## The optimal value for `\emergencystretch`

Udo Wermuth

### Abstract

As a reaction to authors using very high values for the integer parameter `\tolerance` in their texts instead of rewriting them if TeX creates overfull lines, Donald E. Knuth introduced in TeX 3.0 the dimen `\emergencystretch` and a third pass for TeX's line-breaking algorithm. The parameter should only be used in an emergency situation but such a situation can occur, for example, if a typist cannot change the text written by an author and TeX produces overfull lines. Then this parameter comes to the rescue although the output might not look good, as spaces can spread much more than before. This article tries to find all factors that have an impact on the value of `\emergencystretch`. Besides pure theory, experiments are performed and the impact factors are briefly discussed.

### 1 Introduction

Since its introduction with TeX 3.0 in 1989 [8, p. 327] the parameter `\emergencystretch` has asked for attention as no hints about useful values are given in the article. Of course, it could be used to avoid awful lines in a better way than before: "[P]eople [...] tried to do this by setting `\tolerance` = 10000, but the result was terrible because TeX would tend to consolidate all the badness in one truly horrible line." So its usage is a "better way to avoid overfull boxes, for people who don't want to look at their documents to fix unfeasible line breaks manually."

*The TeXbook* describes this dimen [3, p. 107] and gives its default value in the `plain` format, 0 pt. I assume that — whatever research went into the design and the implementation — the idea remains that authors should rewrite their texts if they cannot be broken by TeX. The dimen is an instrument for people who have to work with a text which cannot be changed by them.

The number of relevant publications about this parameter appears to be very small. In the first years after its introduction the LaTeX3 group looked for volunteers to make experiments with the parameter [12, p. 511]. But it seems that no one has published such experiments or research. A check with several well-known textbooks to find a hint about useful settings and not merely the description of this parameter resulted in just one hit: In [1], p. 333, it is written that "a likely value seems to be around 5 pt." A search in *TUGboat* archives produces again only one hit: In [13], p. 139, the following statement is made: "However, a sound rule of thumb is to set it to 1 em (based on the primary text font); this value, strange as it may seem, appears equally suitable for both wide and narrow measures." (His "pragmatic approach" can result in high values for `\tolerance`.)

Of course a single value cannot always solve the problem and make TeX find line breaks without overfull lines. It might nevertheless be seen as an upper bound on what is seen as tolerable for the visual output. A value of 5 pt appears to be small but in a line with only one stretchable interword space it creates a hole in the text as it stretches the space to twice its maximal value in `cmr10`. Therefore, whatever value for the parameter `\emergencystretch` is used, the output should be checked to see if it does not create awful looking lines. The author of a text should think about a textual change before eliminating overfull lines with `\emergencystretch`.

Here is a short overview of the structure of this article: Section 2 sketches briefly and somewhat vaguely aspects of glue and establishes the notation that helps to write about spaces and stretchability. Section 3 presents an experiment with many overfull lines and shows how they can be removed by assigning appropriate values to the dimension `\emergencystretch`. The situation is analyzed from a theoretical point of view in section 4. Section 5 applies the formula created in section 4 to some cases of experiment 1 and presents numerical results as well as ideas for TeX macros to do the calculations. The next three sections look at the various parameters that occur in the formula for `\emergencystretch` stated in section 4. The theoretical results are extended in section 9 and applied to a second experiment in section 10. Removing overfull lines is only one application of `\emergencystretch`; another is discussed in section 11. The last section summarizes the results and gives a rule of thumb for the calculation of `\emergencystretch`.

### 2 Notational conventions

A paper about the dimension `\emergencystretch` should explain the material that is able to stretch or shrink: glue. Therefore the basic principles of glue in texts and in math mode are reviewed first. To handle all the different forms a consistent notation for numbers, dimensions, and skips is needed. So let's start with some notational conventions and definitions.

**Conventions.** I use the following notation for variables and functions; variables are written in lowercase, functions in uppercase letters:

1. math italic Latin for dimensions, the most frequently used variables and functions;
2. numbers are written with Greek letters;
3. boldface Latin is used for glue, i.e., a triple of dimensions, for example, $\mathbf{g} = (g^\circ, g^+, g^-)$ for glue, that has the natural width $g^\circ$ with the ability to stretch by $g^+$ and to shrink by $g^-$;
4. boldface Greek is used for pairs of numbers and triples of such pairs.

A typewriter font is used for the input: String variables are written in uppercase and single command or character input get letters in lowercase.

When glue is added the values of the corresponding components are added; multiplication and division by an integer is also done by multiplying or dividing each component by that integer. These operations are represented in TeX by the commands `\advance`, `\multiply`, and `\divide` for skips.

Some important elements receive fixed names:

- $e$ is the value of `\emergencystretch`;
- $f_\nu$ denotes the `\fontdimen` $\nu$ of the current font;
- $h$ stands for the `\hsize`;
- $k$ represents a kern;
- $m$ is the current value of `\mathsurround` at the end of a formula, i.e., at the math-shift character that ends the math mode;
- $o$ is the value by which an overfull line is too wide;
- $\epsilon$ specifies the *environmental condition* that explains how spaces get their width (see below);
- $\tau$ represents the value of `\tolerance`;
- $\pi$ stands for a penalty;
- $\mathbf{l} = (l^\circ, l^+, l^-)$ is the `\leftskip`;
- $\mathbf{r} = (r^\circ, r^+, r^-)$ is the `\rightskip`;
- $\mathbf{s} = (s^\circ, s^+, s^-)$ represents the `\spaceskip`;
- $\mathbf{x} = (x^\circ, x^+, x^-)$ stands for the `\xspaceskip`;
- $\mathbf{z} = (0\,\mathrm{pt}, 0\,\mathrm{pt}, 0\,\mathrm{pt})$ is the zero glue; in the `plain` format it is called `\z@skip`;
- $\mathbf{Z} = (0, 0)$ is the pair of two zeros.

Important functions are:

$W(\mathtt{T})$ is the width of the input `T` with an unspecified stretch or shrink amount; otherwise the subscript "nw" or "mt" are used for the natural width and the maximal tight width, resp.;

$L(\mathtt{S}\&\mathtt{T})$ stands for the change of width that the concatenation of `S` and `T` differs from the sum of the individual widths because of ligatures or kerning, i.e., $W_{\mathrm{nw}}(\mathtt{ST}) = W_{\mathrm{nw}}(\mathtt{S}) + W_{\mathrm{nw}}(\mathtt{T}) + L(\mathtt{S}\&\mathtt{T})$;

$\Phi(\mathtt{T})$ returns the value of the space factor that is active at the end of input `T`;

$\boldsymbol{\Omega}(\mathtt{T})$ stands for a triple of pairs — each pair gives the amounts of stretchability and shrinkability of one of the three infinite glue orders in input `T`;

$\mathbf{G}_\epsilon(\sigma, \mathtt{W})$ is the finite glue that stems from the input `W` for white space, given that the space factor is $\sigma$ at the start of the input;

$\mathbf{S}_\epsilon(\mathtt{T})$ represents all finite glue contained in `T`.

The first three functions should be clear enough from the given description. They are easily computed by TeX. The last two functions compute glue. Together with the fourth function they are explained and precisely defined in the next subsections together with a few more specialized functions.

The rest of this section describes and defines these functions; skip to section 3 if you are not interested in the details.

**Glue in paragraphs.** In general an input `T` can be seen as a sequence of text parts $\mathtt{T}_\kappa$, which do not have any author-entered glue except if it is already set, for example, in an `\hbox`, and input $\mathtt{W}_\kappa$ that represents the glue $w_\kappa^\circ$ plus $w_\kappa^+$ minus $w_\kappa^-$:

$$\mathtt{T} = \mathtt{T}_0\mathtt{W}_0\mathtt{T}_1\mathtt{W}_1\ldots\mathtt{T}_{\omega-1}\mathtt{W}_{\omega-1}\mathtt{T}_\omega. \qquad (*)$$

The text elements $\mathtt{T}_\kappa$ contain more than characters that are typeset, for example, TeX control sequences, implicit kerns, mathematics, or boxes. An assumption is made: The input can be processed unconditionally, that means control sequences like `\if`, `\unkern`, etc. are resolved and token lists are expanded. This is not a real restriction but avoids subcases and is reasonable for user input. Each entered white space sequence is related to a glue item $\mathtt{W}_\kappa = \mathtt{w}_{\kappa,1}\mathtt{w}_{\kappa,2}\ldots\mathtt{w}_{\kappa,\mu_\kappa}$, whose elements might be not only spaces but also penalties and dimens from a kern or a math-on/math-off switch. For example, the author might have entered an italic correction $\mathtt{w}_{\kappa,1}$ and then a space $\mathtt{w}_{\kappa,2}$: The natural width of the glue item $w_\kappa^\circ$ is the sum of the kern and the natural width of a space. Penalties are also allowed in $\mathtt{W}_\kappa$, though they are not white space. Then a kern followed by a tie is covered by the description too.

If leaders are used then the skip part is listed as glue input, since the box part just fills the created white space with some pattern [3, p. 223].

**The glue function.** A function is needed to find a unified way to write about finite glue in a text. TeX has several mechanisms to deal with it [3, pp. 75–77] which result in many different cases in equations about glue. Therefore, the *environmental condition*, called $\epsilon$, is introduced. It is a number defined as

$$\epsilon := \begin{cases} 0, & \mathbf{s} = \mathbf{z} = \mathbf{x}; \\ 1, & \mathbf{s} = \mathbf{z} \neq \mathbf{x}; \\ 2, & \mathbf{s} \neq \mathbf{z} = \mathbf{x}; \\ 3, & \mathbf{s} \neq \mathbf{z} \neq \mathbf{x}. \end{cases}$$

So $\epsilon = 0$ means that the normal interword spaces are used, odd values stand for a nonzero `\xspaceskip`,

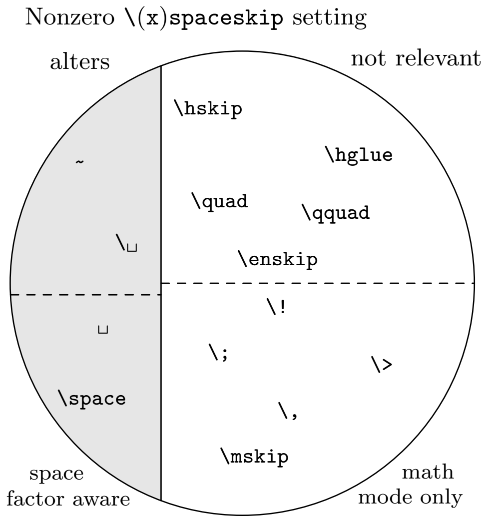


**Figure 1:** Input methods in `plain` TeX for glue influenced by nonzero `\(x)spaceskip` (shaded area) and glue that is space factor-aware (lower left area)

and values 2 and 3 signal that the `\spaceskip` is nonzero.

TeX knows several kinds of white space. Some are aware of the environment, while others ignore it; let's call the former kind *eaglue*. For example, the normal spaces, the control word `\space`, the control space and the tie belong to eaglue but author-entered white space through `\hskip` is not eaglue. Further, normal spaces and control spaces are different as the first is affected by the space factor before the white space [3, p. 76]. To distinguish between these two kinds of eaglue the first one is called *sfglue* in the following discussion.

Figure 1 gives an overview of the different kinds of finite glue, including muglue which is used only in math mode. The TeX control sequences for infinite glue, such as `\hfil`, are not listed, since such glue is collected by the function $\mathbf{\Omega}$ described below. Technically, not only glue counts, although all values are treated as glue in this discussion. A user can enter an explicit kern with the italic correction, or TeX inserts the value of `\mathsurround` when it reads a math-shift character.

It is possible to have several glue items in a sequence; sometimes this second glue is ignored and sometimes it counts: Two `\hskips` create glue in which each component is the sum of the corresponding values of the skips, but two normal spaces count usually only as a single space. When white space is placed in a sequence the space factor must be considered for all items. For example, in `:\hglue0pt plus1pt \space\space` the last `\space` is still influenced by the space factor that occurs after the colon.

So it requires several cases to define a function that returns the glue value for any glue item $\mathtt{w}$ when the space factor $\sigma$ is applied under the environmental condition $\epsilon$:

$$\mathbf{G}_\epsilon(\sigma,\mathtt{w}) := \begin{cases} (f_2, \sigma f_3/1000, 1000 f_4/\sigma), \\ \quad \mathtt{w} \text{ is sfglue, } \epsilon < 2,\ \sigma < 2000; \\ (s^\circ, \sigma s^+/1000, 1000 s^-/\sigma), \\ \quad \mathtt{w} \text{ is sfglue, } \epsilon = 2; \\ (s^\circ, \sigma s^+/1000, 1000 s^-/\sigma), \\ \quad \mathtt{w} \text{ is sfglue, } \epsilon = 3,\ \sigma < 2000; \\ (f_2 + f_7, \sigma f_3/1000, 1000 f_4/\sigma), \\ \quad \mathtt{w} \text{ is sfglue, } \epsilon = 0,\ \sigma \ge 2000; \\ (x^\circ, \sigma x^+/1000, 1000 x^-/\sigma), \\ \quad \mathtt{w} \text{ is sfglue, } \epsilon \text{ is odd, } \sigma \ge 2000; \\ (f_2, f_3, f_4), \\ \quad \mathtt{w} \text{ is eaglue, not sfglue, } \epsilon < 2; \\ (s^\circ, s^+, s^-), \\ \quad \mathtt{w} \text{ is eaglue, not sfglue, } \epsilon \ge 2; \\ (w^\circ, v^+, v^-), \\ \quad \mathtt{w} \text{ is explicitly entered glue,} \\ \quad \texttt{\textbackslash hskip} \text{ or the like;} \\ \quad v^+ := w^+ \text{ and } v^- := w^- \text{ if they} \\ \quad \text{are finite, otherwise } 0\,\text{pt}; \\ (k, 0\,\text{pt}, 0\,\text{pt}), \\ \quad \mathtt{w} \text{ is a kern or mkern with width } k; \\ (m, 0\,\text{pt}, 0\,\text{pt}), \\ \quad \mathtt{w} \text{ is a math-shift character; } m \\ \quad \text{is the value of } \texttt{\textbackslash mathsurround}; \\ (0\,\text{pt}, 0\,\text{pt}, 0\,\text{pt}), \\ \quad \mathtt{w} \text{ is ignored or not white space.} \end{cases}$$

As usual the three components of $\mathbf{G}_\epsilon(\sigma,\mathtt{w})$ have the names $G^\circ_\epsilon(\sigma,\mathtt{w})$, $G^+_\epsilon(\sigma,\mathtt{w})$, and $G^-_\epsilon(\sigma,\mathtt{w})$.

