

The `showhanmengrid` package

Show the hanmen grid of Japanese text and its overflow on every page

Munehiro Yamamoto

<https://github.com/munepi/showhanmengrid>

v1.1 (2026/10/25)

1 Introduction

The *hanmen* (kihon hanmen, 基本版面), the basic text block of Japanese typesetting, is built from four values: the character size (1zw, the width of a full-width character), the number of characters per line, the number of lines per page and the line feed. When

- `\textwidth` is a multiple of 1zw, and
- `\textheight` is `\topskip` plus a multiple of `\baselineskip`,

every character of the body text sits on a fixed grid: the text is *set solid* (*beta-gumi*, ベタ組み).

The `showhanmengrid` package draws, in the background of every page,

1. in cyan, one square of $1\text{zw} \times 1\text{zw}$ for each character position, and the frame of the whole grid;
2. in magenta, the *overflow*: the part of the text area that does not fit the grid;
3. in magenta, when the character pitch is not 1zw, a note in the top margin (section 5).

Both horizontal (yoko, 横組み) and vertical (tate, 縦組み) writing are supported.

Supported formats/engines pLaTeX and upLaTeX (with dvipdfmx), and LuaLaTeX with LuaTeX-jā. LaTeX2e 2025-06-01 or later is required. With any other engine (pdfLaTeX, XeLaTeX, or LuaLaTeX without LuaTeX-jā) the package issues a warning and does nothing. Trim marks that move the page are taken into account (section 6).

2 Usage

```
\usepackage{showhanmengrid}
```

Nothing else is needed: the grid is computed at `\begin{document}` from the layout parameters of the class, and the writing direction is taken from the document (`\tate` of pTeX, `direction=3` of LuaTeX-jā). Keys can be given as package options or with `\showhanmengridsetup` in the preamble:

```
\usepackage[color=blue!50, overflow=false]{showhanmengrid}
\showhanmengridsetup{rulewidth=0.2pt}
```

Key	Default	Description
color	cyan	color of the grid
overflowcolor	magenta	color of the overflow
rulewidth	0.1pt	width of the grid lines
frame	true	draw the frame of the whole grid
overflow	true	draw the overflow
chars	(computed)	characters per line
lines	(computed)	lines per page
ascent	0.88zw	yoko: height of the character face above the baseline
pitch	zw	pitch of the squares along the line: zw , natural or a length
note	true	draw the note on a character pitch other than 1 zw

3 How the grid is computed

Let 1zw and b be the full-width character size and `\baselineskip` of `\normalsize`, and p the pitch of the squares along the line (1zw by default; see section 5). In both yoko and tate, `\textwidth` is the length of a line and `\textheight` the extent over which the lines are stacked, so that

$$\begin{aligned}\text{chars} &= \lfloor (\text{\textwidth} - 1\text{zw})/p \rfloor + 1, \\ \text{lines} &= \lfloor (\text{\textheight} - \text{\topskip})/b \rfloor + 1,\end{aligned}$$

and the grid is $(\text{chars} - 1)p + 1\text{zw}$ long along the lines (that is, $\text{chars} \times 1\text{zw}$ for $p = 1\text{zw}$) and $(\text{lines} - 1)b + 1\text{zw}$ wide across the lines. A remainder of less than 0.01 pt is ignored (jlreq, for instance, adds 0.0001 pt to `\textwidth`). Both values and the overflow are written to the log.

Yoko The grid begins at the top of the character face of the first line: `\topskip` minus the ascent (0.88zw by default) below the top of the text area. The lines run from top to bottom.

Tate The baseline of the first line is `\topskip` left of the right edge of the text area, and the character face extends 0.5zw on each side of the baseline. The lines run from right to left.

Side margins The text area of the current page is placed with `\oddsidemargin` or `\evensidemargin` as the LaTeX kernel does it. The `\@outputpage` of pLaTeX has a branch that would swap the two margins in tate, but it tests `\iftdir` inside `\vbox{\yoko ...}` and so never applies; the package follows what is actually typeset.

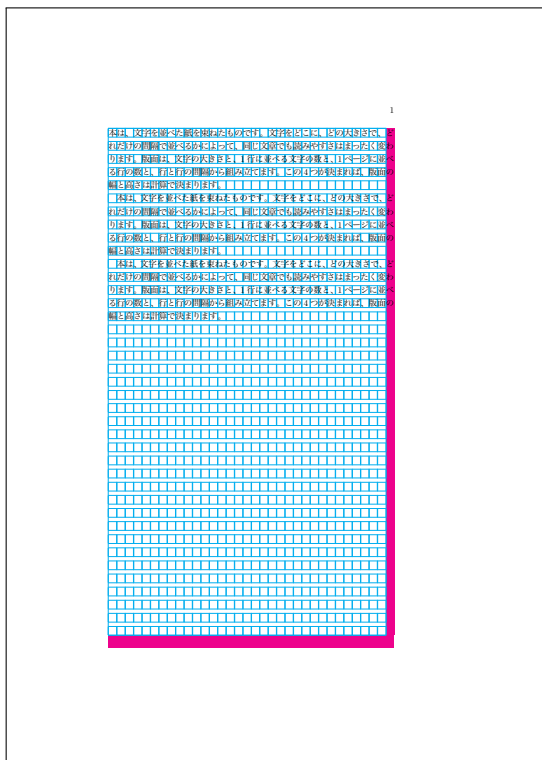
4 The overflow

Along the lines $\text{\textwidth} - (\text{chars} - 1)p - 1\text{zw}$, drawn after the end of the lines (right in yoko, bottom in tate). It is distributed into the inter-character spaces of every justified line: the characters drift away from their squares.

