

plain-grid

Version 1.0, 05.05.2026

Macros for plain T_EX by Udo Wermuth

This package contains macros to typeset straight text with a fix distance between the lines, i.e., the text obeys a baseline grid. Moreover, the package contains macros to handle displayed equations, formulas with unusual height and depth in inline mathematics, vertical boxes, topinserts, and footnotes.

However, the baseline grid is only obeyed if the author avoids all plain T_EX commands that destroy this grid. Therefore, an author must not use vertical skips, i.e., `\vskip` as well as `\smallskip`, etc., or `\smallbreak`, etc., and macros that contain such skips like `\beginsection`.

For additional information and an example see my article in TUGboat 47:1 (2026), 119–134.

1. Activation

To activate the plain-grid macros load one of the following four files after plain T_EX via `\input`.

1. `plaingridT.tex` supports an author to typeset straight text on a baseline grid. It provides macros for endnotes too. It also contains a macro to draw horizontal rules.
2. `plaingridM.tex` helps an author to typeset text and mathematics, like displayed equations and theorems. Moreover, texts in vboxes are supported.
3. `plaingridI.tex` adds to `plaingridM.tex` the treatment of insertions and footnotes. It includes the feature of a *ragged bottom* if an author decides to violate the grid structure.
4. `plaingridB.tex` loads `plaingridI.tex` and contains macros for vboxes and vtops in horizontal mode, i.e., in paragraphs.

The above files usually implement new macros; they do not replace existing commands of T_EX or macros of plain. But there are a few exceptions: `\pagebody`, `\pagecontents`, `\vfootnote`, and the protected macro `\@foot`. The first is changed by `plaingridT.tex` and is therefore active in all files. The others are changed in `plaingridI.tex`.

As mentioned above, an author must use the new macros to stay on the baseline grid. Plain T_EX's macros must only be used in certain structures like math displays or vertical boxes created by the new macros. Then plain's macros are protected and the grid is at most violated inside the structure.

Moreover, an author should be aware that page breaks might occur at typographically bad places. For example, club and widow lines are possible and must be fixed by the author.

The new macros do not work correctly if some parameters of the set-up do not match with each other before they are loaded. This package requires that (i) the natural space of `\baselineskip` equals the sum of height and depth of `\strutbox` and (ii) `\maxdepth` and `\boxmaxdepth` are not smaller than the depth of `\strutbox`. Either check these requirements before any of the above files is loaded or input the file `plaingridX.tex` after one of the above files was input. It reports violations of the stated requirements. Nothing is fixed; do it yourself before the grid macros are loaded. If you continue although the values do not match, grid-aware typesetting is not active. When the extra file does not report any mismatch the file `plaingridX.tex` is no longer needed.

Of course, plain T_EX obeys the requirements.

2. Immediate changes

As soon as one of the four files is input, several T_EX parameters receive new values. Only two parameter changes are shown to the user on the terminal: (a) A change to the value of `\vsize` and (b) a change to the value of `\baselineskip`.

1. `plaingridT.tex` changes the following parameters and macros:
 - a) `\baselineskip` is coerced to a dimension, i.e., any stretchability and shrinkability of this glue parameter is removed. If this changes `\baselineskip`'s value the user is informed via a message.

- b) `\vsize` becomes an integer multiple of the `\baselineskip`. If this changes `\vsize`'s value the user is informed via a message.
 - c) `\normalbottom` is called.
 - d) `\topskip` is set to `\baselineskip`.
 - e) `\lineskip` is set to 0pt.
 - f) `\lineskiplimit` is set to 0pt.
 - g) `\parskip` is set to 0pt.
 - h) `\interlinepenalty` is set to 0.
 - i) `\clubpenalty` is set to 0.
 - j) `\widowpenalty` is set to 0.
 - k) `\brokenpenalty` is set to 0.
 - l) Plain T_EX's macro `\pagebody` is changed.
2. `plaingridM.tex` changes the above and the following parameters:
- a) `\abovedisplayskip` is set to 0pt.
 - b) `\belowdisplayskip` is set to 0pt.
 - c) `\abovedisplayshortskip` is set to 0pt.
 - d) `\belowdisplayshortskip` is set to 0pt.
 - e) `\predisplaypenalty` is set to 0.
 - f) `\postdisplaypenalty` is set to 0.
 - g) `\displaywidowpenalty` is set to 0.
 - h) `\everydisplay` is used by the package. The original token list is stored in a new token register: `\grideverydisplay`.
3. `plaingridI.tex` changes the above and the following parameters and macros:
- a) `\skip\topins` is set to 0pt.
 - b) `\skip\footins` is set to `\baselineskip`.
 - c) Plain's macro `\pagecontents` is changed.
 - d) Plain's macro `\vfootnote` is changed.
 - e) Plain's macro `\@foot` is changed.