Before the sum of the individual parts of $\mathtt{W}_\kappa = \mathtt{w}_{\kappa,1}\mathtt{w}_{\kappa,2}\ldots\mathtt{w}_{\kappa,\mu_\kappa}$ can be built, one technicality needs to be addressed: The value of `\spacefactor` can be changed. When $\mathtt{w}_{\kappa,0}$ is the empty string, the definition is

$$\mathbf{G}_\epsilon(\Phi(\mathtt{T}_\kappa), \mathtt{W}_\kappa) := \sum_{\iota=1}^{\mu_\kappa} \mathbf{G}_\epsilon(\Phi(\mathtt{T}_\kappa \mathtt{w}_{\kappa,1}\ldots\mathtt{w}_{\kappa,\iota-1}), \mathtt{w}_{\kappa,\iota}).$$

To state the amount of stretchability or shrinkability of the input $\mathtt{T}$ of the model of eq. (∗) in one function, the following summations are necessary:

$$\sum_{\kappa=0}^{\omega-1} G^+_\epsilon(\Phi(\mathtt{T}_\kappa), \mathtt{W}_\kappa) \text{ and } \sum_{\kappa=0}^{\omega-1} G^-_\epsilon(\Phi(\mathtt{T}_\kappa), \mathtt{W}_\kappa).$$

This looks very complicated, but is a simple function if the default settings of `plain` TeX are considered. In this case only a counting of spaces and punctuation marks in a single line $\mathtt{L}$ is necessary if the author hasn't entered `\hskip`s. Let

$$\nu_{\mathtt{c}}(\mathtt{L}) := \text{number of occurrences of character } \mathtt{c} \text{ in } \mathtt{L}$$

The optimal value for `\emergencystretch`

and define an extension for more than one character

$$\nu_{\mathtt{c}_1 \ldots \mathtt{c}_\omega}(\mathtt{L}) := \sum_{\kappa=1}^{\omega} \nu_{\mathtt{c}_\kappa}(\mathtt{L}).$$

The counts of the "relevant" punctuation marks, i.e., those punctuation marks not preceded by an uppercase letter, not at end of line, and not followed by a control space or tie, and the `plain` TeX settings give total stretchability and shrinkability:

$$\Big(\nu_{␣}(\mathtt{L}) + \frac{5}{4}\nu_{\mathtt{,}}(\mathtt{L}) + \frac{3}{2}\nu_{\mathtt{;}}(\mathtt{L}) + 2\nu_{\mathtt{:}}(\mathtt{L}) + 3\nu_{\mathtt{.?!}}(\mathtt{L})\Big) f_3,$$

$$\Big(\nu_{␣}(\mathtt{L}) + \frac{4}{5}\nu_{\mathtt{,}}(\mathtt{L}) + \frac{2}{3}\nu_{\mathtt{;}}(\mathtt{L}) + \frac{1}{2}\nu_{\mathtt{:}}(\mathtt{L}) + \frac{1}{3}\nu_{\mathtt{.?!}}(\mathtt{L})\Big) f_4,$$

where $\nu_{␣}(\mathtt{L})$ numbers the spaces that are not preceded by a relevant punctuation mark in `L`. Note the terms $f_3$ and $f_4$, which are `\fontdimen`s 3 and 4, are replaced by $s^+$ and $s^-$ if $\epsilon = 2$.

**Infinite glue.** Inspired by [4, §822], infinite glue is written as a triple of pairs of numbers. Each pair represent one of the three orders of infinite glue: fil, fill, and filll. So the amount of infinite glue in `g` is $\boldsymbol{\Omega}(\mathtt{g}) = (\boldsymbol{\Omega}_1(\mathtt{g}), \boldsymbol{\Omega}_2(\mathtt{g}), \boldsymbol{\Omega}_3(\mathtt{g}))$ and each pair has two numbers: $\boldsymbol{\Omega}_\iota(\mathtt{g}) = (\Omega_\iota^+(\mathtt{g}), \Omega_\iota^-(\mathtt{g}))$.

For example, the glue specification $\mathtt{g} = $ `\hskip` $\chi$ pt `plus` $2\alpha$ `fil minus` $-\beta$ `fill` has two different orders of infinite glue, therefore $\boldsymbol{\Omega}_1(\mathtt{g}) = (2\alpha, 0)$, $\boldsymbol{\Omega}_2(\mathtt{g}) = (0, -\beta)$, and $\boldsymbol{\Omega}_3(\mathtt{g}) = \mathbf{Z}$.

To define for `T` of eq. $(*)$

$$\boldsymbol{\Omega}(\mathtt{T}) = (\boldsymbol{\Omega}_1(\mathtt{T}), \boldsymbol{\Omega}_2(\mathtt{T}), \boldsymbol{\Omega}_3(\mathtt{T}))$$

a definition of $\boldsymbol{\Omega}_\iota(\mathtt{T})$ for $1 \le \iota \le 3$ is needed. With

$$\Omega_\iota^+(\mathtt{w}) := \begin{cases} \alpha, & \text{if } \mathbf{w} \text{ contains the amount } \alpha \\ & \text{of infinite stretchability of order } \iota \\ 0, & \text{otherwise} \end{cases}$$

$$\Omega_\iota^-(\mathtt{w}) := \begin{cases} \alpha, & \text{if } \mathbf{w} \text{ contains the amount } \alpha \\ & \text{of infinite shrinkability of order } \iota \\ 0, & \text{otherwise} \end{cases}$$

$\boldsymbol{\Omega}_\iota(\mathtt{W}_\kappa)$ when $\mathtt{W}_\kappa = \mathtt{w}_{\kappa,1}\mathtt{w}_{\kappa,2} \ldots \mathtt{w}_{\kappa,\mu_\kappa}$ is defined as

$$\begin{aligned} \boldsymbol{\Omega}_\iota(\mathtt{W}_\kappa) &= (\Omega_\iota^+(\mathtt{W}_\kappa), \Omega_\iota^-(\mathtt{W}_\kappa)) \\ &:= \left( \sum_{\chi=1}^{\mu_\kappa} \Omega_\iota^+(\mathtt{w}_{\kappa,\chi}), \sum_{\chi=1}^{\mu_\kappa} \Omega_\iota^-(\mathtt{w}_{\kappa,\chi}) \right) \end{aligned}$$

so that

$$\begin{aligned} \boldsymbol{\Omega}_\iota(\mathtt{T}) &= (\Omega_\iota^+(\mathtt{T}), \Omega_\iota^-(\mathtt{T})) \\ &:= \left( \sum_{\kappa=0}^{\omega-1} \Omega_\iota^+(\mathtt{W}_\kappa), \sum_{\kappa=0}^{\omega-1} \Omega_\iota^-(\mathtt{W}_\kappa) \right). \end{aligned}$$

**Glue in math mode.** TeX is often used to typeset mathematics, so the glue that is present in math mode should be analyzed too. Only inline formulas are treated here. (Overfull lines in display math mode are a different topic, not handled in this article; see [3], pp. 195–197.)

When TeX operates in math mode the space factor and normal spaces have no meaning: TeX inserts the spacing according to its own rules.

In math mode TeX typesets the formulas in *styles*. Eight styles are defined: the four basic styles are scriptscript style *SS*, script style $S$, text style $T$, and display style $D$, each with two versions [3, pp. 140–141]. They are usually sorted as

$$SS' < SS < S' < S < T' < T < D' < D$$

so that it makes sense to represent them by the numbers 0 to 7. Inside the formulas TeX operates with *atoms*. There are thirteen types, but only eight are important for spacing as the others are transformed into these eight (see [3], p. 158 and Appendix G). Again the types are sorted

$$\begin{aligned} &\text{Ord} < \text{Op} < \text{Bin} < \text{Rel} \\ &\quad < \text{Open} < \text{Close} < \text{Punct} < \text{Inner} \\ &\qquad < \text{Over} < \text{Under} < \text{Acc} < \text{Rad} < \text{Vcent} \end{aligned}$$

so that the numbers 0 to 12 can be assigned to the atoms. The names stand for atoms of types ordinary, large operator, binary operator, relation, opening, closing, punctuation, inner, overline, underline, accented, radical, and vcenter; examples of atoms for the first eight types are given in Fig. 2, first column. Note, however, it is not the symbol that counts but its usage: In `$+1$` the '+' is not a binary operator and therefore not of type Bin.

In math mode TeX doesn't use single fonts but font triples combined in a *family*; there are up to 16 families. A triple of fonts consists of a font for normal text symbols, one for subscripts, and one for sub-subscripts; the first font is called the `\textfont`. In `plain` TeX, family 1 contains the math italic letters, family 2 the math symbols, and family 3 large symbols. The fonts in families 2 and 3 need special `\fontdimen` parameters but only one of them is of interest for our glue concerns: `\fontdimen6` of family 2's `\textfont` (named $f_6^{t2}$), the quad width of the font, plays an important role. It is used to convert the *muglue*, which is measured in mu, into glue measured in pt: $1\,\text{mu} = f_6^{t2}/18$.

As mentioned above, TeX inserts glue in math mode and ignores normal white space but, of course, an author can explicitly enter `\hskip` or `\kern` or `\mskip` or `\mkern` commands; the latter two use muglue. In the model $(*)$ each explicitly entered glue is represented by a $\mathtt{W}_\kappa$ but inserted glue in math mode does not create such a glue token. The input $\mathtt{T}_\kappa$ may contain material in math mode which stretches or shrinks because of glue inserted by TeX.

| | | | Insert the specified glue after Atom if it is followed by an atom of type | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Ex. | Atom | type | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $a$ | Ord | 0 | | $\mathbf{n}^1$ | $\mathbf{m}$ | $\mathbf{t}$ | | | | $\mathbf{n}$ |
| $\sum$ | Op | 1 | $\mathbf{n}^1$ | $\mathbf{n}^1$ | | $\mathbf{t}$ | | | | $\mathbf{n}$ |
| $+$ | Bin | 2 | $\mathbf{m}$ | $\mathbf{m}$ | | | $\mathbf{m}$ | | | $\mathbf{m}$ |
| $=$ | Rel | 3 | $\mathbf{t}$ | $\mathbf{t}$ | | | $\mathbf{t}$ | | | $\mathbf{t}$ |
| ( | Open | 4 | | | | | | | | |
| ) | Close | 5 | | $\mathbf{n}^1$ | $\mathbf{m}$ | $\mathbf{t}$ | | | | $\mathbf{n}$ |
| : | Punct | 6 | $\mathbf{n}$ | $\mathbf{n}$ | | $\mathbf{n}$ | $\mathbf{n}$ | $\mathbf{n}$ | $\mathbf{n}$ | $\mathbf{n}$ |
| $\frac{1}{2}$ | Inner | 7 | $\mathbf{n}$ | $\mathbf{n}^1$ | $\mathbf{m}$ | $\mathbf{t}$ | $\mathbf{n}$ | | $\mathbf{n}$ | $\mathbf{n}$ |

[1] applied in styles 0–7; otherwise 4–7 only

**Figure 2:** TEX's *space table* [3, p. 170], with examples for the atoms

There are three amounts of glue that TEX inserts into formulas:

- **n** is the glue created by the command `\mskip\thinmuskip` in pt, i.e., it is muglue expressed in pt;
- **m** is like **n** but `\medmuskip` is used;
- **t** is like **n** but `\thickmuskip` is used.

What type gets inserted is determined by the *space table* shown in Fig. 2. Most glue is inserted only in styles $T'$, $T$, $D'$, and $D$.

The glue of these three types is inserted around selected atoms only if certain conditions are met. So first a counting of atom pairs is defined:

$\mu(\chi, \alpha_1\alpha_2, \mathtt{T}) :=$ number of times that the atom sequence $\alpha_1\alpha_2$ occurs in style $\chi$ inside text $\mathtt{T}$.

The *bracket notation* (a.k.a. *Iverson's convention*)

$$[\,\text{statement}\,] := \begin{cases} 1 & \text{statement is true} \\ 0 & \text{statement is false} \end{cases}$$

helps to describe the glue inside math mode in $(*)$ with three functions.

$$\begin{aligned}\mathbf{N}(\mathtt{T}) := \sum_{\kappa=0}^{\omega}\Bigg( & \sum_{\chi=0}^{7}\big(\mu(\chi, \alpha 1, \mathtt{T}_\kappa)\mathbf{n}\,[\alpha \in \{0,1,5,7\}] \\ & \qquad + \mu(\chi, 10, \mathtt{T}_\kappa)\big) \\ & + \sum_{\chi=4}^{7}\big(\mu(\chi, \alpha 7, \mathtt{T}_\kappa)\mathbf{n}\,[\alpha \in \{0,1,5\}] \\ & \quad + \mu(\chi, 6\alpha, \mathtt{T}_\kappa)\mathbf{n}\,[\alpha \in \{0,1,3,4,5,6,7\}] \\ & \quad + \mu(\chi, 7\alpha, \mathtt{T}_\kappa)\mathbf{n}\,[\alpha \in \{0,4,6,7\}]\big)\Bigg).\end{aligned}$$

The formula for **m** looks only at styles 4–7:

$$\begin{aligned}\mathbf{M}(\mathtt{T}) := \sum_{\kappa=0}^{\omega}\sum_{\chi=4}^{7} & \big(\mu(\chi, 2\alpha, \mathtt{T}_\kappa)\mathbf{m}\,[\alpha \in \{0,1,4,7\}] \\ & + \mu(\chi, \alpha 2, \mathtt{T}_\kappa)\mathbf{m}\,[\alpha \in \{0,5,7\}]\big).\end{aligned}$$

And for **t**, $\mathbf{T}(\mathtt{T})$ is computed in a similar way:

$$\begin{aligned}\mathbf{T}(\mathtt{T}) := \sum_{\kappa=0}^{\omega}\sum_{\chi=4}^{7} & \big(\mu(\chi, 3\alpha, \mathtt{T}_\kappa)\mathbf{t}\,[\alpha \in \{0,1,5,7\}] \\ & + \mu(\chi, \alpha 3, \mathtt{T}_\kappa)\mathbf{t}\,[\alpha \in \{0,1,4,7\}]\big).\end{aligned}$$

At least with the defaults of `plain` TEX, the split into three functions makes some sense as only $\mathbf{M}(\mathtt{T})$ can stretch and shrink; $\mathbf{T}(\mathtt{T})$ cannot shrink and $\mathbf{N}(\mathtt{T})$ can neither stretch nor shrink. For the applications of this paper the shrinkability and stretchability is of interest, so the function $\mathbf{N}(\mathtt{T})$ is never used as long as `\thinmuskip` is not changed.

Again this looks more complicated than it is in order to cover all theoretical aspects: $\mathbf{M}(\mathtt{T})$ counts the number of Bin atoms in text and display style and this number is multiplied by $2\mathbf{m}$. Similarly, $\mathbf{T}(\mathtt{T})$ counts the number of Rel atoms that don't follow a Punct atom; this count is multiplied by $2\mathbf{t}$. The identification of the relevant atoms is easily described for the styles $T'$ and $T$ that appear in paragraphs if the style isn't explicitly changed with `\displaystyle`: A Bin or Rel atom is typeset in text style if it is neither raised nor lowered with respect to the baseline.

To capture inserted glue at a line break the glue function is extended to math mode. In styles 4–7, TEX inserts a penalty after Bin and Rel atoms to create an opportunity to break formulas.

$$\mathbf{G}_\epsilon(\sigma, \mathtt{m}) := \begin{cases} \mathbf{m}, & \mathtt{m} \text{ is a Bin atom in styles 4–7} \\ \mathbf{t}, & \mathtt{m} \text{ is a Rel atom in styles 4–7.} \end{cases}$$

**Summary: Glue in the input.** In the model $(*)$ the infinite glue is captured by $\mathbf{\Omega}(\mathtt{T})$.

The finite glue of explicitly entered glue $\mathtt{W}$ that follows the input $\mathtt{T}$ is $\mathbf{G}_\epsilon(\Phi(\mathtt{T}), \mathtt{W})$. Three functions are defined to represent the finite glue that TEX inserts into input $\mathtt{T}$ which contains material in math mode: $\mathbf{T}(\mathtt{T})$, $\mathbf{M}(\mathtt{T})$, and $\mathbf{N}(\mathtt{T})$. Therefore:

$$\begin{aligned}\mathbf{S}_\epsilon(\mathtt{T}) &:= \text{finite glue in } \mathtt{T} \\ &= \sum_{\kappa=0}^{\omega-1}\mathbf{G}_\epsilon(\Phi(\mathtt{T}_\kappa), \mathtt{W}_\kappa) + \mathbf{T}(\mathtt{T}) + \mathbf{M}(\mathtt{T}) + \mathbf{N}(\mathtt{T})\end{aligned}$$

when $\mathtt{T} = \mathtt{T}_0\mathtt{W}_0\mathtt{T}_1\mathtt{W}_1 \ldots \mathtt{T}_{\omega-1}\mathtt{W}_{\omega-1}\mathtt{T}_\omega$ as defined in $(*)$. Of course, $\mathbf{S}_\epsilon(\mathtt{T}) = (S^\circ_\epsilon(\mathtt{T}), S^+_\epsilon(\mathtt{T}), S^-_\epsilon(\mathtt{T}))$.

