `\tracingparagraphs` output. TEX has a parameter to report line-breaking decisions and write them to the log file: `\tracingparagraphs` [19]. Its output has already been shown several times, for example, in the discussion after experiment 3. This trace shows all the ways for TEX to typeset the text of a paragraph with the current values of the line-breaking parameters. The glue `\parfillskip` is not explicitly mentioned in a trace but the possible last lines carry in their badness and thus in their line demerits the influence of this parameter.

Experiment 16 sets `\parfillskip` to a strange value, creating for the last line a solution that is not part of the network of line breaks that is built from the default settings of plain TEX. But the effect is limited to the last line — and its previous line if it becomes the last line — as no other line breaks can be affected by the bad setting of `\parfillskip`, i.e., all sets of line breaks up to the start of the penultimate line are also valid line breaks for the default setting. A second important aspect of giving `\parfillskip` finite values is the interaction with infinite stretchability in the text. For example, the trace for experiment Xb with the default setting of `\parfillskip` and a positive value for `\looseness` to force a second pass lists five different paths; with a stretchability of 6`\hsize` this number increases to nine. But as stated before, input with infinite stretchability is not analyzed in this article.

Therefore only the trace of a "normal text" with a reasonable `\parfillskip` is considered. Experiment 6 with a forced second pass outputs this trace:

```
@secondpass
\ninerm 6. One line or two for this text?
 That is the ques-
@\discretionary via @@0 b=84 p=50 d=11336
@@1: line 1.1- t=11336 -> @@0
tion,
@ via @@0 b=2 p=0 d=144
@@2: line 1.2 t=144 -> @@0
or?
@\par via @@0 b=65 p=-10000 d=5625
@@3: line 1.3- t=5625 -> @@0
@\par via @@2 b=0 p=-10000 d=100
@\par via @@1 b=0 p=-10000 d=5100
@@4: line 2.2- t=244 -> @@2
```

This trace contains three paths which are described in Table 4. All the paths have been selected with some specification of `\parfillskip`: The single line and the first two-line solution that is typeset with the default setting are shown in experiment 6, while

Table 4: Badness, penalties, and additional demerits of the line breaks for the three paths of the trace listing

		`\par via @@` (`*` is typeset)		
`@@`	Class	`0`$_0$	`*2`$_0$	`1`$_0$
`1`	l			84_{50}
`2`	d		2	
`3`	t	65		
`4`	d		0	0^{f}
# lines =		1	2	2
$\sum$ badness =		65	2	84
# a/d/f =		0/0/0	0/0/0	0/0/1
$\Lambda_{\mathrm{p}}(10)$ =		5625	244	16436
Last line with `\parfillskip` $\equiv \mathbf{p} = (0\,\mathrm{pt}, 0.8h, 0\,\mathrm{pt})$:				
class : data =		`*` t : 65	v : 164^{a}	v : 115^{f}
$\Lambda_{\mathrm{p}}(10)[\mathbf{p}]$ =		5625	40420	31961
stretch/pt =		n/a	0	1.92706

the third path is typeset in experiment 19. In the following, we analyze how a setting of `\parfillskip` can be determined to select one of the possible paths given the data of Table 4.

Here is a brief description of the table. The headline shows the three paths that have lines in the trace which start with `@\par`, i.e., lines 9, 11, and 12 and list the feasible breakpoint. The typeset `\par` column gets an asterisk. Two lines at the left show in each row the number of the feasible breakpoint and the first letter of the associated fitness class (coded as a number in the trace): very loose, loose, decent, tight. Each feasible breakpoint of a path gets an entry of the form

$$\text{badness}\begin{array}{l}\text{additional demerits as letters a, d, and f}\\ \text{penalty at the break if} \neq -10000\end{array}$$

in which the applied additional demerits must be determined by a calculation using the demerits `d` of the trace line [21]. The other two values are listed directly in the trace line as `b` and `p`. The four rows at the end of the table, i.e., the number of lines, the sum of the badness values, the occurrences of the additional demerits, and the path demerits, are calculated from the entries in each column.

As mentioned earlier, the sum of all line demerits for a set of line breaks is called the path demerits; the smallest value of all path demerits are the total demerits of the paragraph and the associated set of line breaks is used by TeX to typeset the text [11, p. 97].

The following formula calculates the line demerits Λ_ι for line number ι [11, p. 98; 20, section 2]:

$$\Lambda_\iota = (\lambda + \beta_\iota)^2 + \operatorname{sgn}(\pi_\iota)\pi_\iota^2 + \delta_\iota$$

where λ is the `\linepenalty`, β_ι stands for the badness of the line, π_ι represents the penalty at the line break, and δ_ι is the sum of the applicable additional demerits, i.e., the values of `\finalhyphendemerits`, `\doublehyphendemerits`, and `\adjdemerits`.

The table must be extended compared to previous instances; see [20] and [21]. To get badness values that distinguish last lines of different length, a second trace is generated with the finite stretchability of 0.8`\hsize` for `\parfillskip`. (The trace can be generated directly with this `\parfillskip` and only one additional line would be needed.) The fitness class and the new entry for the last line, the new path demerits, and the sum of all stretchability of the material in each last line are shown in three more lines. The last line in Table 4 contains data that cannot be extracted from the trace. It is shown to allow exact computations but the influence is so small that the data can be dropped.

With the default `\parfillskip` — and therefore also with a stretchability of 6`\hsize` — the path of column $\mathtt{2}_0$ is typeset and with the 0.8`\hsize` column $\mathtt{0}_0$ is chosen. (That is known, as the stretchability of $p^+ = h$ proved it in experiment 6.) But what is the maximum value for p^+ to select column $\mathtt{0}_0$? What setting for `\parfillskip` causes TeX to select column $\mathtt{1}_0$? These theoretical questions are answered in the next subsections.

Not all paths shown in a path table might be distinguishable through settings for `\parfillskip` as the difference can lie in the line breaks before the last line. Two sets of line breaks cannot be distinguished by a setting of `\parfillskip` if their last lines are identical. In this case the column head of the path table shows the same feasible breakpoint number but with a different subscript. No such case appears in Table 4.

And the final remark: A last line can be selected with certainty by a setting of `\parfillskip` only if its badness with $\mathbf{p}$ differs from all other last lines. A change of the stretchability of `\parfillskip` produces different line demerits for last lines only if the change assigns different badness values to the last lines. For this reason the `\parfillskip` $\mathbf{p} = (0\,\text{pt}, 0.8h, 0\,\text{pt})$ was selected in the path table as it often assigns different badness values to short lines of different lengths.

Absorb short last line. To get path $\mathtt{0}_0$ of Table 4 instead of $\mathtt{2}_0$ their demerits must obey the relation

$$\Lambda_{\mathrm{p}[2_0]}(10)[\mathbf{p}'] > \Lambda_{\mathrm{p}[0_0]}(10)[\mathbf{p}']$$

for the unknown `\parfillskip` $\mathbf{p}'$. The constant `\linepenalty` is not changed and therefore "(10)" is dropped together with the $\mathbf{p}'$ in the notation and only the prime is moved to Λ_p to signal that a different `\parfillskip` is used. So the notation of the relation is simplified to

$$\Lambda'_{\mathrm{p}[2_0]} > \Lambda'_{\mathrm{p}[0_0]}.$$

Note: $\Lambda_{\mathrm{p}[0_0]}$ is not changed when the stretchability changes as it is a tight line; i.e., $\Lambda'_{\mathrm{p}[0_0]} = \Lambda_{\mathrm{p}[0_0]}$.

Let's say that with p'^+ the badness of the last line in path $\mathtt{2}_0$ changes from β to $\beta\chi$. To get an approximation for p'^+, the formula for the approximation of badness values is used twice. First calculate

$$\beta \approx 100\Bigl(\frac{\text{used stretchability}}{0.8h + g^+}\Bigr)^3$$

where g^+ is the stretchability in the text.