The package declares one output stream and twelve registers. It uses several scratch registers.

3. Macros and conditionals in `plaingridT.tex`

The first set of macros defines vertical skips.

- 1. `\gridskiponline` leaves a single blank line.
- 2. `\gridbackonline` jumps back one line.
- 3. `\gridskipmax{n}` skips n lines. (Braces are only needed for $n > 9$.) If the current page has only m lines left with $m < n$ then m lines are skipped. The next page starts at its first line.
- 4. `\gridsnaptonextbaseline` jumps to the next baseline of the grid if it is not already on one. The used amount is always written to the log.

5. `\gridskiphalfaline` skips—as its name indicates— $0.5 \times \text{\baselineskip}$. Thus, it leaves the grid structure. It can be used to work in a two-column format with a shifted grid, for example, to surround some material with some space. As an example, see the macro `\gridproclaim`.

6. `\grideject` fills the current page and ejects it.

7. `\gridbye` ejects the current page, performs like `\bye` a `\supereject`, and ends the T_EX run.

The second set of macros handles endnotes.

8. `\griduseendnotes` must be called to initialize the handling of endnotes. It prepares a file in which the endnotes are stored. If it is called although such a file is currently opened it closes this file, opens a new one, and writes a message on the terminal that the first file has to be loaded manually.

9. `\gridenfilename`: The file that stores endnotes is named by default `endnotes.tex`. If more than one file is used a roman numeral is added, i.e., the second file is named `endnotesi.tex`, then it is `endnotesii.tex`, and so on. The stem for the file names is defined in the macro `\gridenfilename`.

10. `\gridendnote`, 11. `\gridnoteend` start and end the endnote in the text. The first line of the endnote must be placed on the same line as the `\gridendnote` (or use %). The `\gridnoteend` must be placed left-aligned in a line by itself.

The macro `\griduseendnotes` must be called before `\gridendnote` can be used. Otherwise the user gets an error message.

I assume `\gridendnote` is called after a punctuation mark. If an author has other ideas—or an em-dash follows—the line after `\gridnoteend` should start with `\unskip` as a space is inserted.

Endnotes are numbered. The numbering starts after every `\griduseendnotes` with 1.

12. `\gridenmark` is used to add a `\mark` to endnotes. The user can fill its replacement text, for example, (i) at the start of every chapter with the number of the chapter or (ii) in `\advancepageno` with the current page number. The `\mark` is placed in an `\immediate\write`.

Later in the section with the endnotes a running head can be defined that uses `\firstmark` and/or `\botmark` to help a reader to find the place where the endnote was used.

13. `\gridprintendnotes` closes the current file for endnotes and directly inputs it again. A user must provide a section title.

I think of several use cases. (i) All endnotes in a document are collected and output at the end. Call `\griduseendnotes` once at the beginning of the document and `\gridprintendnotes` at its end. (ii) The output of the endnotes should be split with subsection headers, for example, by chapter. Call `\griduseendnotes` at the start of every chapter and load the shown file names in sequence at the end of the document. The last file is input with `\gridprintendnotes`. (iii) The endnotes are output at the end of every chapter of a long document. Call `\griduseendnotes` at the beginning of each chapter and `\gridprintendnotes` at every chapter's end.

The third set of macros defines macros to enter section headers.

14. `\gridbeginsection` is build after plain T_EX's `\beginsection`. It takes an argument delimited by `\par` or an empty line. The argument should be at most one line and it is output in bold without indentation. The macro leaves two empty lines before and one empty line below this output.

If the current page has less than six lines free then the page is filled with empty lines and the section header appears at the top of the next page without white space above it but with an empty line below.

The argument of `\gridbeginsection` is shown on the terminal. One can change this by setting a conditional: `\gridhidesectiontitlestrue`.

15. `\gridstartsection` is like the previous macro but uses a different amount of white space around the section header. It starts at a new page if less than five lines are free on the current page. By default the macro skips `1.5\baselineskip` above and `0.5\baselineskip` below if it appears mid-page. Thus the section head is not printed on a grid line but in the middle of two. For the skips at the start of a page see the next macro.