## 3 An experiment

I use a well-known text [3, p. 24] for the first experiment. The text is typeset with a smaller `\hsize` than the column width in order to produce overfull boxes. Then the value of the dimension parameter `\emergencystretch` is increased up to the point where an overfull box disappears.

For example, when `\hsize` $= 100\,$pt, five overfull boxes are produced. Then the value of the dimen

The optimal value for `\emergencystretch`

`\emergencystretch` is increased in steps of 0.1 pt, and at `\emergencystretch` = 1.5 pt TEX is able to typeset the text with only two overfull lines.

An experiment starts with a boldface line showing the number of the experiment for reference and it ends with this end-of-experiment marker: □.

**Experiment 1: TEX definitions**

```
\hsize=100pt \overfullrule=1pt
```

**TEX input**

```
Once upon a time, in a distant galaxy called
\"O\"o\c c, there lived a computer named
R.~J. Drofnats.

Mr.~Drofnats---or ``R. J.,'' as he preferred to
be called---was happiest when he was at work
typesetting beautiful documents.
```

**TEX output**

i) `\emergencystretch`: left col. 0.1 pt, right 0.2 pt

| | |
|---|---|
| Once upon a time,<br>in a distant galaxy called<br>Ööç, there lived a com-<br>puter named R. J. Drof-<br>nats.<br>Mr. Drofnats—or “R.<br>J.,” as he preferred to<br>be called—was happiest<br>when he was at work<br>typesetting beautiful doc-<br>uments. | Once upon a time,<br>in a distant galaxy called<br>Ööç, there lived a com-<br>puter named R. J. Drof-<br>nats.<br>Mr. Drofnats—or “R.<br>J.,” as he preferred to<br>be called—was happi-<br>est when he was at work<br>typesetting beautiful doc-<br>uments. |

ii) `\emergencystretch`: left 1.4 pt, right 1.5 pt

| | |
|---|---|
| Once upon a time,<br>in a distant galaxy called<br>Ööç, there lived a com-<br>puter named R. J. Drof-<br>nats.<br>Mr. Drofnats—or “R.<br>J.,” as he preferred to<br>be called—was happi-<br>est when he was at work<br>typesetting beautiful doc-<br>uments. | Once upon a time,<br>in a distant galaxy called<br>Ööç, there lived a com-<br>puter named R. J. Drof-<br>nats.<br>Mr. Drofnats—or<br>“R. J.,” as he preferred<br>to be called—was hap-<br>piest when he was at<br>work typesetting beau-<br>tiful documents. |

iii) `\emergencystretch`: left 9.2 pt, right 9.3 pt

| | |
|---|---|
| Once upon a time,<br>in a distant galaxy called<br>Ööç, there lived a com-<br>puter named R. J. Drof-<br>nats.<br>Mr. Drofnats—or<br>“R. J.,” as he preferred<br>to be called—was hap-<br>piest when he was at<br>work typesetting beau-<br>tiful documents. | Once upon a time,<br>in a distant galaxy<br>called Ööç, there lived<br>a computer named R. J.<br>Drofnats.<br>Mr. Drofnats—or<br>“R. J.,” as he preferred<br>to be called—was hap-<br>piest when he was at<br>work typesetting beau-<br>tiful documents. |

iv) `\emergencystretch`: left 11.1 pt, right 11.2 pt

| | |
|---|---|
| Once upon a time,<br>in a distant galaxy<br>called Ööç, there lived<br>a computer named R. J.<br>Drofnats.<br>Mr. Drofnats—or<br>“R. J.,” as he preferred<br>to be called—was hap-<br>piest when he was at<br>work typesetting beau-<br>tiful documents. | Once upon a time,<br>in a distant galaxy<br>called Ööç, there lived<br>a computer named<br>R. J. Drofnats.<br>Mr. Drofnats—or<br>“R. J.,” as he preferred<br>to be called—was hap-<br>piest when he was at<br>work typesetting beau-<br>tiful documents. |

□

During the experiment seven overfull lines occur. Five exist at the beginning, and two are created during the experiment. We'll name the five cases a) to e); the overfull line that is created in the first paragraph is referenced by b′), the other by d′).

The result of the experiment is captured in the following summary, stating in the first line the number of the experiment and its parameters and in the second line the measured values for the dimension `\emergencystretch`. Values separated by a slash mean: The resolution of one overfull line created another, which was resolved with the second value.

**Experiment 1 continued: Results**

a) parameters; b) used `\emergencystretch`

1a): `\hsize`=100 pt

b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt □

**A few dimensions.** TEX reports by how much the overfull lines are too wide.

$$W(\texttt{called}) = 25.00005\,\text{pt} \quad \text{juts} \quad 5.88911\,\text{pt} \qquad (1a)$$
$$W(\texttt{Drof-}) = 22.94449\,\text{pt} \quad \text{juts} \quad 2.13905\,\text{pt} \qquad (1b)$$
$$W(\texttt{``R.}) = 15.13892\,\text{pt} \quad \text{juts} \quad 12.36128\,\text{pt} \qquad (1c)$$
$$W(\texttt{happiest}) = 36.7223\,\text{pt} \quad \text{juts} \quad 2.33350\,\text{pt} \qquad (1d)$$
$$W(\texttt{doc-}) = 18.61115\,\text{pt} \quad \text{juts} \quad 10.61136\,\text{pt} \qquad (1e)$$

where $W(\texttt{text})$ stands for the width of `text` as defined in the previous section. For a single word or word fragment this is always the natural width, i.e., there is no stretching or shrinking. When we explicitly mention that the natural width of a text with white space is computed then we use the subscript “nw” with $W$.

As mentioned above, later two more lines become overfull:

$$W_{\text{nw}}(\texttt{R. J.}) = 21.38892\,\text{pt} \quad \text{juts} \quad 1.97240\,\text{pt} \qquad (1b')$$
$$W(\texttt{work}) = 21.13893\,\text{pt} \quad \text{juts} \quad 2.08344\,\text{pt} \qquad (1d')$$

The overfull lines are resolved by various means: in a) and c) only stretchability is needed, in d) a word gets hyphenated, d′) is similar to a) and c) but some fixed width material is added at the left side of the line, variable sized material is added to b), in b′) variable sized material is moved to the next line, and e) is similar to that but a word is hyphenated instead of breaking at glue.

In order to have all dimensions at hand for later calculations, here are the values of the material moved to the next line in cases with hyphenation and the material added to some lines at the left:

$$
\begin{aligned}
W_{\mathrm{nw}}(\texttt{a com}) &= 26.11116\,\mathrm{pt} && \text{is added} && (2b)\\
W(\texttt{est}) &= 12.27779\,\mathrm{pt} && \text{is moved} && (2d)\\
W(\texttt{pi}) &= \phantom{0}8.33336\,\mathrm{pt} && \text{is added} && (2d')\\
W_{\mathrm{nw}}(\texttt{tiful doc-}) &= 40.00009\,\mathrm{pt} && \text{is moved} && (2e)
\end{aligned}
$$

**The badness values.** With the help of the parameter `\tracingparagraphs`, the badness values of the line-breaking algorithm can be found [14]. The lines of the paragraph without using `\emergencystretch` and the paragraphs at the right of runs i) to iv) have the following badness values:

$$
\begin{array}{lcl r}
\overline{7},\ *,\ \overline{15},\ *,\ 0 & \text{and} & *,\ 17,\ *,\ 171,\ *,\ 0 & (3)\\
\overline{7},\ *,\ \overline{15},\ *,\ 0 & \text{and} & *,\ 15,\ 190,\ *,\ *,\ 0 & (3\mathrm{i})\\
\overline{7},\ *,\ \overline{15},\ *,\ 0 & \text{and} & 184,\ \overline{17},\ 0,\ 113,\ \overline{2},\ 0 & (3\mathrm{ii})\\
\overline{7},\ 198,\ 0,\ *,\ 0 & \text{and} & 4,\ \overline{17},\ 0,\ 15,\ \overline{2},\ 0 & (3\mathrm{iii})\\
\overline{7},\ 136,\ 0,\ 200,\ 0 & \text{and} & 3,\ \overline{17},\ 0,\ 11,\ \overline{2},\ 0 & (3\mathrm{iv})
\end{array}
$$

The symbol $*$ stands for the infinite badness of the overfull lines. Such a line has the fitness class *tight* [3, p. 97]. Finite badness values of lines containing glue that shrinks are written with a bar above them. In this way all fitness classes can be identified.

In this list the badness values for lines in the fitness classes very loose and loose change to lower values when the `\emergencystretch` increases. The values of some decent lines change too and then it is known that the white space in such a line has to stretch. Only tight lines and decent lines, in which the glue shrinks, keep a constant badness value. As indicated by the bar the 7 and the 2 in (3iv) belong to decent lines in which the glue shrinks.

But the "real" badness values obtained with `\hbadness` [14, p. 367] in the last run are

$$\overline{7}, 4660, 7, 10000, 0 \quad\text{and}\quad 1264, \overline{17}, 0, 206, \overline{2}, 0. \qquad (3\mathrm{iv}')$$

## 4 Some theory

Do we need to perform this stepwise increment in order to obtain the values for `\emergencystretch`, or is there a way to compute the values?

First, let's think about infinite glue. TeX throws an error if it finds infinite shrinkability in a paragraph. Infinite stretchability in a line makes it nearly impossible to generate an overfull line as TeX can break at any glue, penalty or hyphenation possibility. Therefore, in this analysis all glue in the text is assumed to be finite.

As was noted above TeX can get rid of an overfull line in many different ways. It is too complex to handle all the cases at once. It is better to start with a simple model. Therefore, let's make the following assumption: The overfull line is changed only by moving material to the next line. In other words, no new material is added to the previously overfull line, so that only the cases a), b′), c), and d) of experiment 1 are considered in this section.

**A formula for `\emergencystretch`.** The badness is a heuristic based on the amount by which the available spaces have to shrink or to stretch in order to make the line fill a predefined length. Sometimes the formula for the badness is stated as

$$100 \times \left( \frac{\text{used } \genfrac{}{}{0pt}{}{\text{stretch}}{\text{shrink}} \text{ability in the line}}{\text{available } \genfrac{}{}{0pt}{}{\text{stretch}}{\text{shrink}} \text{ability in the line}} \right)^3 \qquad (4)$$

but this is only an *approximation* [3, p. 97]. (Of course, the same word in the numerator and denominator must be selected.) It's not only that the badness is always an integer $\le 10000$, but the computation given in §108 of [4] computes it without the "need to squeeze out the last drop of accuracy." The section further states that the routine is "capable of computing at most 1095 distinct values." For the purpose of the heuristic this computation is sufficient. Nevertheless it should be remembered that the badness value is not computed by a continuous function as the formula might suggest.

As the case of an overfull line is analyzed, formula (4) is used for the stretchability of a line. Let $\beta$ be the badness, $u$ the dimension of used stretchability, $a$ the available stretchability in the line, and $e$ the value of `\emergencystretch`. Then for $e$ large enough to remove the overfull line

$$\beta \approx 100 \left( \frac{u}{a+e} \right)^3 .$$

Of course, $a + e \neq 0\,\mathrm{pt}$. But to see an impact of `\emergencystretch` a line must have some stretchability, i.e., $a > 0\,\mathrm{pt}$.

The badness of a line is difficult to estimate. As mentioned, it can be seen in the data written to the log file if `\tracingparagraphs` is set to 1.

The badness $\beta$ of a non-overfull line is a fraction of the `\tolerance` $\tau > 0$; so for $\varphi$ with $0 \le \varphi \le 1$ the equation

$$\beta = \varphi\tau$$

holds. Replacing the left hand side of the approximation with the right hand side of the equation gives

$$\varphi\tau \approx 100 \left( \frac{u}{a+e} \right)^3 \quad\text{or}\quad \sqrt[3]{\frac{\varphi\tau}{100}} \approx \frac{u}{a+e}$$

The optimal value for `\emergencystretch`

and if $\varphi \neq 0$

$$e \approx \sqrt[3]{\frac{100}{\varphi\tau}}\, u - a. \tag{5}$$

Therefore three values $a$, $u$, and $\varphi$ must be found to get an approximation of $e$ to resolve a given overfull line. In such a line $\varphi > 0$, of course.

As stated here, the values are not obvious to calculate or even to guess but they can be factored to a level which allows estimation of their values. The values $a$ and $u$ can be computed from trace output shown by TeX but not when an overfull line is reported — this is discussed later.

**The plan for the analysis.** It's probably best to look at the situation when the overfull line gets the right amount for `\emergencystretch` so that a break creating an acceptable line occurs. Therefore the first question is: Where can TeX break a line? On p. 96 of [3], five cases are listed. A line break might be at

1. glue if a non-discardable item (not glue, kern, penalty or a math switch) appears before the glue;
2. a kern if it is followed by glue;
3. a math-off if it is followed by glue;
4. a penalty;
5. a hyphen, either inserted by TeX or an explicit hyphen present in the text.

The first three items are combined in this analysis into one case (named "break at glue") using the definitions made in section 2. The glue might be entered by the author or it is inserted by TeX after a binary operation or a relation in math mode. Number 4 can also be added except when the penalty is not followed by glue. So the second case is a break after a penalty that is not followed by glue. The third case represents number 5, the discretionary break.

**Case 1: Break at glue.** A before and after comparison of the line contents shows the situation when the overfull line gets the right amount for the dimension `\emergencystretch` so that a break at the last breakable white space can occur.

In the following analysis, we assume the input `L` doesn't start and the input `M` doesn't end with glue that can be dropped at the beginning or end of a line. Such glue is ignored by TeX and it makes the description much easier if such glue isn't present in the first place.

$$\begin{array}{|c|c|} \hline l_{\mathrm{mt}}\, W_{\mathrm{mt}}(\mathtt{L})\, g_{\mathrm{mt}}\, W_{\mathrm{mt}}(\mathtt{M})\, r_{\mathrm{mt}} & \\ \hline h & o \\ \end{array} \rightarrow \begin{array}{|c|} \hline l\, W(\mathtt{L})\, r \\ \hline h \\ \end{array}$$

**Figure 3:** Resolve overfull line with break at glue

At the left side of Fig. 3 some input `L` of width $W_{\mathrm{mt}}(\mathtt{L})$ is typeset, with `\leftskip` at the left, then it is followed by glue $\mathbf{g}$ of width $g_{\mathrm{mt}}$ at a place with a certain `\spacefactor` $\Phi(\mathtt{L})$ and some material `M` that contains no breakable glue and that is followed by `\rightskip`. The width of the material is named $W_{\mathrm{mt}}(\mathtt{M})$. The white space in the texts, the glue and the skips on both sides are shrunk to their maximum as the line is overfull and therefore it is *maximally tight*; this is indicated by the subscript "mt" to $W$ and other variables. The length of the line at the left is the sum of the `\hsize` $h$ (it may be the width of a `\parshape` or a `\hangindent`) and the amount that the line is too wide, let's call it $o$.

The right hand side contains also some information. The natural width of the three items in the line, $l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ$, must be smaller than $h$ because of the following observation: TeX would not create an overfull line if the stretchability in the line would allow a break at the glue $\mathbf{g}$; at the left the amount of stretchability was too small to resolve the problem. The width $l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ$ can only stretch to the width $h$ with the help of $e$, so it must be smaller than the width $l + W(\mathtt{L}) + r$.