For all p'^+ the used stretchability is the same; only the available stretchability changes. Instead of $0.8h+g^+$ the available stretchability is now $p'^+ + g^+$.

Therefore

$$\beta\chi \approx 100\Big(\frac{\text{used stretchability}}{p'^{+} + g^{+}}\Big)^{3}.$$

Dividing the quotients for $\beta\chi$ and β gives a formula for p'^{+}. First

$$\chi = \frac{\beta\chi}{\beta} \approx \Big(\frac{0.8h + g^{+}}{p'^{+} + g^{+}}\Big)^{3}$$

and thus

$$p'^{+} \approx \frac{0.8h + g^{+}}{\sqrt[3]{\chi}} - g^{+} = \frac{0.8}{\sqrt[3]{\chi}}h - (1 - \frac{1}{\sqrt[3]{\chi}})g^{+}.$$

The value of h is known and g^{+} is given in the path table. Only χ must be determined from the above stated inequality.

To calculate χ the change of the line demerits must be analyzed using the formula for line demerits. The relation $\Lambda'_{\mathrm{p}[2_0]} > \Lambda'_{\mathrm{p}[0_0]} = \Lambda_{\mathrm{p}[0_0]}$ becomes

$$\Lambda_{\mathrm{p}[2_0]} - (\lambda+\beta)^2 - \delta_a[\beta\chi < 100] + (\lambda+\beta\chi)^2 > \Lambda_{\mathrm{p}[0_0]}$$

as (i) only the last line changes its demerits as it changes its badness, (ii) penalties in the last line are not affected, and (iii) the `\adjdemerits`, δ_a, might disappear if the last line is no longer very loose. (This is expressed with *Iverson's convention*: If the condition in the brackets is true then the value of the bracket is 1, otherwise 0.) Actually the bracket can be set to 1 as it is already known that the badness is less than 100 because `\hsize` is sufficient for p'^{+}.

$$-2\lambda\beta - \beta^2 + 2\lambda\beta\chi + \beta^2\chi^2 > \Lambda_{\mathrm{p}[0_0]} - \Lambda_{\mathrm{p}[2_0]} + \delta_a$$

$$\iff \qquad \chi^2 + 2\frac{\lambda}{\beta}\chi - 1 - 2\frac{\lambda}{\beta} > \frac{\Lambda_{\mathrm{p}[0_0]} - \Lambda_{\mathrm{p}[2_0]} + \delta_a}{\beta^2}$$

Adding $0 = (\lambda/\beta)^2 - (\lambda/\beta)^2$ to the left side generates two quadratic terms on this side. Now the inequality can be solved for χ.

$$\Big(\chi + \frac{\lambda}{\beta}\Big)^2 > \frac{\Lambda_{\mathrm{p}[0_0]} - \Lambda_{\mathrm{p}[2_0]} + \delta_a}{\beta^2} + \Big(1 + \frac{\lambda}{\beta}\Big)^2$$

$$\implies \qquad \chi > \sqrt{\frac{\Lambda_{\mathrm{p}[0_0]} - \Lambda_{\mathrm{p}[2_0]} + \delta_a + (\lambda+\beta)^2}{\beta^2}} - \frac{\lambda}{\beta}.$$

The last step is of course valid only if $\Lambda_{\mathrm{p}[0_0]} - \Lambda_{\mathrm{p}[2_0]} + \delta_a + (\lambda+\beta)^2 \geq 0$. This condition means that the path demerits $\Lambda_{\mathrm{p}[2_0]}$ without the line demerits of its last line must not be greater than the line demerits of the path $\mathtt{0}_0$. This is of course true, otherwise path $\mathtt{2}_0$ would not be typeset instead of the single line.

With plain TeX's value $\delta_a = 10000$ [11, p. 98] and with the numbers listed in Table 4, χ must obey

$$\chi > \sqrt{\frac{5625 - 40420 + 10000 + 174^2}{164^2}} - \frac{10}{164}$$
$$= \sqrt{\frac{5481}{26896}} - \frac{10}{164} \approx 0.3904.$$

Thus $\chi = 0.391$ is sufficient. The badness becomes $\approx 0.391 \times 164 \approx 64$ for the new last line of column $\mathtt{2}_0$. Thus, as $g^{+} = 0\,\mathrm{pt}$ for column $\mathtt{2}_0$,

$$p'^{+} \approx \frac{0.8}{\sqrt[3]{0.391}}h \approx 1.0940h.$$

Experiment (verification): Description
Show that the computed value — with an accuracy of two digits after the decimal point — selects the requested path.

TeX output

- `\parfillskip` ← 0 pt plus 1.09`\hsize`:

6. One line or two for this text? That is the question, or? ▯

Of course, badness is only a heuristic and there is only an approximation for its computation, so the result cannot be exact. The factor for h in p'^{+} so that $\mathtt{0}_0$ is used instead of $\mathtt{2}_0$ is ≈ 1.09088.

Extend short last line. With 0.8`\hsize` the path demerits of column $\mathtt{1}_0$ are lower than the path demerits of $\mathtt{2}_0$; that means the path of $\mathtt{1}_0$ is now preferred to $\mathtt{2}_0$. But the path $\mathtt{0}_0$ receives the lowest demerits. The latter choice can be eliminated with a positive natural width, for example, 0.02`\hsize` is sufficient for the stretchability 0.8`\hsize` — or better 0.78`\hsize` according to the above theory — to switch to $\mathtt{1}_0$ as the single line is already very tight. (Note as the badness is 65 the glue in the single line shrinks already by more than 86%. So a small natural width excludes this path.)

Nevertheless it is interesting to find a maximal value for the stretchability to make TeX choose $\mathtt{1}_0$. The approach of the previous subsection can be applied here too. But as now two badness values are changed, there are two factors χ and χ'. As the penultimate line in $\mathtt{1}_0$ has fitness class "loose" there is no δ'_a.

$$\Lambda_{\mathrm{p}[2_0]} - (\lambda+\beta)^2 + (\lambda+\beta\chi)^2 - \delta_a[\beta\chi < 100]$$
$$> \Lambda_{\mathrm{p}[1_0]} - (\lambda+\beta')^2 + (\lambda+\beta'\chi')^2$$

The value χ' depends on the stretchability of the text in last line, i.e, on the 1.92706 pt as noted in Table 4. But to simplify the calculations χ' is replaced by χ as they are nearly equal in this case. As shown before,

$$\chi \approx \Big(\frac{0.8h + g^{+}}{p'^{+} + g^{+}}\Big)^{3} \quad \text{and} \quad \chi' \approx \Big(\frac{0.8h + g'^{+}}{p'^{+} + g'^{+}}\Big)^{3}$$

so that

$$\frac{\chi'}{\chi} = \Big(\frac{0.8h + g'^{+}}{0.8h + g^{+}} \cdot \frac{p'^{+} + g^{+}}{p'^{+} + g'^{+}}\Big)^{3}.$$

Both quotients on the right side are ≈ 1 as a short line with at most one space is extended to a little bit longer line with maybe one more space. Here $g^{+} = 0\,\mathrm{pt}$ and $g'^{+} = 1.92706\,\mathrm{pt} < 0.0086h$. To drop such small values compared to $0.8h$ and p'^{+} in a calculation that is based only on approximations should not do much harm.