16. `\gridkeepspace` configures how the previous macro `\gridstartsection` behaves at the start of a page.

The value 1 skips `0.5\baselineskip` above and below the title, the value 0 none above and one line below, and with 2 one gets 1.5 lines above and `0.5\baselineskip` below, i.e., the same amounts as

if it occurs mid-page. (The default is 1.) No case leaves an author with a shifted grid.

One sets the value with a definition, for example, `\def\gridkeepspace{1}`.

17. `\gridrule` has two arguments. The first is optional, the second must be surrounded by parentheses. It draws a grid-aware horizontal rule of height 0.4pt, depth 0pt, and width 20pt. Its height and width can be changed by the second argument. For example, `\gridrule(height 0.8pt width 40pt)` creates a horizontal rule twice as long and twice as thick as with the default setting. The depth cannot be changed.

The first argument raises the horizontal rule. For example, `\gridrule 3pt(height 0.8pt width 40pt)` draws the line of the previous example 3pt above the baseline.

And here is the last set of macros.

18. `\gridlog` determines the verbosity of logging. Some messages appear on the terminal (value 0), others in the log (value 1). Additional interesting or debugging information can be output with values 2 and 3. To change to level *n* use a definition: `\def\gridlog{ n }`. Default is level 1.

19. `\gridrestoreparams` restores the plain T_EX values of all changed parameters. Moreover, it restores the macro `\pagebody`.

Besides macros the file defines conditionals.

20. `\ifgridhidesectiontitles` (default setting: false) configures if section headers of the two macros `\gridbeginsection` and `\gridstartsection` are shown on the terminal. They are shown if the conditional is false.

21. `\ifgridnewpageOK` (default setting: false) determines if the message that is prepared in macros `\gridbeginsection` and `\gridstartsection` (as well as in `\gridproclaim` and `\gridbeginbox` of file `plainingridM.tex`) is shown or suppressed for the next execution. The setting true suppresses the message. For the message see section 7b), item number 4.

22. `\ifgridshowlinenos` (default setting: false) shows small line numbers in the vertical distance of `\baselineskip` on a page if it is found true at the time of the shipout.

4. Macros and registers in plaingridM.tex

This file adds the following macros to the above set. Note the macro `\gridrestoreparams` is extended to restore the plain T_EX values of the additionally changed parameters.

23. `\gridskip1/` adds a way to use a vertical skip between lines of a paragraph. One of 1, 2, 3, or 4 should follow the slash to skip 1, 1/2, 1/3, or 1/4 of the `\baselineskip`. For a value larger than 1 it is assumed that the macro is called two, three, or four times in a paragraph so that its last line on the current page is back on the grid. It is used together with `\gridinline` (see below) if the inline math does not fit into the leading.

Use `\gridsnaptonextbaseline` after a paragraph with `\gridskip` to fix small rounding errors. It depends on the `\baselineskip` and the argument if T_EX can compute a perfect result.

24. `\gridinline` has one optional argument and must be called in front of inline math, i.e., it should be followed by a `$`. It repeats the `\gridskip` except if there is an argument. It should be 0, 1, 2, 3, or 4 and this value is used as argument for the implicit `\gridskip1/`.

Note: Every large math construction in a line with `\gridinline` must be prefixed. The first performs the skip, the others must get `\gridinline0` to avoid multiple vertical skips. In a long formula use `\gridinline` only for the grid-violating constructions, so that T_EX can break the formula.

25. `\griddisplay` has two arguments separated by a colon and must be followed by `$$`.

Usually displayed equations between double-dollar signs are handled by the package automatically as `\everydisplay` is used. The display is put into a vbox in a way that the complete display fits into the grid. The size of this vbox is calculated to need the smallest number of lines. To overrule this decision the display can be prefixed by `\griddisplayi:j` to force *i* lines for the display and to add *j* \jot space at the top. But depending on the available size the display might start with less than this amount of white space.

26. `\gridproclaim` works like T_EX's `\proclaim`. It has two arguments separated by a period and must be followed by `\par` or an empty line. The first argument is set in bold with a period and a horizontal skip of `\enspace` after it. The second

argument is typeset as a paragraph with a slanted font. Above and below skips of `0.5\baselineskip` are added. That is, the material is placed on a shifted grid; it is also kept at the start of a page.

27. `\gridbeginbox` and **28.** `\gridendbox` build a pair that can be used to surround non grid-aware material. The output is a vbox with a height and depth that fits into the grid structure. Of course, a user should remember that vboxes cannot be broken between pages.