**The available stretchability** of the resolved overfull line (the right side of Fig. 3) lies in the stretchability of the two glue items $\mathbf{l}$ and $\mathbf{r}$ and in the stretchability of the text `L`. All these elements were named in section 2: The available stretchability on the right hand side of Fig. 3 is the sum of the stretchability of the `\leftskip`, $l^+$, of the text `L`, $S^+_\epsilon(\mathtt{L})$, and of the `\rightskip`, $r^+$:

$$a = l^+ + S^+_\epsilon(\mathtt{L}) + r^+. \tag{6}$$

**The used stretchability** is the next difficult-to-guess value. As explained above in the right hand side of Fig. 3 the text is typeset with some amount of stretch. This is the used stretchability $u$. So the following equations hold:

$$h = l + W(\mathtt{L}) + r = l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ + u.$$

This leads to an equation for $u$: $u = h - l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ$. But it is too soon to stop here. A term like $W_{\mathrm{nw}}(\mathtt{L})$ is so content-dependent that it should be analyzed further. In (6) the value $S^+_\epsilon(\mathtt{L})$ is used. This is more or less the number of spaces in `L`. It is an important characteristic and not dependent on all elements of the input `L`. $W_{\mathrm{nw}}(\mathtt{L})$ is different, as hardly any other text will have the same width as `L`.

Ok, the analysis continues and the equation for the right hand side is reordered

$$l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ = h - u.$$

The left hand side gives another equation

$$h + o = l_{\mathrm{mt}} + W_{\mathrm{mt}}(\mathtt{L}) + g_{\mathrm{mt}} + W_{\mathrm{mt}}(\mathtt{M}) + r_{\mathrm{mt}}\,.$$

where $W_{\mathrm{mt}}(\mathtt{L})$ typesets the text `L` with the minimal amount of white space as explained above. Of course, $l_{\mathrm{mt}} = l^\circ - l^-$ and $r_{\mathrm{mt}} = r^\circ - r^-$. For the second term the equation $W_{\mathrm{mt}}(\mathtt{L}) = W_{\mathrm{nw}}(\mathtt{L}) - S^-_\epsilon(\mathtt{L})$ holds. And $g_{\mathrm{mt}} = G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g})$ with the function $\mathbf{G}_\epsilon$ of section 2.

When all these equations are applied to the equation of the left hand side it becomes:

$$\begin{aligned} h + o &= l^\circ - l^- + W_{\mathrm{nw}}(\mathtt{L}) - S^-_\epsilon(\mathtt{L}) + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) \\ &\qquad - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) + W_{\mathrm{mt}}(\mathtt{M}) + r^\circ - r^- \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ &\qquad + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) + W_{\mathrm{mt}}(\mathtt{M}). \end{aligned}$$

Now the first three summands are replaced by $h - u$ from the equation of the right hand side:

$$\begin{aligned} h + o = h - u - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) + W_{\mathrm{mt}}(\mathtt{M}). \end{aligned}$$

The goal is reached if $u$ is moved to the left side and $o$ to the right:

$$\begin{aligned} u = W_{\mathrm{mt}}(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g}). \end{aligned}$$

All values except $W_{\mathrm{mt}}(\mathtt{M})$ are known and the width of the material `M` that is moved to the next line is much easier to estimate than the value of $u$, especially if it is of fixed width. Otherwise the material that is moved contains one or more ties. As $W_{\mathrm{mt}}(\mathtt{M})$ is the width of the material with all white spaces shrunk to maximum the following equation with the natural width $W_{\mathrm{mt}}(\mathtt{M}) = W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M})$ can be applied to the result. In total this gives

$$\begin{aligned} u = W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g}). \end{aligned} \tag{7}$$

**Case 2: Break at penalty.** As mentioned above the only situation that must be considered here is a break at a penalty that is not followed by glue. Let's call the penalty $\pi$; it separates inputs `L` and `M` such that no kerning or ligatures must be taken into account.

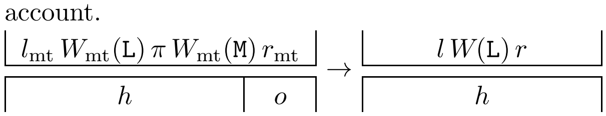


**Figure 4:** Resolve overfull line with break at penalty

The resulting line at the right in Fig. 4 looks identical to the right hand side in Fig. 3. So (6) holds without a change. The first difference occurs when the formula for the left hand side is written down; now the summand $g_{\mathrm{mt}}$ disappears. Or one can say the glue $\mathbf{g}$ of Fig. 3 must be replaced by $\mathbf{z}$ and all formulas stay the same. Therefore the equation for $u$ becomes

$$u = W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^-.$$

Using the bracket notation of section 2, an abbreviation for the case when the line break occurs at glue is defined

$$\Gamma := [\text{break occurs at glue}]$$

so that the cases 1 and 2 can be combined:

$$\begin{aligned} u = W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ + \big(G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\big)\Gamma. \end{aligned} \tag{8}$$

**Case 3: Break at hyphen.** Another form of removing an overfull line is given in Fig. 5. Here the material that sticks out in the overfull line is not moved completely to the next line. It is a discretionary break and only the fragment of width $W(\mathtt{M})$ is moved; material with width $W(\mathtt{K}')$ is kept on the line together with the hyphen character of width $W(\mathtt{-})$. (The format `plain` TeX assigns the value of `\defaulthyphenchar` to `\hyphenchar`. Here $W(\mathtt{-})$ is used instead of $W($`\char\hyphenchar\font`$)$.) As the characters of a font can interact through ligatures and kerns, the sum of the two widths is not necessarily the width of the concatenated strings. The notation $L(\mathtt{S}\&\mathtt{T})$ is used to denote the change of width (compared to the sum) either through a ligature or kern when the strings `S` and `T` are concatenated, i.e., $W_{\mathrm{nw}}(\mathtt{ST}) = W_{\mathrm{nw}}(\mathtt{S}) + W_{\mathrm{nw}}(\mathtt{T}) + L(\mathtt{S}\&\mathtt{T})$ as explained in section 2.

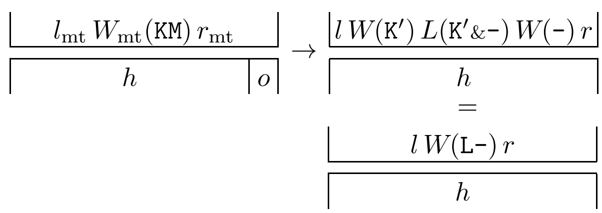


**Figure 5:** Resolve overfull line with break at hyphen

If the discretionary break occurs at an explicit hyphen then $\mathtt{K} = \mathtt{K}'\mathtt{-}$; otherwise, $\mathtt{K} = \mathtt{K}'$. The distinction can be handled via the abbreviations

$$\Theta := [\text{the line ends with an inserted hyphen}]$$

$$\Xi := [\text{the line ends with an explicit hyphen}]$$

to avoid long statements in bracket notation in the formulas.

Using the first abbreviation the widths of `K` and $\mathtt{K}'$ fulfill the following equation:

$$\begin{aligned} W_{\mathrm{nw}}(\mathtt{K}) + \big(L(\mathtt{K}\&\mathtt{-}) + W(\mathtt{-})\big)\Theta = \\ W_{\mathrm{nw}}(\mathtt{K}') + L(\mathtt{K}'\&\mathtt{-}) + W(\mathtt{-}). \end{aligned} \tag{9}$$

Note that $\mathtt{K}'$ might end in a hyphen, for example, if the break occurs at an em-dash, and therefore the summand $L(\mathtt{K}'\&\mathtt{-})$ is important.

**The available stretchability** comes from the `\leftskip`, the `\rightskip`, and the text $\mathtt{K}'$. But as Fig. 5 shows, the concatenation of this string together with the fixed width hyphen is `L-` and therefore all the stretchability comes from `L`, the text that remains in the line: Equation (6) is still valid.

**The used stretchability** requires a bit more complicated equivalence of the left and right hand side equations that were discussed in case 1. Figure 5 together with an application of (9) gives

$$\begin{aligned} h &= l^\circ + W_{\mathrm{nw}}(\texttt{L-}) + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\texttt{K}') + L(\texttt{K}'\&\texttt{-}) + W(\texttt{-}) + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\texttt{K}) + \big(L(\texttt{K\&-}) + W(\texttt{-})\big)\Theta + r^\circ + u \end{aligned}$$

or

$$l^\circ + W_{\mathrm{nw}}(\texttt{K}) + r^\circ = h - \big(L(\texttt{K\&-}) + W(\texttt{-})\big)\Theta - u.$$

The left hand side gives

$$\begin{aligned} h + o &= l_{\mathrm{mt}} + W_{\mathrm{mt}}(\texttt{KM}) + r_{\mathrm{mt}} \\ &= l_{\mathrm{mt}} + W_{\mathrm{mt}}(\texttt{K}) + L(\texttt{K\&M}) + W_{\mathrm{mt}}(\texttt{M}) + r_{\mathrm{mt}} \\ &= l^\circ + W_{\mathrm{nw}}(\texttt{K}) + r^\circ - l^- - S^-_\epsilon(\texttt{K}) - r^- \\ &\qquad + L(\texttt{K\&M}) + W_{\mathrm{mt}}(\texttt{M})\,. \end{aligned}$$

With the equation for the right hand side and with $S^-_\epsilon(\texttt{K}) = S^-_\epsilon(\texttt{L})$ this equals

$$\begin{aligned} h + o &= h - \big(L(\texttt{K\&-}) + W(\texttt{-})\big)\Theta - u \\ &\qquad - l^- - S^-_\epsilon(\texttt{L}) - r^- + L(\texttt{K\&M}) + W_{\mathrm{mt}}(\texttt{M}) \end{aligned}$$

and therefore with $W_{\mathrm{mt}}(\texttt{M}) = W_{\mathrm{nw}}(\texttt{M}) - S^-_\epsilon(\texttt{M})$

$$\begin{aligned} u &= W_{\mathrm{nw}}(\texttt{M}) - S^-_\epsilon(\texttt{M}) - o - l^- - S^-_\epsilon(\texttt{L}) - r^- \\ &\qquad - \big(L(\texttt{K\&-}) + W(\texttt{-})\big)\Theta + L(\texttt{K\&M})\,. \end{aligned}$$

The last step is to remove the reference to `K`. When the hyphen is inserted $L(\texttt{K\&-}) = L(\texttt{L\&-})$ and $L(\texttt{K\&M}) = L(\texttt{L\&M})$. If the hyphen is explicit, then $L(\texttt{K\&M}) = L(\texttt{L-\&M})$. Using $\Xi$ the equation for $u$ can be stated without `K` as

$$\begin{aligned} u &= W_{\mathrm{nw}}(\texttt{M}) - S^-_\epsilon(\texttt{M}) - o - l^- - S^-_\epsilon(\texttt{L}) - r^- \\ &\qquad + \big(L(\texttt{L\&M}) - L(\texttt{L\&-}) - W(\texttt{-})\big)\Theta \\ &\qquad + L(\texttt{L-\&M})\,\Xi. \end{aligned} \tag{10}$$

**Combining the cases.** The equations (8) and (10) can be combined:

$$\begin{aligned} u &= W_{\mathrm{nw}}(\texttt{M}) - S^-_\epsilon(\texttt{M}) - o - l^- - S^-_\epsilon(\texttt{L}) - r^- \\ &\qquad + \big(G^\circ_\epsilon(\Phi(\texttt{L}),\texttt{g}) - G^-_\epsilon(\Phi(\texttt{L}),\texttt{g})\big)\,\Gamma \\ &\qquad + \big(L(\texttt{L\&M}) - L(\texttt{L\&-}) - W(\texttt{-})\big)\Theta \\ &\qquad + L(\texttt{L-\&M})\,\Xi \end{aligned} \tag{11}$$

and (6) is used in all three cases for $a$.

**The factor $\varphi$.** Two of the three variables have been transformed into “simpler” forms (6) and (11); at least they are simpler to estimate.

The value $\varphi$ can be set to 1 with the following reasoning: The overfull line is removed as soon as possible, so the maximum allowed stretchability will be used that results in the badness $\tau$. See the data in (3iii), (3iv), (3ii), and (3i) which show the badness values after the overfull line disappears for the cases a), b′), c), and d), resp. That $\varphi$ is sometimes less than 1 in experiment 1 can be explained by the fact that the increment for the stretchability was done in steps by 0.1 pt instead of 1 sp.

For example, if, let's say, 0.00001 pt is needed as additional stretchability when $e_\iota$ has already been applied, the situation can be described by the following approximation using $\tau = 200$:

$$200 \approx 100\left(\frac{u + 0.00001\,\mathrm{pt}}{a + e_\iota + 0.00001\,\mathrm{pt}}\right)^3,$$

but

$$\alpha \approx 100\left(\frac{u + 0.00001\,\mathrm{pt}}{a + e_\iota + 0.1\,\mathrm{pt}}\right)^3$$

is computed with the strategy of experiment 1 where $e_{\iota+1} = e_\iota + 0.1\,\mathrm{pt}$. If we divide the left side of the second equation by the left side of the first equation, and the right side of the second equation by the right side of the first equation, the quotients are

$$\frac{\alpha}{200} \approx \left(\frac{a + e_\iota + 0.00001\,\mathrm{pt}}{a + e_\iota + 0.1\,\mathrm{pt}}\right)^3.$$

With $a + e_\iota = 5\,\mathrm{pt}$ the quotient on the right hand side is $(5.00001\,\mathrm{pt}/5.1\,\mathrm{pt})^3 \approx 0.98^3 = 0.941192$, so that $\alpha \approx 188$. And with $a + e_\iota = 10\,\mathrm{pt}$ $\alpha$ gets no larger than 194.

Again, this argument is only valid if $a > 0\,\mathrm{pt}$, as otherwise TeX cannot make use of the additional stretchability, i.e., if the line has no place to stretch additional stretchability cannot change the output.

**The result.** With $\varphi = 1$ and equations (6) and (11) the optimal value of the dimen `\emergencystretch` $e$ is given for the abovementioned cases by

$$\begin{aligned} e \approx \sqrt[3]{\frac{100}{\tau}}\Big(&W_{\mathrm{nw}}(\texttt{M}) - S^-_\epsilon(\texttt{M}) - o - l^- - S^-_\epsilon(\texttt{L}) - r^- \\ &+ \big(G^\circ_\epsilon(\Phi(\texttt{L}),\texttt{g}) - G^-_\epsilon(\Phi(\texttt{L}),\texttt{g})\big)\,\Gamma \\ &+ \big(L(\texttt{L\&M}) - L(\texttt{L\&-}) - W(\texttt{-})\big)\Theta \\ &+ L(\texttt{L-\&M})\,\Xi\Big) \\ &- l^+ - S^+_\epsilon(\texttt{L}) - r^+ \end{aligned} \tag{12}$$

if $l^+ + S^+_\epsilon(\texttt{L}) + r^+ > 0\,\mathrm{pt}$.

## 5 Numerical calculations

Approximation (12) looks rather complicated with its many parameters but some of them are zero when others are nonzero. Several parameters are determined by the font used and some are specified via the line-breaking parameters. Here are the relevant values for the situation of experiment 1.

1. First, the environmental condition $\epsilon$ is 0, meaning that the font-related parameters have to be used. These have the following values in `cmr10`:

1a. `\fontdimen2` $= f_2 = 3.33333\,\mathrm{pt}$
1b. `\fontdimen3` $= f_3 = 1.66666\,\mathrm{pt}$
1c. `\fontdimen4` $= f_4 = 1.11111\,\mathrm{pt}$
1d. `\fontdimen7` $= f_7 = 1.11111\,\mathrm{pt}$
1e. and it means that $\mathbf{s} = \mathbf{z}$ as well as that $\mathbf{x} = \mathbf{z}$.