This time it is expected that $\beta\chi > 100$ holds so the term with δ_a is dropped; i.e., the value of `\adjdemerits` is still applied in the new set of line breaks. Therefore

$$(\beta^2 - {\beta'}^2)\chi^2 + 2\lambda(\beta - \beta')\chi \\ > \Lambda_{\mathrm{p}[1_0]} - (\lambda + \beta')^2 - \Lambda_{\mathrm{p}[2_0]} + (\lambda + \beta)^2.$$

The formulas get rather wide for this small column width so instead of stating the difference between path demerits and part of the line demerits of the last line, the line demerits of the first line are used. That is

$$\Lambda'_1 = \Lambda_{\mathrm{p}[1_0]} - (\lambda + \beta')^2 - \delta_f$$
$$\Lambda_1 = \Lambda_{\mathrm{p}[2_0]} - (\lambda + \beta)^2 - \delta_a.$$

So δ_a gets back and δ_f is introduced into the inequality:

$$(\beta^2 - {\beta'}^2)\chi^2 + 2\lambda(\beta - \beta')\chi > \Lambda'_1 + \delta_f - \Lambda_1 - \delta_a.$$

After the division with $\beta^2 - {\beta'}^2 > 0$ and the addition of $\lambda^2/(\beta + \beta')^2$ this relation becomes

$$\Bigl(\chi + \frac{\lambda}{\beta{+}\beta'}\Bigr)^2 > \frac{\Lambda'_1 + \delta_f - \Lambda_1 - \delta_a}{\beta^2 - {\beta'}^2} + \frac{\lambda^2}{(\beta{+}\beta')^2}$$

and then

$$\chi > \sqrt{\frac{\Lambda'_1 + \delta_f - \Lambda_1 - \delta_a}{\beta^2 - {\beta'}^2} + \frac{\lambda^2}{(\beta{+}\beta')^2}} - \frac{\lambda}{\beta{+}\beta'}.$$

With the numbers of Table 4, $\Lambda'_1 = 94^2 + 50^2 = 11336$ and $\Lambda_1 = 12^2 = 144$, as well as $\delta_f = 5000$ [11, p. 98], so that

$$\chi > \sqrt{\frac{11336 + 5000 - 144 - 10000}{164^2 - 115^2} + \frac{10^2}{279^2}} - \frac{10}{279}$$
$$\approx 0.6739 - 0.0358 = 0.6381$$

and then follows as before

$$p'^{+} = \frac{0.8}{\sqrt[3]{\chi}}\,h \approx \frac{0.8}{\sqrt[3]{0.639}}\,h \approx 0.929h.$$

Therefore the specification 0.02`\hsize` for the natural width and 0.90`\hsize` for the stretchability of `\parfillskip` avoids the selection of column $\mathtt{0}_0$ and makes TeX typeset $\mathtt{1}_0$. Again the result is not exact: Setting $p'^{+} = 0.9086$`\hsize` together with a natural width of 0.02`\hsize` does the job too.

Experiment (verification): Description
Show that the abovementioned values for natural width and stretchability of `\parfillskip` typeset the selected path.

TeX output

- `\parfillskip` ← 0.02`\hsize` plus 0.90`\hsize`:

6. One line or two for this text? That is the question, or? □

Comparing finite stretchability. The length of a short last line has no influence on TeX's line-breaking decisions with the default plain TeX settings. TeX picks the set of line breaks that gives the smallest sum of line demerits for all but the last line. If the text has μ lines then

$$\sum_{\iota=1}^{\mu-1} \Lambda_\iota + \delta_\mu < \sum_{\iota=1}^{\mu-1} \Lambda'_\iota + \delta'_\mu$$

where the non-primed line demerits Λ_ι represent the typeset output. Then 100 demerits have to be added for both short last lines to get the path demerits.

With a finite stretchability, different lengths of the last line generate different line demerits, so instead of 100 demerits on each side of the inequality different numbers are added, and that might change the relation symbol from less than to greater than for the path demerits. This means that finite stretchabilities can be compared by the maximum value of demerits for all but the last line by the point where the relation symbol change will happen. That the demerits can increase significantly was seen with experiment 11 in section 4.

Here are the differences for the line demerits of short last lines for four stretchabilities (without looking at the stretchability of the text):

p^+	2% → 4%	4% → 6%	6% → 8%	8% → 10%
h	1212	955	905	855
$.9h$	2421	1524	1920	1356
$.8h$	3740	3883	3320	2817
$.7h$	8800	8789	6804	6855

Therefore to extend the last line from $0.02h$ to $0.04h$ with $p^+ = h$ TeX saves 1212 demerits so that the sum of line demerits for the rest of the paragraph might have up to 1211 demerits more than the original typeset lines. And it can be $1212 + 955 - 1$ demerits if the last line can be extended to $0.06h$.

For $p^+ = 0.8h$ and $p^+ = 0.7h$ all, and for $p^+ = 0.9h$ all but the last listed line length produce very loose last lines if the text in this line has no stretchability, and `\adjdemerits`, $\delta_a = 10000$, are charged except the penultimate line is loose. In this case it can be said that $p^+ = 0.9h$ tolerates ≈ 2.0 times more demerits than $p^+ = h$ when a line of $0.02h$ is extended. The factor for $p^+ = 0.8h$ lies at ≈ 3.5 and for $p^+ = 0.7h$ it is ≈ 8.0. When the default `\adjdemerits` are applied to the last line only $p^+ = 0.7h$ starting at $0.06h$ has positive factors.

8 Replacing stretchability by shrinkability

The overfull line problem that exists in the combination of natural width with stretchability can be addressed by specifying natural width together with finite shrinkability instead of stretchability in the glue parameters of `\parfillskip`. Recall that in the case of stretchability the interword space can change without limit but with shrinkability, only its given maximum is allowed. Also, the shrinkability cannot be of infinite order as TEX throws an error with such a specification [12, §853, §826]. Besides this, there is a fundamental difference between a specification that has stretchability and one with natural width and shrinkability. For example, if all values are specified as `\hsize` then in the first case short lines have a high badness as more stretchability must be used. In the second case lines that contain a lot of text have greater badness as the shrinkability gets higher to compensate for the natural width.

Shrinkability larger than the natural width. A shrinkability that is larger than the natural width might make TEX put some text into the right margin as it creates a situation similar to negative natural width with stretchability; see experiment 16.

Experiment 20: Description
Show that the shrinkability of `\parfillskip` should not be larger than its natural width.

TEX output

- `\parfillskip` ← `\hsize` minus 3`\hsize`:

    1. Please answer if my topic is "in" or "out". TEX: in

- `\parfillskip` ← 0.9`\hsize` minus 0.9`\hsize` + 1 em:

    1. Please answer if my topic is "in" or "out". TEX: in▯

Note again that TEX gives no warning or error message in these cases.

Shrinkability equals natural width. When the input (∗) of section 4 is processed by TEX with a setting of `\parfillskip=`ν`\hsize minus `ν`\hsize` the badness β is:

$\nu =$	1	2	3	4	5	6	7	8	9
$\beta =$	0	12	30	42	51	57	63	66	70

As with this specification more text in the last line of the paragraph results in a higher badness value, it seems that in this case the input

```
\noindent\hbox to 0.9\hsize{\hss}    (**)
```

is also of interest, i.e., the line is 90% filled without additional shrinkability.

$\nu =$	1	2	3	4	5	6	7	8	9
$\beta =$	73	86	90	92	94	95	95	96	96

In contrast to a specification with natural width and stretchability, this specification can cancel the natural width. Nevertheless, short new lines are preferred by TEX here too as the badness of long last lines is so high. That is, the effect of O10 "Add short line" is also seen with this specification.

Experiment 21: Description
Show that short last lines can occur with natural width and the same amount of shrinkability for `\parfillskip`.

TEX output

- `\parfillskip` ← `\hsize` minus `\hsize`:

    2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?
    4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a ▯

The text of experiment 4 demonstrates that even a single letter on the last line is not moved to the penultimate line to form the new last line. Actually it will never happen that a paragraph with a single word of width up to 0.17`\hsize` in the last line is reduced by one line with the specification of `\parfillskip` used in experiment 21; see the next section. Nevertheless, it does not mean that a line that ends at the right margin is impossible.

Experiment 22: Description
Show that a last line might end flush right.