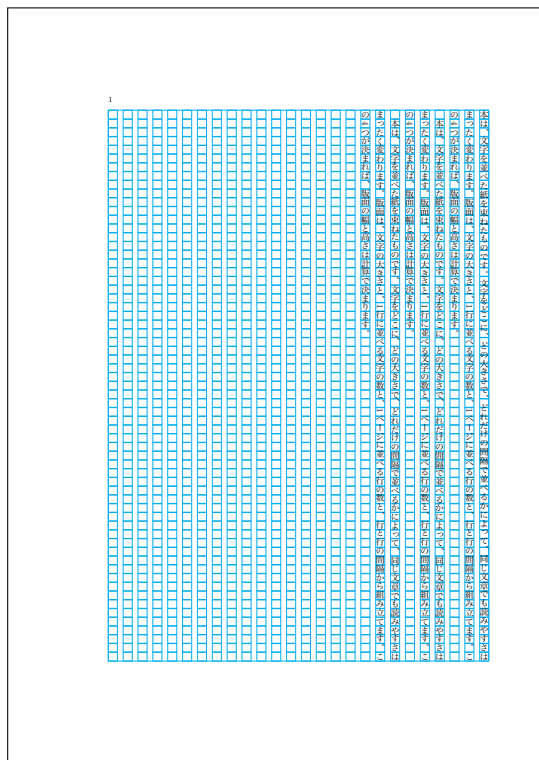
Across the lines $\text{\textheight} - \text{\topskip} - (\text{lines} - 1)b$, drawn after the last line (bottom in yoko, left in tate). The page is not filled by whole lines; with `\flushbottom` the remainder is distributed into vertical glue.

A text area with an overflow is what you get when only the four margins are given, as in word processors or with the `geometry` package. The default layout of the classic Japanese classes is another example: `jbook` (pLaTeX), `ujbook` (upLaTeX) and `ltjbook` (LuaTeX-ja) take their `\textwidth` from the standard book class, $\min(\text{\paperwidth} - 2\text{in}, 327\text{pt})$, a value chosen for Computer Modern; at 10 pt it is 33.98 zw, i.e. 9.47 pt longer than 33 characters. Their `\textheight`, $\lfloor (\text{\paperheight} - 3.5\text{in})/b \rfloor b + \text{\topskip}$, is then truncated to whole points by `\@settopoint`; since `\topskip` is not a whole number of

points, the truncation makes the last line no longer fit, and one line is lost (39 lines instead of 40 on A4, with an overflow of about 14pt). Only the `compatibility`-mode sizes of these classes use multiples of `\Cwd`. `jsbook` and `ljtsbook` set `\textwidth` to a multiple of 1zw, and `jlreq` sets the text solid by itself.



ujbook, default layout (A4): 33 chars $\times$ 39 lines with an overflow in both directions



jlreq, `tate`: 63 chars $\times$ 26 lines, set solid

5 Character pitch other than 1zw

Japanese text is not always set with the characters side by side. *Shibu-aki* (四分アキ, 0.25zw between characters) is common in *tate*, and some designs spread 34 characters over the width of 35. The space can be given by `\kanjiskip`, which is simple, or, more properly, by the glue between ideographs in the JFM (TFM).

The package measures the natural character pitch of `\normalsize` as

$$p_{\text{natural}} = (\text{width of 「漢漢漢」} - \text{width of 「漢」})/2,$$

which includes both. When it differs from 1zw, a one-line note is drawn in the top margin of every page, just above the head, and written to the log:

```
showhanmengrid: character pitch = 1.25zw, not 1zw
(\kanjiskip = 0zw, JFM +0.25zw); squares at pitch=zw
```

The parenthesis splits the extra space into the natural width of `\kanjiskip` and the part given by the JFM. (With LuaTeX-ja, `\kanjiskip` of `\maxdimen` means that the value of the JFM is used; the note then says “`\kanjiskip = JFM`”.) The key `note=false` turns the note off.