The `\tolerance` equals 200 in `plain` TeX when the emergency pass is performed so

2. $\sqrt[3]{100/\tau} = \sqrt[3]{0.5} \approx 0.794$.

The `plain` format leaves both `\leftskip` and `\rightskip` zero.

3. $\mathbf{l} = \mathbf{z}$, i.e., $l^+ = 0\,\mathrm{pt}$ and $l^- = 0\,\mathrm{pt}$
4. $\mathbf{r} = \mathbf{z}$, i.e., $r^+ = 0\,\mathrm{pt}$ and $r^- = 0\,\mathrm{pt}$

Two values can be determined for all `cm` fonts:

5. $L$(`T&-`) $= 0\,\mathrm{pt}$ for all `T` not ending in a hyphen,
6. $L$(`S-&T`) $= 0\,\mathrm{pt}$ for any text `S` and all `T` not beginning with a hyphen.

A hyphen has no kerning with any other character and ligatures are built only with other hyphens. But a hyphen is never inserted when explicit hyphens are present to break the line.

The width of a hyphen in `cmr10` is:

7. $W$(`-`) $= 3.33333\,\mathrm{pt}$.

The following parameters are not used in the calculations as no math typesetting occurs in the experiment. Nevertheless the values are given for completeness:

8. `\fontdimen6\textfont2` $= f_6^{\mathrm{t2}} = 10.00002\,\mathrm{pt}$ as the `\textfont2` is `cmsy10`;
9. `\medmuskip` $= (4\,\mathrm{mu}, 2\,\mathrm{mu}, 4\,\mathrm{mu})$ and therefore $\mathbf{m} = (4, 2, 4) \times 10.00002/18\,\mathrm{pt}$ which gives the glue $(2.22223\,\mathrm{pt}, 1.11111\,\mathrm{pt}, 2.22223\,\mathrm{pt})$;
10. similarly, `\thickmuskip` $= (5\,\mathrm{mu}, 5\,\mathrm{mu}, 0\,\mathrm{mu})$ so $\mathbf{t} = (5.55557\,\mathrm{pt}, 5.55557\,\mathrm{pt}, 0\,\mathrm{pt})$.

The next parameters depend upon the content of the line and the type of the break. The value of $o$ is shown by TeX in the message about the overfull line. From this message the possible type of break, i.e., $\Gamma$, $\Theta$, or $\Xi$ can be determined.

| | case a) | case b′) | case c) | case d) |
|---|---|---|---|---|
| 11. $o$/pt = | 5.88911 | 1.97240 | 12.36128 | 2.3335 |
| 12. $\Xi$ = | 0 | 0 | 0 | 0 |
| 13. $\Theta$ = | 0 | 0 | 0 | 1 |
| 14. $\Gamma$ = | 1 | 1 | 1 | 0 |

Then `L` and $\Phi$(`L`) as well as two values of the glue $\mathbf{g} = (g^\circ, g^+, g^-)$ where the break occurs if $\Gamma = 1$ are calculated. The values of the $\mathbf{S}_\epsilon$ function can be counted by the method described in section 2.

| | case a) | case b′) | case c) | case d) |
|---|---|---|---|---|
| 15. (end of) `L` = | galaxy | named | or | happi |
| 16. $\Phi$(`L`) = | 1000 | 1000 | 1000 | n/a |
| 17. $g^\circ$ = | $f_2$ | $f_2$ | $f_2$ | n/a |
| 18. $g^-$ = | $f_4$ | $f_4$ | $f_4$ | n/a |
| 19. $S_\epsilon^+$(`L`) = | $3f_3$ | $2f_3$ | $f_3$ | $2f_3$ |
| 20. $S_\epsilon^-$(`L`) = | $3f_4$ | $2f_4$ | $f_4$ | $2f_4$ |

The text `M` is shown by TeX in the message about the overfull line, but the value $W_{\mathrm{nw}}$(`M`) must be either measured or guessed. For experiment 1 the values have been documented in (1) and (2).

| | case a) | case b′) | case c) | case d) |
|---|---|---|---|---|
| `M` = | called | R. J. | “R. | est |
| 21. $W_{\mathrm{nw}}$(`M`)/pt = | 25.00005 | 21.38892 | 15.13892 | 12.27779 |
| 22. $S_\epsilon^-$(`M`) = | 0 | $f_4$ | 0 | 0 |
| 23. $L$(`L&M`) = | n/a | n/a | n/a | 0 pt |

Although 23 parameters have been listed, yet more are involved which are hidden in other parameters. One is the set of `\sfcode`s for all the characters, which are used in the calculation of $S_\epsilon^-()$ and $S_\epsilon^+()$. Another is the font size (or the width of the characters), which is important for $W_{\mathrm{nw}}$(`M`).

As mentioned above, $a$ and $u$ can be derived from the trace data written by TeX, but only after the overfull line has been resolved. The output from `\tracingoutput` can be used: The sum of the stretchability of a line gives $a$ and the multiplication with the *glue set* value computes $u$ ([3], p. 75 and p. 79, resp.). The values are listed in lines A and B, their product $u$ rounded up to five places is given in C.

| | case a) | case b′) | case c) | case d) |
|---|---|---|---|---|
| A. stretch/pt = | 4.99998 | 3.33332 | 1.66666 | 3.33332 |
| B. glue set = | 3.59998 | 5.49165 | 2.33325 | 1.31664 |
| C. A ∗ B/pt = | 17.99983 | 18.30543 | 3.88873 | 4.38878 |
| D. $a$/pt (6) = | 4.99998 | 3.33332 | 1.66666 | 3.33332 |
| E. $u$/pt (11) = | 17.99963 | 18.30541 | 3.88875 | 4.38874 |
| F. $e$/pt (12) ≈ | 9.286 | 11.196 | 1.4198 | 0.15 |
| G. $e$/pt ex. 1 = | 9.3 | 11.2 | 1.5 | 0.2 |

The calculated values in line F agree with the stepwise measured data in row G. In the rest of this article, the computations round up the value of $e$ to one decimal place; a higher precision is not needed.

**Macros.** When the parameter `\showboxbreadth` is large enough, say 100 for lines with 60–70 characters, the complete overfull line is written by TeX to the log file and there all stretch and shrink values of the glue can be found. But since the glue set ratio is reported as −1.0 in an overfull line, only the value of $a$ can been determined from the message.

The optimal value for `\emergencystretch`

Nevertheless, macros can be written to make the calculation from the message shown. In this subsection a brief description for the design of a set of macros based on (12) is given.

Except for the cube root the calculation uses only simple arithmetic, which is available in TeX. For the cube root I use the formula $\sqrt[3]{\alpha^3+\beta} \approx \alpha + \beta/(3\alpha^2)$. The number of parameters makes the calculations a little bit complicated but TeX can do it, especially as it knows all the font parameters, the `\tolerance`, and the width of strings. A set of macros can be designed that receives the data about the moved and the kept material, the type of the expected line break, and the overhang to perform the calculation to get $e$.

Usually only five values must be specified as parameters or by macro names — three of them are displayed by TeX and two are known by the user. The other parameters can be calculated by TeX, although a user should be able to change them.

The following four macros are used, for example, to compute the value of `\emergencystretch` in case a) of experiment 1:

```
1. \dataEtextM(called)
2. \dataEtextL(in a distant galaxy)
3. \breakEglue(1000)
4. \calcEoverhang(5.88911pt)
```

1. The first macro receives the text `M`. It allows the calculation of $W_{\mathrm{nw}}(\mathtt{M})$ and $S_\epsilon^-(\mathtt{M})$.
2. The second macro gets the text `L` and it determines its stretch and shrink units $S_\epsilon^+(\mathtt{L})$ and $S_\epsilon^-(\mathtt{L})$.
   - Now $a$ is known and for $u$ only the three terms which are multiplied by $\Gamma$, $\Xi$, and $\Theta$ as well as $o$ are missing.
3. One of the following five macros is called to specify the type of break:
   a) `\breakEhyphen`,
   b) `\breakEexhyphen`,
   c) `\breakEmath` with a parameter for the atom (Bin or Rel) after which the break occurs, and a glue specification for user-entered glue,
   d) `\breakEotherglue` — used when the glue is, for example, an `\hskip` — which gets the three dimen values $g^\circ$, $g^+$, and $g^-$; or it is a break with glue **z** at penalty and
   e) `\breakEglue` that has one parameter: $\Phi(\mathtt{L})$.
   - Now $u + o$ is known.
4. The last macro receives the value $o$ as parameter. It calculates $u$, the factor with the cube root and determines $e$ rounded up to one decimal place.
5. All macros use the `plain` TeX defaults for their calculation. As mentioned above, to change the font, `\leftskip`, etc., some macros are written that can be called before the first macro to specify different values.

This makes it easy to find $e$ for the original five cases of overfull lines in experiment 1:

| | |
|---|---|
| Case a): 9.3 pt | Case c): 1.5 pt |
| Case b): 10.7 pt | Case d): 0.2 pt |
| Case e): 5.6 pt | |

and for the two created overfull lines:

| | |
|---|---|
| Case b′): 11.2 pt | Case d′): 6.7 pt |

although the theory of section 4 does not apply to the cases b), e), and d′). So their computed values do not agree with the measured ones.

## 6 Line-breaking parameters

In the approximation (12) for $e$ many different values are used, and it seems useful to discuss some of them. As `\tolerance` plays a very prominent rôle, and is the only parameter with a non-linear relationship to `\emergencystretch`, the group of line-breaking parameters is analyzed first.

**The `\tolerance`.** The dimen `\emergencystretch` was introduced to avoid large values of `\tolerance`. The latter is more of a document-wide parameter, while the former should be applied to a single paragraph. An increase of `\tolerance` in order to lower the value of `\emergencystretch` is therefore not a good idea: Tight lines cannot become tighter and loose lines benefit from `\emergencystretch`.

When (5) is written as a function of the parameter `\tolerance` for an overfull line

$$f(\xi) = \sqrt[3]{100}u\,\xi^{-1/3} - a, \quad \xi > 0,$$

it is a monotone decreasing function as in a situation with an overfull line $u > a > 0$ and its derivative is $< 0$. For $\xi \to \infty$ the limit is $-a$. As usual, values above 10000 are of no use in the application and the `\emergencystretch` cannot be a negative distance.

With $\xi_0 = 100$, which is the `plain` TeX default value of `\pretolerance`, the function value $f(\xi_0)$ is $u - a > 0$ pt so there must be a $\xi_1$ for which the function becomes zero. This is $\xi_1 = 100(u/a)^3$, which is the approximation for the badness. Or in other words: With the real badness values, as in (3iv′), the required `\emergencystretch` is 0 pt. The value $f(\tau)$ computes the right hand side of (5) so it is the additional stretchability $e$ that is needed to resolve the overfull line.

A concrete numerical example might help to understand this better. For case d) the values $u = 4.38878$ pt and $a = 3.33332$ pt were stated above.

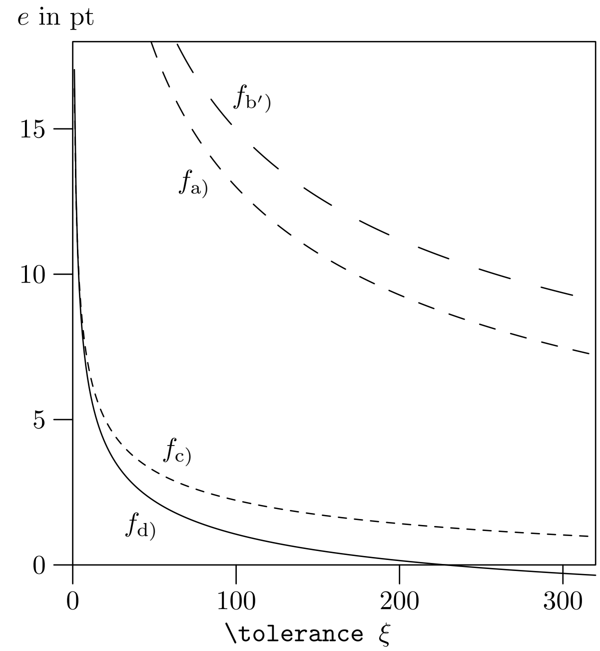


**Figure 6:** Graphs of function $f$ for the cases discussed in section 4

As $\sqrt[3]{100} \approx 4.642$ the function becomes

$$f_{\mathrm{d})}(\xi) = 20.37091\,\xi^{-1/3}\,\mathrm{pt} - 3.33332\,\mathrm{pt}.$$

The other three cases a), b′), and c) have similar formulas. The graphs of all four functions are drawn in Fig. 6. It shows the typical curve of these functions: A small $\xi$ has a high value but the function values drop quickly as $\xi$ is increased. This effect slows down and a larger $\xi$ reduces the required `\emergencystretch` by only a small amount. The value of $\xi = \tau = 200$ gives $e$.

**Hyphenation.** In the context of the removal of overfull lines the assumption was made that the hyphenation of words is a valid option. This might not always be the case. There are several primitive TeX commands that prevent hyphenation completely or for certain words only: The assignment of 10000 to the penalty `\hyphenpenalty` and its companion `\exhyphenpenalty` suppresses hyphenation completely as does the assignment `\hyphenchar` $= -1$. And `\uchyph` $= 0$ suppresses hyphenation of words that start with an uppercase letter. All these settings influence the possibility of hyphenating certain words and therefore the values that are available to remove overfull lines with `\emergencystretch` might increase. If hyphenation is suppressed only $\Gamma$ can be 1, $\Theta$ and $\Xi$ must be 0 in (12).

The commands act like switches, not continuous functions. As the final result of experiment 1 doesn't hyphenate a word which starts with an uppercase letter, the switch `\uchyph` $= 0$ only changes the initial situation ("Drofnats" cannot be hyphenated) but the data for `\emergencystretch` doesn't change, as the following experiment shows. Its results are found by the procedure used in experiment 1.

**Experiment 2: Description**
Suppress hyphenation of words that start with an uppercase letter.

**TeX definitions**

```
\uchyph=0
```

**Result (experiment 1 vs. experiment 2)**

a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt
2a): `\uchyph`=0
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt □

The next two experiments show that the penalties `\hyphenpenalty` and `\exhyphenpenalty` act as switches. The badness of an overfull box is reported as 1000000 [3, p. 229]. An overfull line is so bad in TeX's opinion that no finite setting for the hyphenation parameters should make a difference.

**Experiment 3: Description**
All penalties are set to the maximum finite value and all demerits are set to a high value.

**TeX definitions**

```
\hyphenpenalty=9999 \exhyphenpenalty=9999
\linepenalty=9999 \finalhyphendemerits=1000000
\adjdemerits=1000000 \doublehyphendemerits=1000000
```

**Result (experiment 1 vs. experiment 3)**

a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt
3a): `\...penalty`=9999 and `\...demerits`=1000000
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt □

The situation is different when hyphenation is completely suppressed, since the final result of experiment 1 contains hyphenated words.

**Experiment 4: Description**
Hyphenation of all words is suppressed.

**TeX definitions**

```
\hyphenpenalty=10000 \exhyphenpenalty=10000
```

**Result (experiment 1 vs. experiment 4)**

a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt
4a): `\hyphenpenalty`=`\exhyphenpenalty`=10000
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  15.2 pt  5.6 pt □

The necessary values for $e$ are found with (12) when only the break at glue or penalty is considered although TeX will still show hyphenation points in the messages for the overfull lines. Note: case e) is now covered by (12); the computation at the end of section 5 gives $e$.