TEX output

- `\parfillskip` ← `\hsize` minus `\hsize`:

    2′. Has the last line of this paragraph badness 0 and has no interword space to stretch?! Do they shrink now?! ▯

If the specification uses a fraction of `\hsize`, `\parfillskip=`$(\nu/10)$`\hsize minus `$(\nu/10)$`\hsize`, the input (∗) always produces last lines with infinite badness, as the text width plus the natural width of `\parfillskip` is shorter than `\hsize` and there is no stretchability. Such specifications have infinite badness for very short lines but the badness becomes 0 at the moment the text of the line reaches $(1-\nu/10)$`\hsize` to make a perfect fit with the line width. Longer lines get more and more badness.

When such a specification is confronted with a situation in which an interword space is available in a short last line, this glue has to stretch.

Experiment 23: Description
Show that a short line gets extremely spaced out if it has to stretch.

TEX output

- `\parfillskip` ← 0.8`\hsize` minus 0.8`\hsize`:

    3. A short text in 1 line. Or has the paragraph 2 or 3 lines?

- `\parfillskip` ← 0.7`\hsize` minus 0.7`\hsize`:

    3. A short text in 1 line. Or has the paragraph 2 or 3 lines? ▯

The second specification produces a different line break in the text of experiment 3 than plain TEX's default setting or a stretchability of 0.7`\hsize` does; see experiment 12. It is like experiment 3 with

`\parfillskip` set to 0 pt except that here the natural width limits the amount by which the space has to stretch. Of course, the last line has artificial demerits. This effect helps to avoid short last lines, for example, with texts 2, 4, and 6. They cannot have a second line as it would be a line with artificial demerits, i.e., a line that TEX does not consider as long as there are other ways to typeset the text.

Experiment 24: Description
Show that a shorter paragraph is possible with natural width and shrinkability.

TEX output

• `\parfillskip` ← 0.7`\hsize` minus 0.7`\hsize`:

2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?
4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a
6. One line or two for this text? That is the question, or? □

Note that the text of experiment 6 does not end flush right; its badness is 97. All glue items with shrinkability are involved in the process to shrink the line, and only with a badness of exactly 100 does the natural width of `\parfillskip` vanish completely.

Shrinkability smaller than natural width. If the shrinkability is smaller than the natural width recommendation (2) is supported: There is always some white space at the end of the last line. This is similar to the case of natural width and stretchability. But as the results of section 6 show, a positive contribution to the width of the last line by `\parfillskip` can produce overfull lines.

The following specification

```
\parfillskip=\hsize minus \hsize
\advance\parfillskip by -1.25\parindent
                        minus -1.25\parindent
\advance\parfillskip by 0pt minus -10pt
```

states that the natural width needs text with the width of 1.25`\parindent` to fill the line and if there is a lot of text the shrinkability must leave at least 10 pt of white space, as this is the amount by which it is shorter than the natural width.

Experiment 25: Description
Show some effects of the above specification.

TEX output

• `\parfillskip` ← `\hsize` − 1.25`\parindent` minus `\hsize` − 1.25`\parindent` − 10 pt:

1. Please answer if my topic is "in" or "out". TEX: in
2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?
6. One line or two for this text? That is the question, or?
15. Each interword glue in this single-line text shrinks. □

Texts 1 and 2 both produce an underfull last line with badness 10000 and no demerits.

When lines are very short and contain only one space this space must stretch a lot. The text must at least cover the width that is subtracted from `\hsize` in `\parfillskip`'s natural width.

Experiment 26: Description
Show that with such settings a last line must be as wide as the amount that `\parfillskip`'s natural width is smaller than `\hsize`.

TEX output

• `\parfillskip` ← `\hsize` − 1.1`\parindent` minus `\hsize` − 1.1`\parindent` − 10 pt:

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a
13. a

• `\parfillskip` ← `\hsize` − 1.25`\parindent` minus `\hsize` − 1.25`\parindent` − 10 pt:

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a
13. a □

In the first case the text of experiment 13 has badness 3, in the second it is 88. A larger factor, for example, 1.5, in front of the `\parindent`s creates an underfull last line with badness 10000 and no demerits.

Negative values. A negative natural width for `\parfillskip` behaves as if the line width increases and overfull lines without the overfull rule are easily produced.

Experiment 27: Description
Show that lines might not end at the right margin if the natural width is negative.

TEX output

• `\parfillskip` ← −10 pt minus −10 pt:

3. A short text in 1 line. Or has the paragraph 2 or
3 lines?

• `\parfillskip` ← −10 pt minus `\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or
3 lines? □

Negative shrinkability outputs an underfull last line if the sum of the text width and the natural width is < `\hsize`; overfull lines are possible too.

Experiment 28: Description
Show that last lines can be underfull or overfull if the shrinkability is negative.

TEX output

• `\parfillskip` ← 0.75`\hsize` minus −0.1`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or
3 lines?

• `\parfillskip` ← 0.87`\hsize` minus −0.1`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or
3 lines? □

With such specifications the stretch- or shrinkability of the text in the last line must work together perfectly with the natural width to output valid last lines; otherwise the line is under- or overfull.

Discussion. Specifications with natural width and shrinkability for `\parfillskip` can replace a specification that uses only stretchability or stretchability and natural width. But they have different properties. Long last lines receive high badness values so TeX now prefers line breaks that create a short last line. This contradicts recommendation (1). By the same token a short last line is not absorbed by the penultimate line to form a new last line.

Figure 3 shows the badness values for three specifications when fixed width text is used; note that the scale of the y-axis is not the same as in Fig. 1. The length of the text is again reported on the x-axis as a percentage of the `\hsize`. The first 30 values for `0.7\hsize minus 0.7\hsize` have infinite badness.

A setting where the natural width equals the shrinkability seems to be the best choice. If the value is too small, for example, only 0.7`\hsize`, problems like underfull or extremely spaced out last lines easily occur. If the shrinkability and the natural width both equal `\hsize`, recommendation (2) might not be obeyed, although it seems to be rare as in all last lines the glue shrinks. The white space of the natural width disappears only when the badness equals 100. If the shrinkability is smaller than the natural width, recommendation (2) is always obeyed although now overfull lines might occur. Table 5 lists all observations found in the experiments.

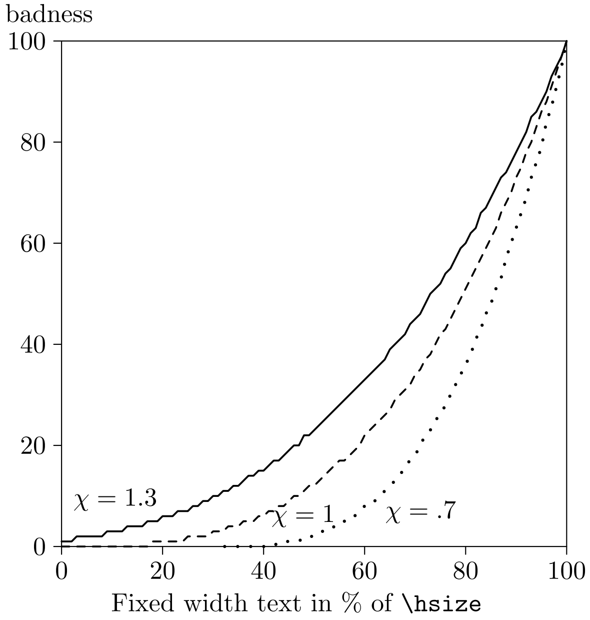

Figure 3: Graphs for $\chi = 0.7$, $\chi = 1$, and $\chi = 1.3$ when `\parfillskip` $\equiv \chi$`\hsize minus` χ`\hsize`

Table 5: Observations for natural width & shrinkability

Specification of `\parfillskip` ($h \equiv$ `\hsize`, $x > 0\,$pt,
$y \equiv h - 1.25$`\parindent`, $z \equiv y - 10\,$pt)

	natural width	0pt	x	h	$.7h$	y	-10pt	x
	stretch	1fil	0pt	0pt	0pt	0pt	0pt	0pt
	shrink	0pt	$>x$	h	$.7h$	z	x	$-.1h$
Observation								
1	Short line	1		21	24	25		
2	Completely filled	2		22	(22)	–		
3	Glue stretches	–		–	23	26		(28)
4	2nd pass forced	–		–	23	25		(28)
5	Underfull line	–		–	23	25		(28)
6	No demerits	–		–	23	25		(28)
7	Remove short line	–		–	24	–		
8	Extend last line	–		–	–	25		
9	Backspaces	–		–	–	–		
10	Add short line	–		21	–	25		
11	Glue shrinks early	–		+	24	25		(28)
12	Overfull line	–		–	–	25		(28)
13	Stick out right	–	(20)	–	–	–	(27)	

Legend: –/ /+: never/don't care/always
n/(n): (implicitly) shown in example number n

As the glue in the last line of a paragraph usually shrinks the problem with the spacing around displayed equations exists here too.