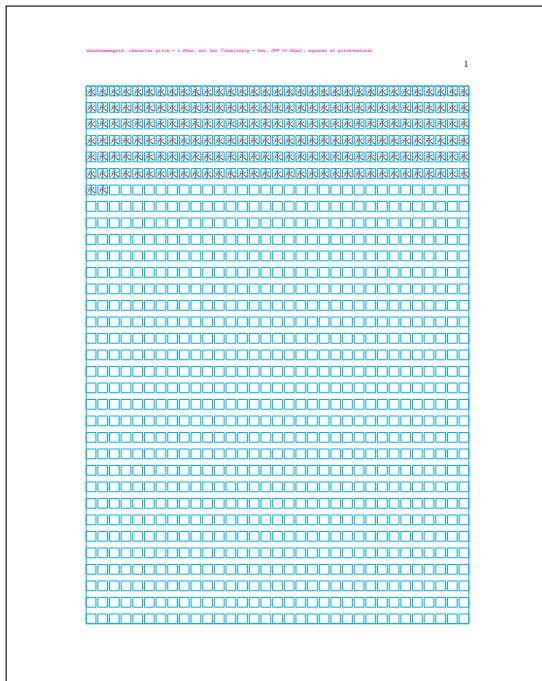
The key `pitch` sets the pitch p of the squares along the line:

pitch=zw (default) the squares of the hanmen design, 1zw each and side by side. The characters drift away from them.

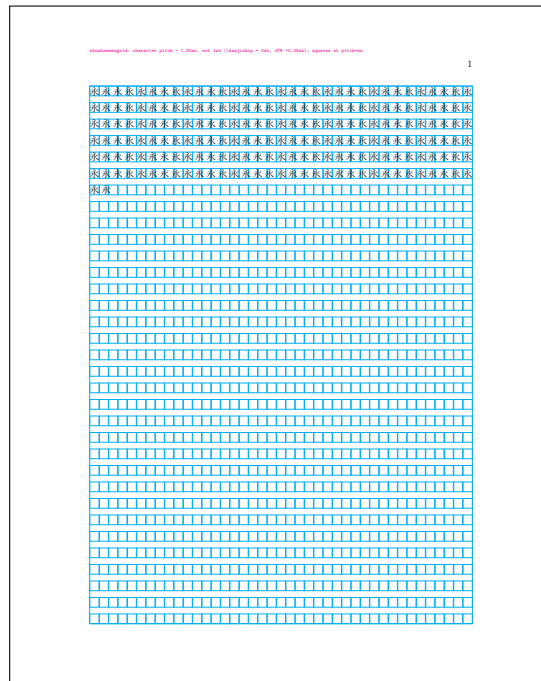
pitch=natural the squares (still 1zw) are placed at p_{natural} , so that every character sits in its square; the number of characters per line and the overflow follow from p .

pitch=<length> the pitch is given directly, e.g. **pitch=1.25zw**.

The line feed is not affected.



shibu-aki by the JFM, **pitch=natural**: 33 chars $\times$ 33 lines



the same text, **pitch=zw**: the grid of the design, 41 chars

6 Trim marks and `\mag`

The grid is drawn in the `shipout/background` picture, whose origin moves with `\hoffset` and `\voffset` as the text does. In addition,

- with pLaTeX, the `tombo` option of `jclasses` and `jsclasses`, and the classes and packages that rely on it (`\iftombow` is true; e.g. `gentombow` and `jlreq-trimmarks` on pLaTeX), move the page body by 1 in in both directions in `\@outputpage`; the grid is moved likewise;
- with LuaLaTeX, `gentombow` and `jlreq-trimmarks` move the page with `\hoffset` and `\voffset`, which needs no correction.

A `\mag` other than 1000 is not supported: the package writes a warning to the log and draws nothing.

7 Limitations

- Two-column layout is not supported: the grid spans `\textwidth`.
- `\mag` other than 1000 is not supported (see above).
- The character pitch is measured once, with 「漢」 of `\normalsize`; a JFM that gives different classes of characters different spaces is represented by the ideographs.

- The grid is computed once, at `\begin{document}`; changes of `\textwidth` or `\textheight` later in the document are ignored.

8 Tests

`make test` compiles every test file with pLaTeX, upLaTeX (+ dvipdfmx) and LuaLaTeX and checks the log. `SHG-EXPECT OK` means that the writing direction, the number of characters and lines, and the overflow are as expected; `SHG-POS OK` means that the first character of the page, measured with `\pdfsavepos` (pTeX) or `\savepos` (LuaTeX), sits in the top left square of the grid as drawn, on odd and even pages; some tests also check the end of the last square of a line. `SHG-NOTE OK` means that the note on the character pitch is as expected (or absent). With LuaTeX-jā in tate the position check is skipped (`SHG-POS SKIP`): LuaTeX-jā typesets tate text in a rotated box whose `\savepos` coordinates are not those of the page.

9 History

- v1.1 (2026/10/25)** Character pitch other than 1 zw: the note in the top margin and the keys `pitch` and `note`. Trim marks (`\iftombow` of pLaTeX, `gentombow`, `jlreq-trimmarks`) are taken into account. `\mag` other than 1000 disables the package with a warning. Fixed: with a non-zero `\hoffset` or `\voffset`, the grid was shifted by that amount twice.
- v1.0 (2026/10/25)** First public release. Generalized from the version written for the book on LaTeX page layout: it no longer needs `\hanmenwidth`/`\hanmenheight`, supports tate and pLaTeX, and draws the overflow.

10 License

This package is licensed under the conditions of the L^AT_EX Project Public License, either version 1.3c of this license or any later version.

Copyright © 2026 Munehiro Yamamoto