The optimal value for `\emergencystretch`

## 7 Character- and font-related parameters

Some of the parameters of (12) are directly related to the properties of characters. Among them are the `\sfcode` and, closely related, the `\spacefactor`. But the largest group of parameters in (12) are related to the font used; for example, there are the `\fontdimen` 2, 3, 4, and 7 and the ligtable entries $L(\mathtt{L\&-})$, $L(\mathtt{L\&M})$, and $L(\mathtt{L-\&M})$. And with $W_{\mathrm{nw}}(\mathtt{M})$ the width of a character string occurs. Some of the listed values can be changed by an author with the help of TeX; others must be handled as constants.

**The `\spacefactor`.** The integer `\spacefactor` is an important characteristic for horizontal mode and the `\sfcode` assigned to each character modifies it. (The assignments are made by INITEX and the format, so the `\sfcode` is not directly a font-related parameter.) In (12) the four parameters $S^-_\epsilon(\mathtt{M})$, $S^-_\epsilon(\mathtt{L})$, $S^+_\epsilon(\mathtt{L})$, and $\Phi(\mathtt{L})$ are affected by all of these. In this subsection the relation of `\spacefactor` to the formula for `\emergencystretch` is analyzed; a TeX command to change the `\sfcode`s is discussed in the next subsection.

**Experiment 5: Description**
Set `\spacefactor` $= 1200$ if it is 1000 before a space.

**TeX definitions**

```
\def\1{\ifnum\spacefactor<1200 \count255=1200
       \else \count255=\spacefactor
       \fi \spacefactor=\count255 }
```

**TeX input**

```
Once\1 upon\1 a\1 time,\1 in\1 a\1 distant\1
galaxy\1 called\1 \"O\"o\c c,\1 there\1 lived\1
a\1 computer\1 named\1 R.\1~J.\1 Drofnats.

Mr.\1~Drofnats---or\1 ‘‘R.\1 J.,’’\1 as\1 he\1
preferred\1 to\1 be\1 called---was\1 happiest\1
when\1 he\1 was\1 at\1 work\1 typesetting\1
beautiful\1 documents.
```

**Result (experiment 1 vs. experiment 5)**
a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt  9.3 pt/11.2 pt  1.5 pt  0.2 pt/1.5 pt  1.5 pt
5a): `\spacefactor`=1200 instead of 1000
b): 8.3 pt  8.3 pt/10.5 pt  1.5 pt  0.0 pt/1.5 pt  1.5 pt ▯

As expected, the increase by 20% of the stretchability per space reduces the required value for the `\emergencystretch`. As $0.2f_3 = 0.2 \times 1.66666\,\mathrm{pt} = 0.333332\,\mathrm{pt}$ the values of cases a) and b) with three spaces can be up to $\approx 1\,\mathrm{pt}$ smaller and this is what the experiment shows. Of course the amount in the resolved line is stretched to its maximum and the additional stretchability is used to its maximum too.

For case b′) with two spaces it is only $\approx 0.7\,\mathrm{pt}$ and case c) stays at 1.5 pt as no space was changed. Case d) vanishes and the cases d′) and e) do not change as the added material is more important than the change in the stretchability of the spaces.

**The `\sfcode`s.** A `plain` TeX control sequence that changes the `\sfcode`s of the punctuation marks `. ? ! : ; ,` is `\frenchspacing`. It makes a space after a punctuation mark equal to a normal space for any text `T`; let's call this space **g**. But in our cases its use will not change anything because there are so few punctuation marks in the text: three commas and seven periods. The space factor of the periods has no influence on the text as two of the seven periods are at the end of the paragraphs, two are followed by a tie which resets the value to 1000, and three follow an uppercase letter so that the value is changed from 999 to 1000. And the three commas are never part of an overfull line.

Therefore, experiment 1 can be treated as if the command `\frenchspacing` was given and so the equalities $S^-_\epsilon(\mathtt{L}) = \nu_{␣}(\mathtt{L})g^-$ and $S^+_\epsilon(\mathtt{L}) = \nu_{␣}(\mathtt{L})g^+$ hold. Approximation (12) can be stated as a function of $\xi = \nu_{␣}(\mathtt{L})$, the number of spaces in `L`:

$$\begin{aligned} g(\xi) = {} & -\left(\sqrt[3]{\frac{100}{\tau}}g^- + g^+\right)\xi \\ & + \sqrt[3]{\frac{100}{\tau}}\Big(W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}) - o - l^- - r^- \\ & \qquad + \big(g^\circ - g^-\big)\Gamma \\ & \qquad + \big(L(\mathtt{L\&M}) - L(\mathtt{L\&-}) - W(\mathtt{-})\big)\Theta \\ & \qquad + L(\mathtt{L-\&M})\,\Xi\Big) - l^+ - r^+. \end{aligned}$$

It is a linear equation with negative slope. For `plain` TeX and `cmr10` $\sqrt[3]{100/\tau}\,g^- + g^+ \approx 2.43\,\mathrm{pt}$. So each space gives a relief of this amount when the command `\frenchspacing` is active.

Let's perform a concrete calculation to find the value of the intercept. For case d) three other terms are nonzero: $W_{\mathrm{nw}}(\mathtt{M}) = 12.27779\,\mathrm{pt}$ by (2d), $o = 2.3335\,\mathrm{pt}$ by (1d), and as $\Theta = 1$ $L(\mathtt{L\&M}) - L(\mathtt{L\&-}) - W(\mathtt{-}) = 0\,\mathrm{pt} - 0\,\mathrm{pt} - 3.33333\,\mathrm{pt} = -3.33333\,\mathrm{pt}$, so $0.794 \times (12.27779 - 2.3335 - 3.33333)\,\mathrm{pt} \approx 5.2491\,\mathrm{pt}$.

Similar equations can be created for the other three cases of section 4, as shown in Fig. 7 (next page). A small dot on the lines shows how many spaces are present in that case. As a parameter to $g$ this gives the value of $e$ because no other input can stretch or shrink. These spaces must additionally stretch; each space widens by `\emergencystretch` divided by the number of spaces in the line.

**The `\fontdimen`s.** The `\fontdimen` parameters are read by TeX from a font's tfm file. Although they

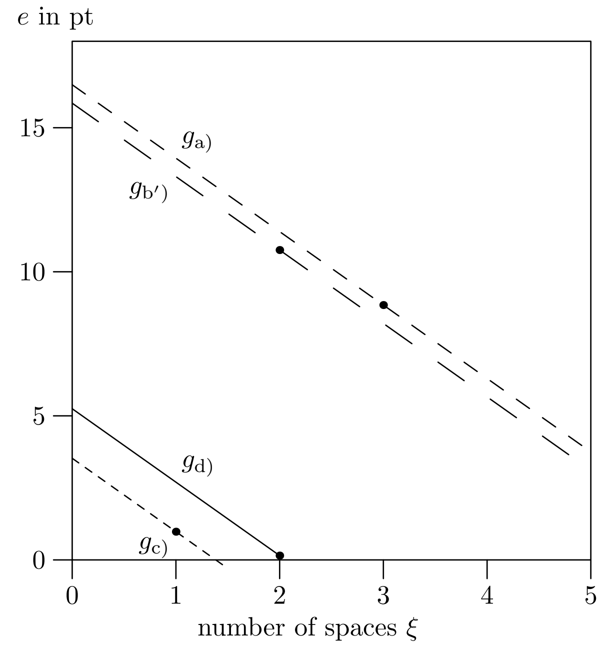


**Figure 7:** Graphs of function $g$ for the cases discussed in section 4

can be changed by an author, this is not recommended except in very unusual situations. I consider them to be constants. Here is a list for several fonts:

| $f_\nu$ | `cmr10` | `cmbx10` | `cmr12` | `cmtt10` |
|---|---|---|---|---|
| 2 | 3.33333 pt | 3.83331 pt | 3.91663 pt | 5.24995 pt |
| 3 | 1.66666 pt | 1.91666 pt | 1.95831 pt | 0.0 pt |
| 4 | 1.11111 pt | 1.27777 pt | 1.30554 pt | 0.0 pt |
| 7 | 1.11111 pt | 1.27777 pt | 1.30554 pt | 5.24995 pt |
| 6 | 10.00002 pt | 11.49994 pt | 11.74988 pt | 10.4999 pt |

The `\fontdimen` parameters change when different sizes of the same font are used, as shown in the column for `cmr12` compared to the values for `cmr10`. For the *Computer Modern* fonts the relations $f_3 = f_2/2$, $f_4 = f_2/3$, and $f_7 = f_4$ seem to hold except for the monospaced font `cmtt10`. But it is better to express this relationship in terms of $f_6$, the quad width, also known as 1 em: $f_2 = f_6/3$, $f_3 = f_6/6$, $f_4 = f_6/9$, and $f_7 = f_6/9$.

**The ligtable.** The values must be treated as constants. For the font `cmr10` $L(\texttt{L}\&\texttt{-}) = L(\texttt{L-}\&\texttt{M}) = 0\,\mathrm{pt}$ as explained in section 5. The dimension $L(\texttt{L}\&\texttt{M})$ can be taken from Fig. 8. The values are small but they can be negative or positive.

The easiest way to get the values is to look at the *property list* of a font, which can be generated from the tfm file with the utility TFtoPL [7]. But note, sometimes letter pairs might have more than one entry, for example, in `cmr10` two values for the letter pair “ka” are specified [6, p. 37]. The first value counts, as explained in [5, p. 317]; so it does not seem to be a “mistake” [2, p. 322], but rather an optimization [11].

$L(\texttt{C}\&\texttt{l})$ when letter `l` follows input of column `C`

| `C` | −0.55556 pt | −0.27779 pt | 0.27779 pt | 0.55554 pt |
|---|---|---|---|---|
| a | | v w y | | j |
| b | | v w x y | c d e o q | j |
| c | | c k | | |
| f | | f i l | | |
| ff | | i l | | |
| g | | | j | |
| h | | b t u v w y | | |
| k | a | c e o | | |
| m | | b t u v w x y | | |
| n | | b t u v w x y | | |
| o | | v w x y | c d e o q | j |
| p | | v w x y | c d e o q | j |
| t | | w y | | |
| u | | w | | |
| v | a | c e o | | |
| w | | a c e o | | |
| y | | a e o | | |

**Figure 8:** Measured ligature and kerning values of lowercase letters in `cmr10`

## 8 Paragraph shape parameters

The last group of parameters in (12) is formed by **l** and **r**, the `\leftskip` and the `\rightskip`. Sometimes these skips are used only with their natural width, for example, in the command `\narrower` but in (12) only the stretchability and shrinkability of the glue counts.

It might have been a surprise that the dimension `\hsize` has no influence on the required value of `\emergencystretch`. At least there is a basic relationship between the two as

$$0 \le \frac{\text{optimal value of }\texttt{\textbackslash emergencystretch}}{\text{value of }\texttt{\textbackslash hsize}} \le 1.$$

Therefore, the parameter `\hsize` is also analyzed.

**No influence: `\hsize`.** Of course, a change of the `\hsize` means different line breaks and that influences other parameters. Often overfull lines go away when the `\hsize` is changed. In this sense this dimension has influence. But when a concrete line-breaking situation is given — which is the precondition for the analysis that leads to (12) — it doesn't have any influence.

In the next experiment a “comparable” line-breaking situation for a larger `\hsize` is used. By “comparable” I mean a situation in which the line breaks are the same as with the previous `\hsize`. Therefore the content must be changed: Each line gets more or less in its middle part a rule with a length equal to the increment of `\hsize`.

**Experiment 6: Description**
The `\hsize` is increased by 40 pt. As input, an instrumented version of the text of experiment 1 is used: In each line the control symbol `\0`, which represents a 40 pt long rule, is added before one word.

**TEX definitions**

```
\hsize=140pt \def\0{\hbox to 40pt{\hrulefill}}
```

**TEX input**

```
Once upon \0a time, in \0a distant galaxy
called \"O\"o\c c, there \0lived a computer
\0named R.~J. Drofnats.

Mr.~Drofnats\0---or ‘‘R. J.,’’ as \0he preferred
to be \0called---was happiest when \0he was at
work \0typesetting beautiful documents.
```

**TEX output**

Once upon ________a time,
in ________a distant galaxy called $o = 5.88911$ pt
Ööç, there ________lived a com-
puter ________named R. J. Drof- $o = 2.13905$ pt
nats.
Mr. Drofnats________—or “R. $o = 12.36128$ pt
J.,” as ________he preferred to
be ________called—was happiest $o = 2.3335$ pt
when ________he was at work
________typesetting beautiful doc- $o = 10.61136$ pt
uments.

**Result (experiment 1 vs. experiment 6)**
a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt 9.3 pt/11.2 pt 1.5 pt 0.2 pt/1.5 pt 1.5 pt
6a): `\hsize`=140 pt with instrumented content
b): 9.3 pt 9.3 pt/11.2 pt 1.5 pt 0.2 pt/1.5 pt 1.5 pt ▯

As expected, the experiment generates the identical values for eliminating overfull lines.

**Ragged-right setting.** As mentioned above the `\fontdimen` values should only be changed in very unusual situations; an example is given in [3], p. 355. If a change of the interword space is required the internal glue registers `\spaceskip` and `\xspaceskip` can be used as explained in section 2. An application occurs when text is typeset ragged-right with the help of a stretchable `\rightskip`.

The `plain` macro `\raggedright` initializes the ragged-right settings:

```
\def\raggedright{\rightskip=0pt plus2em
  \spaceskip=.3333em \xspaceskip=.5em\relax}.
```

The interword spaces are fixed and the `\rightskip` has a large value of stretchability: 2 em is $12f_3$ in `cmr10`. So it is not surprising that the text of experiment 1 can be typeset ragged right without any overfull lines. But with an `\hsize` of 108 pt case a) returns.

**Experiment 7: Description**
Change the `\hsize` to 108 pt and typeset the text ragged right.

**TEX definitions**

```
\hsize=108pt \raggedright
```

**TEX output**

Once upon a time,
in a distant galaxy called $o = 2.33217$ pt
Ööç, there lived a com-
puter named R. J. Drof-
nats.
Mr. Drofnats—or
“R. J.,” as he preferred
to be called—was happi-
est when he was at work
typesetting beautiful
documents.

**Result (experiment 1 vs. experiment 7)**
a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt 9.3 pt/11.2 pt 1.5 pt 0.2 pt/1.5 pt 1.5 pt
7a): `\hsize`=108 pt and `\raggedright`
b): 0.6 pt ▯

The theory developed in section 4 applies as all the different space types were included in the analysis; (12) computes the value: 0.6 pt.

## 9 More theory

Three cases of experiment 1 remain: b), d′), and e). In these cases text is not only moved to the next line but the preceding line hands over new material to the previously overfull line. This changes the picture completely as one of the basic elements on which the theory of section 4 is based is no longer true: The line that appears after applying the minimal `\emergencystretch` is no longer stretched to its maximum. See (3ii) for the case e) where the new badness is 2 and the white space *shrinks* as (3iv′) shows. I doubt that there is a precise approximation to calculate $e$ from the output that contains the overfull line only if new material is added to the line.

As the cases b) and b′) show, it can happen that an overfull line is transformed into another overfull line before the computed `\emergencystretch` resolves the original problem. And the newly created overfull line needs a larger additional stretchability than the original one. Therefore a second calculation of $e$ is sometimes necessary. The repeated application of the theory developed in section 4 is required.

**A procedure.** Compute with (12) all values for the overfull lines. Use the smallest value and apply it to the paragraph. For the selected overfull line and the other overfull lines the following possibilities exist.

1: Text is moved from the overfull line to the next line, no material comes from the previous line. This scenario was analyzed in section 4. There the problem is solved for the selected overfull line. Two results are possible:
   i) The computed value $e$ works and no other overfull line is created — success. The cases a), b′), and c) belong to this scenario. Other overfull lines might disappear too, this happens in the cases d′) and e).
   ii) The computed value $e$ works for the selected line, but a new overfull line is created. This is the case d); the procedure must be repeated.