Summary. The use of natural width and shrinkability in the specification of `\parfillskip` works well but the dimensions must be selected carefully and short last lines might occur more often than before. If the values of natural width and shrinkability are equal and smaller than `\hsize`, many problems occur when the last line is not filled with enough text. If the shrinkability is larger or smaller than the natural width, overfull lines might appear.

9 Theory: Natural width and shrinkability

As mentioned in section 7, the formula to calculate the approximation for the badness in cases that involve glue that shrinks is the same as the one used in section 5 but now with a quotient built from the used shrinkability and the available shrinkability. For example, let's find the width t that a text without shrinkability in the last line can have so that the badness computes to 0 when $p^\circ = p^- = h$. The last line has a width of $t + h$ as p° is added to the width of the material. So the used shrinkability is t to make this sum fit the line width h. The available shrinkability is $p^- = h$ as there is no shrinkability in the text by assumption. Thus:

$$100\Bigl(\frac{t}{h}\Bigr)^3 < 0.5 \quad\Longrightarrow\quad t < \sqrt[3]{1/200}h \approx 0.17h.$$

To state it the other way: A line that contains material with a width up to $0.17h$ has badness 0 for sure when $\mathbf{p} = (h, 0\,\mathrm{pt}, h)$; see also Fig. 3.

It was stated above that a paragraph will never be shortened by one line when $p^\circ = p^- = h$ and the last line has a single word of width $\le 0.17h$, i.e., the value just found. Such a statement cannot be shown by experiment; it requires proof.

But first a note on the notation: Λ_ι is written for the line demerits of line number ι that TeX typesets if `\parfillskip` has the default setting; $\bar\Lambda_\iota$ is written for the line demerits of line ι with the changed `\parfillskip`. A prime is added to Λ and $\bar\Lambda$ if another set of line breaks than the typeset ones is considered.

Assume there is a set of line breaks that create $\mu-1$ lines instead of the μ lines that are typeset with the default `\parfillskip`. The sum of the line demerits for the shorter paragraph must be larger than the total demerits; otherwise it would be typeset:

$$\sum_{\iota=1}^{\mu-1} \Lambda'_\iota > \sum_{\iota=1}^{\mu} \Lambda_\iota.$$

It must be shown that this relation is kept with the changed `\parfillskip`; i.e., a proof for

$$\sum_{\iota=1}^{\mu-1} \bar\Lambda'_\iota > \sum_{\iota=1}^{\mu} \bar\Lambda_\iota$$

is needed. Then no shorter paragraph is considered by TeX if $\mathbf{p} = (h, 0\,\mathrm{pt}, h)$.

With the new setting of `\parfillskip` all lines except the last keep their line demerits as the line breaks are not changed. The line demerits of the last line are either unchanged or they increase because of the larger badness and maybe the addition of `\adjdemerits`. Therefore

$$\sum_{\iota=1}^{\mu-1} \bar\Lambda'_\iota = \sum_{\iota=1}^{\mu-2} \Lambda'_\iota + \bar\Lambda'_{\mu-1} \ge \sum_{\iota=1}^{\mu-2} \Lambda'_\iota + \Lambda'_{\mu-1} = \sum_{\iota=1}^{\mu-1} \Lambda'_\iota.$$

On the other hand, for the line breaks with μ lines the demerits do not change, as the last line is so short that its badness does not increase, as shown above. That is $\bar\Lambda_\mu = \Lambda_\mu$ and

$$\sum_{\iota=1}^{\mu} \bar\Lambda_\iota = \sum_{\iota=1}^{\mu-1} \Lambda_\iota + \bar\Lambda_\mu = \sum_{\iota=1}^{\mu-1} \Lambda_\iota + \Lambda_\mu = \sum_{\iota=1}^{\mu} \Lambda_\iota.$$

Therefore

$$\sum_{\iota=1}^{\mu-1} \bar\Lambda'_\iota \ge \sum_{\iota=1}^{\mu-1} \Lambda'_\iota > \sum_{\iota=1}^{\mu} \Lambda_\iota = \sum_{\iota=1}^{\mu} \bar\Lambda_\iota$$

and TeX does not typeset the $\mu - 1$ lines. QED.

Using the trace data. With specifications of natural width and shrinkability last lines cannot be extended or absorbed unless the specification makes other solutions impossible for TeX, i.e., the badness of the other last lines must become larger than the tolerance. Experiment 24 shows this effect for the text of experiment 6, for which the possible paths are documented in Table 4.

The default setting selects path $\mathtt{2}_0$. To get to path $\mathtt{0}_0$ by a specification for $\mathbf{p}$ with $p^\circ = p^-$ and $p^+ = 0\,\mathrm{pt}$, the short last line of $\mathtt{2}_0$ must receive infinite badness, or in other words p° must be smaller than the used stretchability. Let's name the used stretchability u; then

$$\begin{aligned} &\beta \approx 100\Bigl(\frac{u}{0.8h + g^+}\Bigr)^3 \\ \Longrightarrow\qquad &u \approx \sqrt[3]{\frac{\beta}{100}} \cdot 0.8h + \sqrt[3]{\frac{\beta}{100}}g^+. \end{aligned}$$

With the numbers of Table 4, u is easily computed as $\sqrt[3]{1.64}\cdot 0.8h + 0\,\mathrm{pt} \approx 0.9434\mathrm{h}$ and therefore $p^\circ = p^- = 0.94$ will typeset $\mathtt{0}_0$. (The exact value is 0.94394.)

Experiment (verification): TeX output

- `\parfillskip` ← 0.94`\hsize` minus 0.94`\hsize`:

6. One line or two for this text? That is the question, or? □

Note the badness increases to 98, so the line does not end flush right.

To get the path of column $\mathtt{1}_0$ the column $\mathtt{0}_0$ must be eliminated, for example, by making p° a little bit larger than p^-; the constant $0.02h$ was used in section 7, although this is rather large. Instead of badness 164 the last line now has badness 115. For an exact computation the value $g^+ = 1.97206\,\mathrm{pt}$ should be used; multiplied with $\sqrt[3]{1.15}$ it is $\approx 0.0089h$. As $\sqrt[3]{1.15} \cdot 0.8h \approx 0.8381h$ the setting $p^\circ = p^- + 0.02h = (0.8381 + 0.0089)h = 0.8470h$ typesets path $\mathtt{1}_0$; $p^\circ = 0.83722h$ is sufficient.

Experiment (verification): TeX output

- `\parfillskip` ← 0.84`\hsize` minus 0.82`\hsize`:

6. One line or two for this text? That is the question, or?
□

10 Adding stretchability back again

Experiments 23 and 26 show that spaces in a short line sometimes stretch a lot if natural width and shrinkability are specified for `\parfillskip`. In such a case, can a specification for stretchability help?