Besides macros the file declares one token register.

29. `\grideverydisplay` replaces `\everydisplay`. That is, use the new token register to define what needs to be executed at the start of every display.

5. Macros in plaingridI.tex

This file adds the following macros to the above set. Again, `\gridrestoreparams` is extended to cover all changed parameters of plain T_EX.

The first set of macros handles floating figures through insertions.

30. `\gridmidinsert` and **31.** `\gridtopinsert`, **32.** `\gridpageinsert` must always be followed by **33.** `\gridendinsert`. They work like the original plain T_EX macros except that they put their insertions into a vbox with height and depth that fits into the grid structure.

A `\gridmidinsert` tries to output its material at the place where it appears in the text. If that material does not fit anymore on the page it is transformed into a `\gridtopinsert` for a following page.

A `\gridtopinsert` is placed at the top of the current page if possible. Otherwise it is moved to the insertions for the next page. The third macro, `\gridpageinsert`, is like a `\gridtopinsert` except that it always needs a full page for itself. The sequence of these two insertion types is kept by T_EX.

34. `\grideotspace` is a macro that stores a factor for `\baselineskip`; default 0.5. That amount of space is at least left at the end of a topinsert. We get white space in the interval from `\grideotspace` times `\baselineskip` to $(1 + \text{\grideotspace})$ times `\baselineskip`. The amount is the sum of the

space added to make the topinsert fit into the grid and `\grideotspace`.

The second set of macros looks at footnotes.

35. `\gridfootnote` is followed by the footnote text in curly braces. As endnotes are numbered you should use symbols for footnotes if they are mixed. For example, take *, †, ‡, and § per page; double and triple them if there are more than four, i.e., use **, ††, etc.

Such a footnote is output at the bottom of the page in the normal text size. The first footnote is separated from the main text through a horizontal rule. Footnotes must not contain grid-violating material; use only plain text. They can be split between pages.

The macro requires the new `\pagecontents`.

The last feature fixes pages that aren't completely filled; for example, because there was only one and not two `\gridskip1/2` on the page.

36. `\ifgridragged` (default: false) determines if a ragged bottom is accepted or not.

The pages in this grid package have no stretch- or shrinkability. They must be broken at a multiple of `\baselineskip`. If for some reason a page is shorter then the ragged bottom feature allows a page break without generating an underfull vbox warning.

The macro requires the new `\pagecontents`.

6. Macros in `plaingridB.tex`

This file adds the following experimental macros to the above set.

37. `\gridboxann` and **38.** `\gridvbox` are used to add a vbox with several lines to a paragraph. Macro `\gridboxann` must be called in the line above the one with the vbox or in vertical mode for the first line. (The output becomes grid-unaware if the announcement for the first line is made in horizontal mode.) Macro `\gridvbox` gets as first argument a number; 1 signals that this vbox is the tallest in the current line.

The material for the vbox are short texts that are placed automatically in hboxes. Thus, a typical call is `\gridvbox0{first text}{second text}`. Don't use fillers, for example, spaces or `\relax` between the hboxes.

39. `\gridvtop`. This macro uses the same conventions as `\gridvbox` but creates a vtop not a vbox. It does not need an announcement.

7. Messages

The macros write several messages and the user can configure for some if they should be output at all. Messages on the terminal cannot be deactivated and one message is always written to the log file. The macro `\gridlog` determines the behavior.

a) Error message: "Missing `\griduseendnotes`." The help text for this message is: "Call the macro `\gridendnote` only between `\griduseendnotes` and `\gridprintendnotes` (and this one only once after the first)."

This message is shown (i) if the user enters `\gridendnote` without initializing the use of endnotes with `\griduseendnotes`. The initialization is used to prepare the file that stores the endnotes. As explained in section 3 this technique allows to implement several use cases for endnotes.

It's also shown (ii) if `\gridprintendnotes` is called without a `\griduseendnotes` since its last activation. The file with endnotes was closed and printed. A new file must be initialized by calling `\griduseendnotes`.

When the error message is shown in case (i) then the text of the endnote is output in the text.

b) There are nine terminal messages.

1. GRID: `\baselineskip` is now `<d>`

The `<d>` stands for a dimension in the unit pt. This message is shown if the `\baselineskip` has at the beginning a glue value with stretch- and/or shrinkability. Grid-aware typesetting removes any stretch and shrink components and keeps only the natural space.

Change the `\baselineskip` to a dimension before the macros are loaded and no message is shown.