2: Text is moved from the selected overfull line and new material is added to it. The computed value $e$, which would work for the original line, is not relevant to the changed line.
   i) The overfull line is gone and no other overfull line is created — success, but a smaller value of $e$ might be successful too. No case of this kind occurs in experiment 1.
   ii) The overfull line is gone, but a new overfull line is created. This is the case b). The procedure must be repeated.

The procedure terminates as either the number of overfull lines is decreased or the new overfull line occurs later in the paragraph.

The remaining cases of experiment 1 are handled by this procedure. The only scenario that must be looked at further is 2i) as in other cases at most the procedure must be restarted.

The main question is: Why is some new material moved to the overfull line? Of course, TeX can use the value of `\emergencystretch` to stretch one or more previous lines with less text so that it finds line breaks which avoid the overfull line.

**The missing scenario.** This scenario is described more formally in order to have a clear definition. As the way in which the lines are broken, i.e., whether at glue, at penalty, at an explicit hyphen, or at an inserted hyphen, is not important in the following discussion that part is ignored. Assume that the line $\mathtt{L}_0\mathtt{M}_0$ is overfull and the method of section 4 has computed $e_0$. But when `\emergencystretch` is set to that value the former overfull line doesn't become $\mathtt{L}_0$ but $\mathtt{M}_{-1}\mathtt{L}_0$. Let $-\omega$ be the first preceding line that has not received material from its previous line; such a line must exist as the first line of the paragraph qualifies.

So the lines $\mathtt{L}_{-\omega}\mathtt{M}_{-\omega}$, $\mathtt{L}_{-\omega+1}\mathtt{M}_{-\omega+1}$, …, $\mathtt{L}_0\mathtt{M}_0$, in which the last line is overfull, are transformed into the lines $\mathtt{L}_{-\omega}$, $\mathtt{M}_{-\omega}\mathtt{L}_{-\omega+1}$, …, $\mathtt{M}_{-1}\mathtt{L}_0$.

Some of the new lines must stretch to benefit from $e$; let the index sequence $\kappa_{-\mu}, \ldots, \kappa_0$ represent the indices of lines that stretch. We have to calculate an $e_{\kappa_\iota}$ for all of these lines — although except for one they might not be overfull — to stretch them by the expected amount. The maximum dimension of these calculations is sufficient for `\emergencystretch` instead of the originally calculated $e_0$ to make all transitions happen.

Therefore two formulas are needed. One that computes $e_0$ from the transition $\mathtt{L}_0\mathtt{M}_0 \to \mathtt{M}_{-1}\mathtt{L}_0$; it can only give a useful result if $\mathtt{M}_{-1}\mathtt{L}_0$ stretches. The second formula computes the required amount of `\emergencystretch` that is needed to transform a non-overfull line into another: $\mathtt{L}_{\kappa_\iota}\mathtt{M}_{\kappa_\iota} \to \mathtt{M}_{\kappa_\iota-1}\mathtt{L}_{\kappa_\iota}$. The computation is valid only if $\mathtt{M}_{\kappa_\iota-1}\mathtt{L}_{\kappa_\iota}$ stretches.

**The first formula.** Figure 9 shows how the cases that were discussed in section 4 change when new material is added to the right side. Note the new material N might end in glue. It must be the empty string if L stands for the first line of a paragraph. So there is no problem with an `\indent` or `\noindent`.

$$\begin{array}{|c|c|}\hline \multicolumn{2}{|c|}{l_{\mathrm{mt}}\,W_{\mathrm{mt}}(\mathtt{L})\,g_{\mathrm{mt}}\,W_{\mathrm{mt}}(\mathtt{M})\,r_{\mathrm{mt}}} \\ \hline h & o \\ \end{array} \to \begin{array}{|c|}\hline l\,W(\mathtt{NL})\,r \\ \hline h \\ \end{array}$$

$$\begin{array}{|c|c|}\hline \multicolumn{2}{|c|}{l_{\mathrm{mt}}\,W_{\mathrm{mt}}(\mathtt{L})\,\pi\,W_{\mathrm{mt}}(\mathtt{M})\,r_{\mathrm{mt}}} \\ \hline h & o \\ \end{array} \to \begin{array}{|c|}\hline l\,W(\mathtt{NL})\,r \\ \hline h \\ \end{array}$$

$$\begin{array}{|c|c|}\hline \multicolumn{2}{|c|}{l_{\mathrm{mt}}\,W_{\mathrm{mt}}(\mathtt{KM})\,r_{\mathrm{mt}}} \\ \hline h & o \\ \end{array} \to \begin{array}{|c|}\hline l\,W(\mathtt{NL-})\,r \\ \hline h \\ \end{array}$$

**Figure 9:** Resolve overfull line with new material

The right hand sides must stretch and they must need `\emergencystretch`, i.e., each real badness $\varrho$ must be greater than $\tau$ to benefit from the following theory.

As only the right hand sides change, all equations for the left hand sides developed in section 4 are still valid. In the equations for the right hand sides the input L must be replaced by NL. The equation $W_{\mathrm{nw}}(\mathtt{NL}) = W_{\mathrm{nw}}(\mathtt{N}) + L(\mathtt{N\&L}) + W_{\mathrm{nw}}(\mathtt{L})$ shows that only two new terms occur in these equations; the equation is valid as L doesn't begin with ignored spaces as explained above.

Equation (6) changes to

$$\begin{aligned} a &= l^+ + S^+_\epsilon(\mathtt{NL}) + r^+ \\ &= l^+ + S^+_\epsilon(\mathtt{N}) + S^+_\epsilon(\mathtt{L}) + r^+. \end{aligned} \tag{13}$$

The optimal value for `\emergencystretch`

Equation (10) is no longer valid, now the computation is:

$$\begin{aligned} h &= l + W(\mathtt{NL}) + r \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{NL}) + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ + W_{\mathrm{nw}}(\mathtt{N}) + L(\mathtt{N}\&\mathtt{L}) + u. \end{aligned}$$

The equations for the left hand sides stay the same and with the techniques of section 4 the equivalent of (8) is:

$$\begin{aligned} u = W_{\mathrm{nw}}(\mathtt{M}) &- S^-_\epsilon(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ &- W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N}\&\mathtt{L}) \\ &+ \bigl(G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\bigr)\Gamma. \end{aligned}$$

The first two cases are handled; the last one is not difficult but the argument must involve the strings `K` and `K'`. One more figure might help:

$$\begin{array}{|c|c|} \hline l_{\mathrm{mt}}\, W_{\mathrm{mt}}(\mathtt{KM})\, r_{\mathrm{mt}} & \\ \hline h & o \\ \end{array} \rightarrow \begin{array}{|c|} \hline l\, W(\mathtt{NK'})\, L(\mathtt{NK'}\&\mathtt{-})\, W(\mathtt{-})\, r \\ \hline h \\ = \\ l\, W(\mathtt{NL-})\, r \\ \hline h \end{array}$$

**Figure 10:** Break at hyphen and add new material

The replacement of (9) must deal with `NK` and `NK'`. The relationship becomes

$$\begin{aligned} W_{\mathrm{nw}}(\mathtt{NK}) + \bigl(L(\mathtt{NK}\&\mathtt{-}) + W(\mathtt{-})\bigr)\Theta &= \\ W_{\mathrm{nw}}(\mathtt{NK'}) + L(\mathtt{NK'}\&\mathtt{-}) &+ W(\mathtt{-}) \end{aligned}$$

and for the right hand side the new formula is

$$\begin{aligned} h &= l^\circ + W_{\mathrm{nw}}(\mathtt{NL-}) + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{NK'}) + L(\mathtt{NK'}\&\mathtt{-}) + W(\mathtt{-}) + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{NK}) + \bigl(L(\mathtt{NK}\&\mathtt{-}) + W(\mathtt{-})\bigr)\Theta + r^\circ + u \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{N}) + L(\mathtt{N}\&\mathtt{K}) + W_{\mathrm{nw}}(\mathtt{K}) \\ &\qquad + \bigl(L(\mathtt{NK}\&\mathtt{-}) + W(\mathtt{-})\bigr)\Theta + r^\circ + u \end{aligned}$$

so

$$\begin{aligned} l^\circ + W_{\mathrm{nw}}(\mathtt{K}) + r^\circ = h &- \bigl(L(\mathtt{NK}\&\mathtt{-}) - W(\mathtt{-})\bigr)\Theta - u \\ &- W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N}\&\mathtt{K}). \end{aligned}$$

The equation for the left hand side remains valid and its first three summands are replaced by the right hand side of the previous equation:

$$\begin{aligned} h + o = h &- \bigl(L(\mathtt{NK}\&\mathtt{-}) + W(\mathtt{-})\bigr)\Theta - u \\ &- W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N}\&\mathtt{K}) - l^- - S^-_\epsilon(\mathtt{K}) - r^- \\ &+ L(\mathtt{K}\&\mathtt{M}) + W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}). \end{aligned}$$

The final rearrangement involves the replacement of `K` by `L` as was done before.

$$\begin{aligned} u = W_{\mathrm{nw}}(\mathtt{M}) &- S^-_\epsilon(\mathtt{M}) - o - l^- - S^-_\epsilon(\mathtt{L}) - r^- \\ &- W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N}\&\mathtt{L}) \\ &+ \bigl(L(\mathtt{L}\&\mathtt{M}) - L(\mathtt{NL}\&\mathtt{-}) - W(\mathtt{-})\bigr)\Theta \\ &+ L(\mathtt{L-}\&\mathtt{M})\,\Xi. \end{aligned}$$

**Summary.** Two new summands are added compared to (10), and all the formulas can be combined as in section 4 to get the equivalent of approximation (12):

$$\begin{aligned} e \approx \sqrt[3]{\frac{100}{\tau}}\Bigl(&W_{\mathrm{nw}}(\mathtt{M}) - S^-_\epsilon(\mathtt{M}) - o - l^- - r^- \\ &- S^-_\epsilon(\mathtt{L}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N}\&\mathtt{L}) \\ &+ \bigl(G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) - G^-_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\bigr)\Gamma \\ &+ \bigl(L(\mathtt{L}\&\mathtt{M}) - L(\mathtt{NL}\&\mathtt{-}) - W(\mathtt{-})\bigr)\Theta \\ &+ L(\mathtt{L-}\&\mathtt{M})\,\Xi\Bigr) \\ &- l^+ - S^+_\epsilon(\mathtt{N}) - S^+_\epsilon(\mathtt{L}) - r^+ \end{aligned} \tag{14}$$

if $\varrho > \tau$ and if there is stretchability in the line, i.e., $l^+ + S^+_\epsilon(\mathtt{N}) + S^+_\epsilon(\mathtt{L}) + r^+ > 0\,\mathrm{pt}$.

**The second formula.** Now the situation changes completely as — to speak in the terms that we have been using — the left hand side is not overfull.

**Case 1: Break at glue.** Let's look at the situation again in the form of a simple picture, Fig. 11, the companion of Fig. 3.

$$\begin{array}{|c|} \hline l\, W(\mathtt{L})\, g\, W(\mathtt{M})\, r \\ \hline h \end{array} \rightarrow \begin{array}{|c|} \hline l\, W(\mathtt{NL})\, r \\ \hline h \end{array}$$

**Figure 11:** Break at glue and add new text

As before the text `N` might end in a glue item; the glue **g** might be user-entered or TeX-inserted.

Much of the information from the above Fig. 3 is lost. But on both sides the badness values are now finite, and as observed in the previous subsection for the first formula, the right hand side must stretch; further, it must stretch so much that the additional stretchability of `\emergencystretch` is required. In other words: The real badness $\varrho$ of the right hand side must be larger than `\tolerance`. The glue of the left side has no restriction, i.e., it can shrink, stretch or use its natural width. To distinguish the common variable names for the right and the left sides the subscripts $\rho$ and $\lambda$ are used for $a$ and $u$.

The right hand sides have been analyzed for the first formula and the value $a_\rho$ is given by (13); as before, both sides give an equation for $h$:

$$\begin{aligned} h &= l^\circ + W_{\mathrm{nw}}(\mathtt{N}) + L(\mathtt{N}\&\mathtt{L}) + W_{\mathrm{nw}}(\mathtt{L}) + r^\circ + u_\rho. \\ &= l^\circ + W_{\mathrm{nw}}(\mathtt{L}) + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}) + W_{\mathrm{nw}}(\mathtt{M}) + r^\circ \\ &\qquad + \delta u_\lambda. \end{aligned}$$

The factor $\delta \in \{-1, +1\}$ represents the fact that the value $u_\lambda$ has to be added if the line on the left stretches otherwise it must be subtracted.

The equations can be rearranged to find the equation for $u_\rho$:

$$u_\rho = W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N\&L}) + \delta u_\lambda + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g}).$$

The finite badness values allow to write the following approximations (compare to (5)):

$$\rho \approx 100 \left(\frac{u_\rho}{a_\rho + e}\right)^3 \quad \text{or } u_\rho \approx \sqrt[3]{\frac{\rho}{100}}\,(a_\rho + e).$$

As above $\rho = \tau$ to get the minimal $e$:

$$e \approx \sqrt[3]{100/\tau}\, u_\rho - a_\rho. \tag{$**$}$$

Therefore

$$e \approx \sqrt[3]{\frac{100}{\tau}} \Big(W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N\&L}) + \delta u_\lambda + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\Big) - l^+ - S^+_\epsilon(\mathtt{N}) - S^+_\epsilon(\mathtt{L}) - r^+$$

if stretchability is present in the line.

The term $\delta u_\lambda$ could be analyzed further, but that leads to a lot of subcases. I think it is best to keep it in the formula.

**Case 2: Break at penalty.** This case is handled exactly like the previous one. If $\Gamma$ is introduced, as in section 4, then the combined equation becomes

$$e \approx \sqrt[3]{\frac{100}{\tau}} \Big(W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N\&L}) + \delta u_\lambda + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\Gamma\Big) - l^+ - S^+_\epsilon(\mathtt{N}) - S^+_\epsilon(\mathtt{L}) - r^+.$$

**Case 3: Break at hyphen.** Again the formula for $a_\rho$ is given by (13).

$$\begin{array}{c} \boxed{l\,W(\mathtt{KM})\,r} \\ \boxed{h} \end{array} \rightarrow \begin{array}{c} \boxed{l\,W(\mathtt{NK}')\,L(\mathtt{NK}'\&\mathtt{-})\,W(\mathtt{-})\,r} \\ \boxed{h} \\ = \\ \boxed{l\,W(\mathtt{NL-})\,r} \\ \boxed{h} \end{array}$$

**Figure 12:** Break at hyphen and add new text

The right hand side was analyzed above for the first formula and the left hand side gives the equation:

$$h = l^\circ + W_{\mathrm{nw}}(\mathtt{K}) + L(\mathtt{K\&M}) + W_{\mathrm{nw}}(\mathtt{M}) + r^\circ + \delta u_\lambda.$$

Therefore

$$u_\rho = W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) + L(\mathtt{K\&M}) - L(\mathtt{N\&K}') - \big(L(\mathtt{NK\&-}) + W(\mathtt{-})\big)\Theta + \delta u_\lambda$$

and with the replacement of `K` by `L` as it was done for the first formula it becomes

$$u_\rho = W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N\&L}) - \big(L(\mathtt{NL\&-}) - L(\mathtt{L\&M}) + W(\mathtt{-})\big)\Theta + L(\mathtt{L-\&M})\,\Xi + \delta u_\lambda.$$

**Summary.** The previous equation can be applied to ($**$). The result isn't shown here as all cases can be combined directly as before into one equation. The approximation of $e$ for the second formula is

$$e \approx \sqrt[3]{100/\tau}\,\Big(W_{\mathrm{nw}}(\mathtt{M}) - W_{\mathrm{nw}}(\mathtt{N}) - L(\mathtt{N\&L}) + \delta u_\lambda + G^\circ_\epsilon(\Phi(\mathtt{L}), \mathtt{g})\Gamma + \big(L(\mathtt{L\&M}) - L(\mathtt{NL\&-}) - W(\mathtt{-})\big)\Theta + L(\mathtt{L-\&M})\,\Xi\Big) - l^+ - S^+_\epsilon(\mathtt{N}) - S^+_\epsilon(\mathtt{L}) - r^+ \tag{15}$$

if $\varrho > \tau$ and $-l^+ - S^+_\epsilon(\mathtt{N}) - S^+_\epsilon(\mathtt{L}) - r^+ > 0\,\mathrm{pt}$.