With finite dimensions in the specification of `\parfillskip` the glue of the last line of a paragraph might have its natural width by luck but usually it either has to shrink or to stretch. To avoid underfull lines the natural width and the stretchability should reach `\hsize` (see section 6) and to avoid overfull lines the shrinkability should have the same width as the natural width (see section 8). Therefore a specification like

$$\texttt{\textbackslash parfillskip} \equiv x \texttt{ plus } \texttt{\textbackslash hsize} - x \texttt{ minus } x$$

with $0\,\mathrm{pt} < x <$ `\hsize` $= h$ seems to qualify, as a first attempt. If the length of the last line is named t, then one the following cases

$$\texttt{\textbackslash parfillskip} \equiv \begin{cases} x \texttt{ plus } \texttt{\textbackslash hsize}-x, & t < h-x \\ x, & t = h-x \\ x \texttt{ minus } x, & t > h-x \end{cases}$$

are used with dimensions from `\parfillskip` depending on t. So short last lines, i.e., the width is less than $h-x$, use natural width and stretchability and long last lines, i.e., the width is greater than $h-x$, deploy natural width and shrinkability.

With $x = 0.75h$ the width of a last line reaches quite early the point where the badness becomes 0; see Fig. 4. The setting $x = 0.25h$ has the opposite effect. From the previous sections it is known that these settings behave quite differently with respect to adding or removing a short last line.

Experiment 29: Description

Show differences for short last lines with these settings.

TEX output

• `\parfillskip` ← 0.75`\hsize` plus 0.25`\hsize` minus 0.75`\hsize`:

2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?

4′. My keyboArd is broken; when I press an ‘A’ in lowercAse (only) the screen repeAts it four times: a a a a

• `\parfillskip` ← 0.25`\hsize` plus 0.75`\hsize` minus 0.25`\hsize`:

2. Has the last line of this paragraph badness 0 and has no interword space to stretch? Do they shrink now?

4′. My keyboArd is broken; when I press an ‘A’ in lowercAse (only) the screen repeAts it four times: a a a a □

The experiments show that both settings might typeset a paragraph differently from the output produced by the default setting. The smaller natural width prefers long last lines and absorbs a short last line, the larger one produces short last lines. Long but not completely filled lines are typeset nearly identically.

Experiment 30: Description

Show a minor difference for a long last line with these settings.

TEX output

• `\parfillskip` ← 0.75`\hsize` plus 0.25`\hsize` minus 0.75`\hsize`:

5. With enough interword glue as well as short words at the end of the 1st line the 2nd can be extended.

• `\parfillskip` ← 0.25`\hsize` plus 0.75`\hsize` minus 0.25`\hsize`:

5. With enough interword glue as well as short words at the end of the 1st line the 2nd can be extended. □

It is hard to see the difference, but the last line of the first case has badness 46, in the second paragraph the value is 6. Of course, there are several experiments which do not show a difference except for the spacing.

Experiment 31: Description

Show that some texts are not changed compared to the default and some change for both settings.

TEX output

• `\parfillskip` ← 0.75`\hsize` plus 0.25`\hsize` minus 0.75`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a

6. One line or two for this text? That is the question, or?

• `\parfillskip` ← 0.25`\hsize` plus 0.75`\hsize` minus 0.25`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?

4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a

6. One line or two for this text? That is the question, or? □

The stretchability must not be exactly `\hsize`$-x = h - x$. If the second pass is accepted for short lines the stretchability can be lowered to $(h-x)/\sqrt[3]{2}$ or to $4(h-x)/5$, if the `\tolerance` is 200; $5/4 < \sqrt[3]{2}$ but quite close, so the division with a smaller number makes the stretchability a little bit larger. In fact, the stretchability can be set to any valid value, for example, to $2h$ or 1 fil to have only decent last lines if the glue has to stretch, as only the condition $h - x > 0\,\mathrm{pt}$ for `\parfillskip`'s natural width x must hold. But such large values for the stretchability do not avoid short last lines, etc.; the only visible change might be some white space at the end of long last lines, for example, in the text of experiment 2.

Of course, the stretchability can also be quite small, it might be smaller than the difference of line width and natural width. A large natural width with a very small stretchability might absorb a short last line as such lines are underfull. A shorter natural width and enough stretchability so that no underfull lines are created has the same property.

Experiment 32: Description

Show that a smaller stretchability avoids short last lines.

TEX output

• `\parfillskip` ← 0.75`\hsize` plus 0.06`\hsize` minus 0.75`\hsize`:

4′. My keyboArd is broken; when I press an ‘A’ in lowercAse (only) the screen repeAts it four times: a a a a

6. One line or two for this text? That is the question, or?
- `\parfillskip` ← 0.25`\hsize` plus 0.6`\hsize` minus 0.25`\hsize`:

4′. My keyboArd is broken; when I press an ‘A’ in lowercAse (only) the screen repeAts it four times: a a a a
6. One line or two for this text? That is the question, or? □

The first paragraph shows that the penalization of non-short lines can lead to an overreaction. Extended lines can share the same fate.

Experiment 33: Description
Show that in a specification with a very small stretchability last lines are sometimes extended needlessly.

TeX output
- `\parfillskip` ← 0.75`\hsize` plus 0.06`\hsize` minus 0.75`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?
8. One line or two for this text? That’s the question, or not?
- `\parfillskip` ← 0.25`\hsize` plus 0.6`\hsize` minus 0.25`\hsize`:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?
8. One line or two for this text? That’s the question, or not? □

The natural width of `\parfillskip` was expressed as a fraction of h but a fixed value works too. Then the results depend on the current `\hsize`.

Experiment 34: Description
Show that a small value of natural width is sufficient.

TeX output
- `\parfillskip` ← 10 pt plus `\hsize` − 10 pt minus 10 pt:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?
4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a
6. One line or two for this text? That is the question, or?
8. One line or two for this text? That’s the question, or not?
- `\parfillskip` ← 20 pt plus 0.8`\hsize` minus 20 pt:

3. A short text in 1 line. Or has the paragraph 2 or 3 lines?
4. My keyboArd is broken. When I press the key for the lowercAse A the screen repeAts it severAl times: a a a a
6. One line or two for this text? That is the question, or?
8. One line or two for this text? That’s the question, or not? □

Note all paragraphs for the experiments in this section are typeset in the first pass, except 4′ in experiments 29 and 32 and both 8s in experiment 33.

Discussion. The definition of all three dimensions in the glue specification of `\parfillskip` combines two specifications: natural width & stretchability and natural width & shrinkability. If the natural width equals the shrinkability, and the stretchability together with the natural width never outputs underfull lines, most problems that occur with each individual specification are avoided.

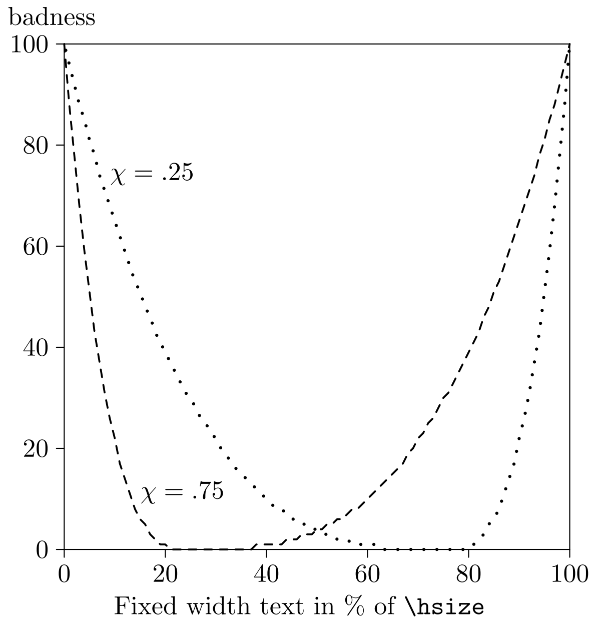

Figure 4: Graphs for $\chi = 0.25$ and $\chi = 0.75$ when `\parfillskip` $\equiv \chi$`\hsize plus` $(1-\chi)$`\hsize minus` χ`\hsize`