2. GRID: `\vsize` changed; new value `<d>` (`<k>` lines)

Again, `<d>` is a dimension and `<k>` represents an integer. Here the `\vsize` was changed as it must be an exact multiple of `\baselineskip`.

Change the `\vsize` to an integer multiple of `\baselineskip` before the macros are loaded and no message is shown.

3. GRID: file with endnotes not yet output: `<t>`

`\griduseendnotes` was called twice without printing the existing endnotes by calling the macro

`\gridprintendnotes`. The macro opens a new file for the next endnotes and the author is informed that the file `<t>` needs to be manually loaded.

4. GRID: unexpected page ejection

This message might indicate an error. The user wants to output a multiline structure as it is created, for example, by `\gridbeginsection`, but it's too large for the current page. Thus, empty lines fill the current page and the structure is output on the next page.

To move the begin of a section might be okay. Then the message can be suppressed by setting `\gridnewpageOKtrue` in front of the section command. The message that is output for the movement of a displayed equation is not affected by the conditional and should not be tolerated.

Messages in the file `plaignridI.tex`.

5. GRID: page too short; fix

There is a footnote in the last available line on the current page. As there is no room for the text of the footnote the last line is moved to the next page leaving an underfull page. A line with more than one footnote might generate message no. 7.

In the log you find the following text: "The page is a line too short because a line with a footnote must be moved."

6. GRID: page ends ragged; fix

The user left the grid on the current page, for example, with `\gridskiphalfaline` or `\gridskip` but does not accept to end a page within shifted material, i.e., we have `\gridraggedfalse`.

7. GRID: ugly page; fix

Something went wrong and the page is underfull. It was not possible to identify the root cause as in the previous two messages.

The message is accompanied in the log by this text: "The page is not filled as expected. Maybe a text line with footnotes is moved to the next page, a skip violates the grid, or a forbidden command was used. Check the `\vbox` shown in TeX's warning."

Messages in the file `plaignridB.tex`.

8. GRID: vtop moved; check

A `\gridvtop` does not fit on the current page and must be moved to the beginning of the next. The current page is underfull.

9. GRID: vbox moved; check

A `\gridvbox` does not fit on the current page and must be moved to the next page. The current page is underfull.

c) There is one log message that is always output.

1. GRID line `<l>`: `\gridsnaptonextbaseline` uses `<d>`

Here `<l>` is an integer and `<d>` a dimension. `\gridsnaptonextbaseline` was used in line `<l>` of the currently processed file. The skip used by this macro is reported as `<d>` in the log.

d) There are four messages that are written to the log if `\gridlog`'s replacement text stores a number larger than 1. Two are written if `\gridlog` contains 2; otherwise all four are output. Often they do not reflect a user's input; different macros, sometimes internal, write them. They help to debug problems.

1. (level 2) GRID line `<l>`: available lines = `<k>`

This message is in file `plaignridT.tex`. A macro wants to check how many lines are available on the page. The value `<k>` represents the number of unused lines.

2. (level 2) GRID: page (`<d>`) [is bad | is okay | with footnote | has a ragged bottom | has a shift | with footnote has a ragged bottom | with footnote has a shift]

This message is in file `plaignridI.tex`. The message outputs the internal computation of the page's fill level, the dimension `<d>`, and the conclusion from this computation, i.e., one of the texts between the brackets where `|` means "or". A page that is labeled "is bad" or "has a shift" is an ugly page and produces one of the above described terminal messages. A "has a ragged bottom" indicates a page that ends in shifted material but the user set `\gridraggedtrue`, i.e., the user accepts such pages. The text "with footnote" indicates that the computation determines that there is a footnote on the page; it does not indicate a problem.

3. (level 3) GRID line `<l>`: box (`<d 1>+<d 2>`) [changed by (`<d 3>+<d 4>`)] output as (`<d 5>+<d 6>`) in `<k>` lines

A vbox is automatically created; it has height and depth of `<d 1>` and `<d 2>`. The call specifies that some white space of `<d 3>` and `<d 4>` are added to height and depth, respectively. The box is output with dimensions `<d 5>` and `<d 6>` in `<k>` lines to make it fit into the grid.

4. (level 3) GRID line `<l>`: user's height `<k>` lines or `<d 1>`; top plus `<d 2>`

The user decided to overrule the automatism in the calculation of a vbox with a displayed equation and this is reported in the log. Dimension `<d 2>` states the amount of white space that the user added at the top of the display.

The two level 3 messages belong to the file `plaignridM.tex`.