## 10 A second experiment

The theory is completely developed. The following experiment contains aspects not seen before, for example, an overfull line in math mode and a break at an explicit hyphen, and it shows a complete cycle to get rid off all overfull lines in a paragraph according to the theory of section 9.

**Experiment 8: Description**

Combine both paragraphs of experiment 1 and add a sentence with mathematics. Reduce `\hsize` by 2 pt.

**TeX definitions**

```
\hsize=98pt
```

**TeX output**

Once upon a time, in
a distant galaxy called
Ööç, there lived a com-
puter named R. J. Drof- $e_1 \approx 9.1\,\mathrm{pt}$ by (12)
nats. Mr. Drofnats—
or "R. J.," as he pre-
ferred to be called—was $e_2 \approx 1.8\,\mathrm{pt}$ by (12)
happiest when he was
at work typesetting beau- $e_3 \approx 4.1\,\mathrm{pt}$ by (12)
tiful documents. In one $e_4 \approx 4.1\,\mathrm{pt}$ by (12)
text he proved $e^{i\pi}+1 =$ $e_5 \approx 0.6\,\mathrm{pt}$ by (12)
0.

As in experiment 1 there are five overfull lines. To remove them the procedure of section 9 that is based on the results of section 4 can be used. As described, $e_5$, the smallest of the five values, can be applied first, which of course resolves the last overfull line and only this line. Then the application of the next smallest value, $e_2$, resolves the second and fourth overfull lines and transfers the third one into a new overfull line. And so on.

But with the formulas of the previous section the maximum value can be used directly and if it is too large the value can be corrected; so the largest value $e_1$ is assigned to `\emergencystretch`.

The optimal value for `\emergencystretch`

**Experiment 8 continued: TEX definitions**

`\emergencystretch=9.1pt`

**TEX output**

Once upon a time,
in a distant galaxy
called Ööç, there lived
a computer named R. J. $e_6 \approx 9.6\,\text{pt}$ by (12)
Drofnats. Mr. Drofnats— $e_7 \approx -0.0\,\text{pt}$ by (12)
or "R. J.," as he pre-
ferred to be called—
was happiest when he
was at work typeset-
ting beautiful docu-
ments. In one text he
proved $e^{i\pi} + 1 = 0$.

The value of $e_1$ was too small to solve all problems; new overfull lines were created.

Note the value of $e_7$ is given as $-0.0\,\text{pt}$ — an impossible result. The reason lies in the content of the line. The space after "`Mr.`" in line 5 is entered as a tie (see the TEX input of experiment 1) so the line cannot be broken there. And "`Drofnats---`" cannot be hyphenated as it contains, in TEX's view, an explicit hyphen. Therefore the line break must occur after the end-of-sentence period. With the theory of section 4 this results in a line that would contain only "`Drofnats.`", i.e., the line has no stretchability. Therefore approximation (12) cannot compute a valid value for the `\emergencystretch` — expressed by the impossible value $-0.0\,\text{pt}$.

But as the other line reports a valid dimension $e_6$, this can be used in the next run to remove at least this overfull line.

**Experiment 8 continued: TEX definitions**

`\emergencystretch=9.6pt`

**TEX output**

Once upon a time,
in a distant galaxy
called Ööç, there lived
a computer named
R. J. Drofnats. Mr. Drofnats— $e_8 \approx 22.5\,\text{pt}$ by (12)
or "R. J.," as he pre-
ferred to be called—
was happiest when he
was at work typeset-
ting beautiful docu-
ments. In one text he
proved $e^{i\pi} + 1 = 0$.

There is still one overfull line. The value 9.6 pt was too small but line 5 now gives a usable value as two spaces occur in the line after a break at the end-of-sentence period. The value $e_8$ removes the overfull line.

**Experiment 8 continued: TEX definitions**

`\emergencystretch=22.5pt`

**TEX output**

Once upon a time, $e_9 \approx 8.2\,\text{pt}$ by (15)
in a distant galaxy $e_{10} \approx 7.7\,\text{pt}$ by (15)
called Ööç, there $e_{11} \approx 15.4\,\text{pt}$ by (15)
lived a computer $e_{12} \approx 16.2\,\text{pt}$ by (15)
named R. J. Drofnats. $e_{13} \approx -0.0\,\text{pt}$ by (14)
Mr. Drofnats—or "R.
J.," as he preferred to
be called—was hap-
piest when he was
at work typesetting
beautiful documents.
In one text he proved
$e^{i\pi} + 1 = 0$.

This is the scenario 2i) of the procedure in section 9. The check with the four preceding lines of the previously overfull line shows that the value 16.2 pt is sufficient to resolve this problem as the input "`named `" is moved to that line. The output for the `\emergencystretch` set to 16.2 pt isn't shown as it is identical to the version with 22.5 pt.

Note the value $e_{13}$ cannot be computed as the former overfull line is transformed into a line of badness 0. Formula (14) needs a line that stretches and has a real badness larger than `\tolerance`.

## 11 Another application

In [14, example 5], `\emergencystretch` — together with `\looseness` to have a trigger for a third line-breaking pass — was used to remove a stack of hyphens. The first three lines of a paragraph end with an inserted hyphen. In the example the value of `\emergencystretch` was set to the width of the string "`Ar`", i.e., to 11.41669 pt to remove the stack. The text is taken from [9].

**Experiment 9: Description**

Remove a stack of hyphens via `\emergencystretch`; first show the original line breaking and then give the additional stretchability the amount 11.4 pt. Force TEX to use a third pass with the setting `\looseness` = 1.

**TEX output**

So instead, I worked only at Stanford, at the Artificial Intelligence Laboratory with the very primitive equipment there. We did have television cameras, and my publisher, Addison-Wesley, was very helpful — they sent me the original press-printed proofs of my book, from which *The Art of Computer Programming* had been made. The process in the 60s …

So instead, I worked only at Stanford, at the Artificial Intelligence Laboratory with the very primitive equipment there. We did have television cameras, and my publisher, Addison-Wesley, was very helpful — they sent me the original press-printed proofs of my book, from which *The Art of Computer Programming* had been made. The process in the 60s ... ▯

No overfull lines are output, but the second formula of section 9 calculates $e$ for a transition that doesn't involve overfull lines. Here are the values for the first three lines of the second paragraph of the above output:

1st line: $e_1 \approx -0.0\,\text{pt}$ by (15)
2nd line: $e_2 \approx 6.9\,\text{pt}$ by (15)
3rd line: $e_3 \approx -0.0\,\text{pt}$ by (15)

So let's try what happens when the value of the dimen `\emergencystretch` is set only to 6.9 pt.

Surprise — nothing happens! The output looks identical to the first paragraph. Well, maybe it is not that much of a surprise as one of the principles on which the derived formula (15) is based is violated: TEX does not have to avoid an overfull line, and therefore it uses the line breaks that result in the minimal sum of the line demerits. The total for the text without the three hyphens is nearly 36,000 demerits higher than for the shown paragraph with $\approx$ 11.4 pt additional stretchability. Note that the second line, which determines the value of $e$, gets a badness of 200 by (15), the default `\tolerance`.

Either the penalty for hyphenation must be increased, for example, to 200 to compensate for the value of `\tolerance`, or `\emergencystretch` must be increased because then the lines that stretch get lower badness values, as observed in experiment 1. The correct amounts are not easy to determine. For example, setting `\hyphenpenalty` to 128 solves the problem, or setting `\emergencystretch` to the minimal 10.9 pt to create the output as shown above, or to 9.8 pt to get a solution that removes the three hyphens but creates a new one.

## 12 Final remarks

In this article theoretical results show how to compute the optimal value for `\emergencystretch` in the case when TEX produces an overfull line. Approximation (12) is always successful as long as the line contains stretchability and (14) and (15) extend the result but need an additional condition: the badness of the new line must be larger than the given `\tolerance`. (This condition is automatically fulfilled when (12) can be used.) The analysis provides some insight into the factors that influence the value for the additional stretchability.

One question has not been addressed yet: Why is a minimal value useful? As explained in [14], and observed in experiment 1, TEX uses the badness value computed from the available stretchability for its line-breaking decision and the stretchability that comes from `\emergencystretch`. High values for the latter assign to every stretchable line a low badness value, and thus TEX starts to prefer stretched lines in the paragraph during the line-breaking procedure, resulting in an excess of spaced-out lines.

Everyone who wants to use a nonzero value for `\emergencystretch` should be aware that an acceptable value depends on the available glue in a line — or in other words: the number of spaces in the line. A high average number of spaces in a text can tolerate higher values of `\emergencystretch`. And a line break with an inserted hyphen is an advantage. Although the following advice is not obeyed in all of the experiments in this article I recommend to apply a positive `\emergencystretch` only to a single paragraph at a time.

**A rule of thumb.** The accuracy with which $e$ was computed in this article is not required, though. This allows us to create a *rule of thumb* to make the theory useful in everyday applications. A close look at (12) shows that many of the summands are zero or small if the `plain` TEX defaults and `cmr10` is used. Only four values are needed:

1. the amount that the line is too wide: $o\,\text{pt}$,
2. the number of characters in `M` which are moved to the next line,
3. the number of spaces in `L`, i.e., in the rest of the line,
4. the type of break: glue, penalty, explicit or inserted hyphen.

The rule of thumb is not very simple, maybe it is too complicated to be easily remembered:

$$\begin{aligned} 4e \approx (&15 \times \text{chars in } \mathtt{M} - 11 \times \text{spaces in } \mathtt{L} \\ &- 3 \times o + 8 \\ &- 18 \text{ if a hyphen is inserted})\,\text{pt}. \end{aligned} \tag{RoT}$$

Of course, if the number of spaces in `L` is zero, the computed value for $e$ makes no sense as the input `L` has no stretchability.

For example, in experiment 1 the message

```
Overfull \hbox (5.88911pt too wide) in paragraph
\tenrm in a dis-tant galaxy called|
```

is shown and therefore $o \approx 5.89$, 6 characters are moved and three spaces remain when "called" is moved at a break at glue. Now use (RoT) to compute $4e \approx (90 - 33 - 17.67 + 8)\,\text{pt} = 47.33\,\text{pt}$ and therefore $e \approx 11.8\,\text{pt}$.

The other cases are treated in the same way.

The results compared to the measured values of experiment 1 are:

a) parameters; b) used `\emergencystretch`
1a): `\hsize`=100 pt
b): 9.3 pt 9.3 pt/11.2 pt 1.5 pt 0.2 pt/1.5 pt 1.5 pt
RoT a): calculated values with the *rule of thumb*
b): 11.8 pt 10.8 pt/13.7 pt 4.9 pt 1.4 pt/4.4 pt 6.2 pt

All computed values are large enough to remove the overfull line. Nevertheless there can be cases, for example, `M` is "WWW", when the factor 15 might be too small — although it looks as if the RoT computes values that are often too large. The simplification compared to (12), (14), and (15) comes with a price.

**Aesthetics.** The results given in this article are rather technical, and do not consider any aesthetic aspects. Of course, aesthetics cannot be measured objectively, as different individuals will prefer different aspects: hyphenation or not, justified or ragged-right text, etc.

This paper is already too long to discuss this, but one more TeX parameter might be briefly mentioned: `\hfuzz`. This has nothing to do with line-breaking decisions, it is used to suppress the warning message for overfull lines, nothing more. As long as an overfull line juts less than `\hfuzz` into the margin, TeX does not output a warning message. The `plain` format sets `\hfuzz` to 0.1 pt. Note that in section 10 the line preceding the overfull line, for which $e_1$ is calculated, is overfull by 0.02792 pt. It is not mentioned by TeX as the value is less than 0.1 pt. Sometimes higher values, for example, 1 pt, are suggested. But an overfull line that is 1 pt too wide is often easily detected by the eye.

An author may change his text to improve aesthetics but of course typesetting problems will always be present. Whatever the author or typesetter prefers or dislikes in a specific situation a manual intervention is sometimes necessary. Never forget: The visual output counts; it must always be checked, especially when `\emergencystretch` is used.

## References


[1] Malcolm Clark, *A plain TeX primer*, Oxford University Press, 1992

[2] Bogusław Jackowski, Piotr Strzelczyk and Piotr Pianowski, "GUST e-foundry font projects," *TUGboat* **37**:3 (2016), 317–336
`tug.org/TUGboat/tb37-3/tb117jackowski.pdf`

[3] Donald E. Knuth, *The TeXbook*, Volume A of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1984
The text of experiment 1 can be found in
`ctan.org/tex-archive/systems/knuth/dist/lib/story.tex`

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

[5] Donald E. Knuth, *The METAFONTbook*, Volume C of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986

[6] Donald E. Knuth, *Computer Modern Typefaces*, Volume E of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986

[7] Donald E. Knuth, "The TFtoPL processor," in *TeXware*, Stanford Computer Science Report *STAN-CS-86-1097*, Stanford, California: Stanford University, 1986 (The first TFtoPL program was designed by Leo Guibas in the summer of 1978)
`ctan.org/pkg/texware`

[8] Donald E. Knuth, "The New Versions of TeX and METAFONT," *TUGboat* **10**:3 (1989), 325–328; "Erratum: The New Versions of TeX and METAFONT *TUGboat* Vol. 10, No. 3," *TUGboat* **11**:1 (1990), 12; reprinted as Chapter 29 in [10], 563–570
`tug.org/TUGboat/tb10-3/tb25knut.pdf`
`tug.org/TUGboat/tb11-1/tb27erratum.pdf`

[9] Donald E. Knuth, "CSTUG, Charles University, Prague, March 1996: Questions and Answers with Prof. Donald E. Knuth," *TUGboat* **17**:4 (1996), 355–367; reprinted as Chapter 32 in [10], 601–624
`tug.org/TUGboat/tb17-4/tb53knuc.pdf`

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

[11] Jerry Leichter, "Re: Strange kern in cmr10," *TeXhax Digest* **89**:70 (1989), 25 July 1989
`ctan.org/tex-archive/info/digests/texhax/89/texhax.70.gz`

[12] Frank Mittelbach, Chris Rowley, Michael Downes, "Volunteer work for the LaTeX3 project," *TUGboat* **13**:4 (1992), 510–513
`tug.org/TUGboat/tb13-4/tb37mitt-l3.pdf`
`ctan.org/tex-archive/info/ltx3pub/vol-task.tex`

[13] Philip Taylor, "A Pragmatic Approach to Paragraphs," *TUGboat* **14**:2 (1993), 138–140
`tug.org/TUGboat/tb14-2/tb39taylor-para.pdf`

[14] Udo Wermuth, "Tracing paragraphs," *TUGboat* **37**:3 (2016), 358–373
`tug.org/TUGboat/tb37-3/tb117wermuth.pdf`



◇ Udo Wermuth
Babenhäuser Straße 6
63128 Dietzenbach
Germany
`u dot wermuth (at) icloud dot com`