Table 6 lists the experiments for the different specification discussed in this section. Figure 4 has the graphs for $x = 0.25h$ and $x = 0.75h$, i.e., the natural width is once a quarter and once three quarters of `\hsize`. With a full glue specification a small value for the natural width (and the shrinkability)

Table 6: Observations for complete glue specifications

	Specification of `\parfillskip` ($h \equiv$ `\hsize`)					
	natural width	0pt	.25h	.75h	.25h	20pt
	stretch	1fil	.75h	.25h	.6h	.8h
	shrink	0pt	.25h	.75h	.25h	20pt
Observation						
1	Short line	1	(1)	29	(1)	(1)
2	Completely filled	2	(22)	(22)	(22)	(22)
3	Glue stretches	–	31	31	33	34
4	2nd pass forced	–	–	–	33	(1)
5	Underfull line	–	–	–	–	–
6	No demerits	–	–	–	–	–
7	Remove short line	–	29	–	32	34
8	Extend last line	–	31	31	33	34
9	Backspaces	–	–	–	–	–
10	Add short line	–	–	29	–	–
11	Glue shrinks early	–	30	30	32	34
12	Overfull line	–	–	–	–	–
13	Stick out right	–	–	–	–	–

Legend: −/ /+: never/don’t care/always
$n/(n)$: (implicitly) shown in example number n

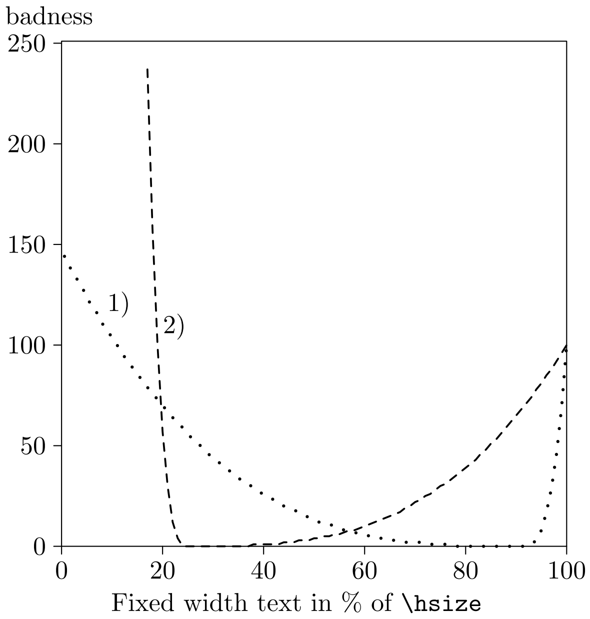

Figure 5: Graphs of two `\parfillskip` specifications
1) `20 pt plus 0.8\hsize minus 20 pt`
and 2) `0.75\hsize plus 0.06\hsize minus 0.75\hsize`

assigns to short lines higher badness values than a large value for the natural width does. The first setting prefers nearly filled lines over short lines, i.e., recommendation (1) is followed more often than recommendation (2). On the other hand the small value needs much more text in the last line before the last line gets decent and the interword glue stretches less than 50%. Figure 5 shows two specifications which might force a second pass; one does not obey the tolerance in all cases.

Summary. Setting all dimensions of the specification to a nonzero value creates a complex scenario which applies two different specifications to short and long lines. The results of the previous sections suggest equality for natural width and shrinkability and to assign so much stretchability that, together with the natural width, it reaches the `\hsize` (in either the first or the second pass). A small value for the natural width works better than a large one although spaces in very short lines stretch a little bit more.

11 A closer look at a few specifications

Let's look at the specifications for `\parfillskip` that are mentioned in the summaries of the previous sections. The experiments in this article are quite short and a setting of `\parfillskip` that makes last lines longer works much better when there are more sets of valid line breaks; paragraphs with plenty of text usually have a greater selection of possible line breaks. Nevertheless, a general specification must be able to handle one- or two-line paragraphs.

I do not list a specification that might throw an error. An author has to decide if a forced second pass based only on the setting of `\parfillskip` is acceptable. Such settings are often successful when the first pass fails to lengthen the last line or to absorb a short last line into the former penultimate line. Values smaller than 0.8`\hsize` for the stretchability and a value smaller than `\hsize` for natural width and shrinkability are possible but then underfull lines and other problems might occur. And it seems questionable to make TeX choose line breaks that output a paragraph that has several thousand demerits more than the optimum instead of giving a hint that the last line is short.

1) `0pt plus 0.8\hsize` (Fig. 1): Instead of 1 fil a finite stretchability is used. Spaces in the last line stretch; in a very short line they stretch more than the specified stretchability as TeX might execute a second pass.

2) `\hsize minus \hsize` (Fig. 3): With this value, spaces in the last line shrink. A new short last line might be produced to avoid a completely filled last line, but such a line is very unlikely.

3) `20pt plus 0.8\hsize minus 20pt` (Fig. 5 with *TUGboat*'s `\hsize`): The specification makes completely filled lines unlikely although not impossible. For short lines it acts similarly to case 1) but for long last lines glue shrinks early and, unless badness 100 is reached, some white space at the end of the line is set.

Comparison to other suggestions. In section 2 some specifications for `\parfillskip` suggested in the literature are listed. How do these compare to the above recommendations? Two of them are not meant for general use in documents but in special cases so a comparison is not quite fair.

The specification `0pt plus 0.7\hsize` creates the same output for the experiments as case 1) except that artificial demerits are applied in experiments 1 and 13 as this specification plays with invalid badness values. Case 2) and the non-general specification `2em plus 1fil` produce the same typeset output except experiments 2′ and 15 generate overfull lines in the latter case.

The glue specification with three dimensions, interpreted as `0.33\hsize plus 0.67\hsize minus 0.33\hsize`, used if `\looseness` is set, behaves similarly to case 3) except that the former prefers additional lines. So experiments 2, 4, and 6 are typeset with one more line. The other complete glue specification, `0.75\hsize plus 0.06\hsize minus`

`0.75\hsize`, also behaves similarly to case 3) with wider white space at the end of lines. Further, experiments 1 and 13 produce artificial demerits because the stretchability is so small; and there is a noticeable difference for text 8 (see experiment 33).

The last specification has the most complex assignment. Informally it is `\hsize` − 1.5`\parindent` `minus` `\hsize` − 1.5`\parindent` − `1em`. This specification produces an overfull line for experiment $2'$ and artificial demerits in experiments 1, 2, and 13. In experiment $4'$ it absorbs the short line (typeset as the first version shown in experiment 32) otherwise in experiments 4, 6, and 8 last lines are lengthened.

How much does a single space stretch? Several times in the above analysis the stretching of single spaces in a short last line was mentioned. With experiment 13 this can be visualized; of course with the extra stretchability after a period. The specification `0pt plus 5.17\hsize` used in Fig. 2 is included in the comparison.

Experiment 35: Description
Show how much a single space stretches in a short line.

TEX output

- `\parfillskip` ← `\hsize` minus `\hsize`:

13. a

- `\parfillskip` ← 2 em plus 1fil:

13. a

- `\parfillskip` ← 0 pt plus 5.17`\hsize`:

13. a

- `\parfillskip` ← 0.33`\hsize` plus 0.67`\hsize` minus 0.33`\hsize`:

13. a

- `\parfillskip` ← 20 pt plus 0.8`\hsize` minus 20 pt:

13. a

- `\parfillskip` ← 0 pt plus 0.8`\hsize`:

13. a

- `\parfillskip` ← 0 pt plus 0.7`\hsize`:

13. a

- `\parfillskip` ← 0.75`\hsize` plus 0.06`\hsize` minus 0.75`\hsize`:

13.   a

- `\parfillskip` ← 1`\hsize` − 1.5`\parindent` minus 1`\hsize` − 1.5`\parindent` − 1 em:

13.   a ▯

The last four specifications have artificial demerits, and the last two produce underfull lines.

12 Summary

A change of `\parfillskip`'s specification does not spare an author the need to proofread to check if the last lines of paragraphs are shorter than the indentation or fill the line width.

There is another point on the checklist if finite dimensions are used for `\parfillskip` and the author has not always entered `\hfil\ $$` to start display math mode: If a short last line that contains one space precedes a display then too much vertical space might be used by TEX, as it is forced to apply `\abovedisplayskip` and `\belowdisplayskip`.

Paragraphs with indentation. With the default settings of plain TEX the specification of the glue `\parfillskip` has one typographical problem: Last lines in paragraphs can be shorter than `\parindent`, which has the value 20 pt by default. A reader spots the start of a paragraph easily because of the wide indentation. Thus, completely filled last lines generate no problem for the readability.

If an author wants to avoid last lines of a few characters the simplest way is to use a tie to keep the last short word connected to the text or an hbox to avoid the hyphenation of a word that can leave a fragment of three letters on the last line. If TEX cannot find an acceptable set of line breaks with such input an overfull line message informs the author to rewrite the text or to change TEX's parameters. Note that the ties and the hboxes can make TEX execute a second pass although the default setting would need only a first pass. Thus a longer last line is traded in such cases for higher badness values in other lines and/or hyphens.

To prevent TEX from typesetting short last lines without hints in the input, `\parfillskip`'s specification has to change. For *TUGboat*, the best choice from section 11 seems to be `20pt plus 0.8\hsize minus 20pt`. For other line widths the constant needs to be adjusted.

Paragraphs without indentation. A text that does not signal the start of a paragraph by white space, i.e., through indentation or an empty line between paragraphs, or through other methods, for example, p. 40 in [3] names ornaments and outdented paragraphs, relies on white space at the end of the last line to indicate the end of the paragraph. In this scenario a completely filled last line creates a problem. With plain TEX such lines are possible; see experiment 2.

To avoid a completely filled last line the natural width must make a nonzero contribution. The simple addition of natural width to the default setting of `\parfillskip`, for example, `10pt plus 1fil`, seems to be a valid solution although now overfull lines are possible. A setting of `\parfillskip` in which the natural width equals the shrinkability makes a completely filled last line not quite impossible, but very unlikely, and without generating overfull lines, for example, `20pt plus 0.8\hsize minus 20pt`.

Which setting was used for this article? I decided to make an experiment and to use the macros shown in Fig. 2, that is, at the end of a paragraph `\parfillskip` is set to `0pt plus 5.17\hsize`. It is the first time that I applied these macros. As the *TUGboat* format uses a large indent I was only interested in avoiding short last lines and accepted completely filled ones. After the text became reasonably stable I activated the macro that checks the length of the last lines except in the experiments (verbatim parts and output), figures, tables, and the list of references. Initially, the macro gave an error six times: In two cases I changed the text, in one case the last word was placed in an hbox, and three last lines were accepted as they are false hits, i.e., the last line is tight or long enough in front of a heading. Later revisions changed some of these lines again.

The text contains several formulas and I enter display math to show them. The macros of Fig. 2 use finite stretchability for `\parfillskip` only for the end of a paragraph; otherwise the default setting is applied. So there is no problem with the vertical space around displays.

References

[1] Paul W. Abrahams, Kathryn A. Hargreaves, Karl Berry, *File* `macros.tex` *of package* `impatient`. `ctan.org/tex-archive/info/impatient`

[2] Wolfgang Appelt: *TEX für Fortgeschrittene*, Bonn, Germany: Addison-Wesley, 1988.

[3] Robert Bringhurst, *The Elements of Typographic Style*, 4th edition, version 4.2, Seattle, Washington: Hartley & Marks, 2016.

[4] Anne Brüggemann-Klein, "Re: Obtaining a funny paragraph shape in TeX", *TEXhax Digest* **89**:42 (1989), 28 April 1989. `ctan.org/tex-archive/info/digests/texhax/89/texhax.42.gz`

[5] David Carlisle (based on a `comp.text.tex` article of Peter Schmitt), "Russian Paragraph Shapes", *Baskerville* **6**:1 (1996), 13–15. `uk.tug.org/wp-installed-content/uploads/2008/12/61.pdf`

[6] *The Chicago Manual of Style*, 15th edition, Chicago, Illinois: University of Chicago Press, 2003.

[7] Jean-luc Doumont, *Trees, maps, and theorems*, Belgium, 2009. `treesmapsandtheorems.com`

[8] Jean-luc Doumont, "Quantum space: Designing pages on grids", *TUGboat* **31**:2 (2010), 248. `principiae.be/pdfs/TUG-X-004-slideshow.pdf` `zeeba.tv/quantum-spaces-designing-pages-on-grids`

[9] Friedrich Forssman & Ralf de Jong, *Detailtypographie*, 3rd edition, Mainz, Germany: Verlag Hermann Schmidt, 2004.

[10] Donald E. Knuth and Michael F. Plass, "Breaking paragraphs into lines", *Software — Practice and Experience* **11** (1981), 1119–1184; reprinted with an addendum as Chapter 3 in [14], 67–155.

[11] Donald E. Knuth, *The TEXbook*, Volume A of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1984.

[12] Donald E. Knuth, *TEX: The Program*, Volume B of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986.

[13] Donald E. Knuth, *Literate Programming*, Stanford, California: Center for the Study of Language and Information, CSLI Lecture Notes No. 27, 1992.

[14] Donald E. Knuth, *Digital Typography*, Stanford, California: Center for the Study of Language and Information, CSLI Lecture Notes No. 78, 1999.

[15] Donald Knuth, "TEX's infinite glue is projective", *TUGboat* **28**:1 (2007), 4. `tug.org/TUGboat/tb28-1/tb88knut.pdf`

[16] Frank Mittelbach, "E-TEX: Guidelines for Future TEX Extensions", *TUGboat* **11**:3 (1990), 337–345. `tug.org/TUGboat/tb11-3/tb29mitt.pdf`

[17] Walter Schmidt, "Absätze — einmal anders", *Die TEXnische Komödie* **9**:2 (1997), 19–22. `dante.de/DTK/Ausgaben/komoedie19972.pdf`

[18] Philip Taylor, "Book Design for TEX Users — Part 2: Practice", *TUGboat* **20**:4 (1999), 378–389. `tug.org/TUGboat/tb20-4/tb65tay2.pdf`

[19] Udo Wermuth, "Tracing paragraphs", *TUGboat* **37**:3 (2016), 358–373; Errata in *TUGboat* **38**:3 (2017), 414. `tug.org/TUGboat/tb37-3/tb117wermuth.pdf`

[20] Udo Wermuth, "A note on `\linepenalty`", *TUGboat* **38**:3 (2017), 400–414; Errata in *TUGboat* **39**:1 (2018), 87. `tug.org/TUGboat/tb38-3/tb120wermuth.pdf`

[21] Udo Wermuth, "TEX's 'additional demerits' parameters", *TUGboat* **39**:1 (2018), 81–87. `tug.org/TUGboat/tb39-1/tb121wermuth-adem.pdf`

[22] Peter Wilson, "Glisterings", *TUGboat* **28**:2 (2007), 229–232. `tug.org/TUGboat/tb28-2/tb89glister.pdf`

[23] Peter Wilson, "Glisterings", *TUGboat* **29**:2 (2008), 324–325. `tug.org/TUGboat/tb29-2/tb92glister.pdf`

[24] Peter Wilson, "Glisterings", *TUGboat* **38**:3 (2017), 338–341. `tug.org/TUGboat/tb38-3/tb120glister.pdf`

◇ Udo Wermuth
Dietzenbach, Germany
`u dot wermuth (at) icloud dot com`