How to influence the position of float environments like figure and table in LaTeX?

Frank Mittelbach

Abstract

In 2012, a question "How to influence the float placement in LaTeX" was asked on TeX.stackexchange [3] and as there had been many earlier questions around this topic I decided to treat the topic in some depth and explain most of the mysteries that the underlying mechanism poses to people trying to use it successfully.

Once my answer appeared on the web, people asked to see this converted into an article and I foolishly replied "only if this answer ends up becoming a 'great' answer" (gets 100 votes). At the time of writing this article, the answer stands at 222 votes, so I had better make good on that promise.

Contents

1 Introduction

To answer this question one first has to understand the basic rules that govern LaTeX's standard placement of floats. Once these are understood, adjustments can be made, for example, by modifying float parameters, or by adding certain packages that modify or extend the basic functionality.

2 LaTeX floats terminology

2.1 Float classes

Each float in LaTeX belongs to a class. By default, LaTeX knows about two classes, viz., figure and table. Further classes can be added by a document class or by packages. The class a float belongs to influences certain aspects of the float positioning, such as its default placement specification (if not overridden on the float itself).

One important property of the float placement algorithm is that LaTeX never violates the order of placement within a class of floats. E.g., if you have figure 1, table 1, figure 2 in a document, then figure 1 will always be placed before figure 2. However, table 1 (belonging to a different float class) will be placed independently and hence can appear before, after, or between the figures.

2.2 Float areas

LaTeX knows about two float areas within a column where it can place floats: the top area and the bottom area of the column. In two-column layout, it also knows about a top area spanning the two columns. There is no bottom area for page-wide floats in two-column mode.

In addition, LaTeX can make float columns and float pages, i.e., columns or pages which contain only floats. Finally, LaTeX can place floats in-line into the text (but only if so directed on the individual float).

2.3 Float placement specifiers

To direct a float to be placed into one of these areas, a float placement specifier can be provided as an optional argument to the float. If no such optional argument is given then a default placement specifier is used (which depends on the float class as mentioned

above but usually allows the float to be placed in all areas if not subject to other restrictions).

A float placement specifier can consist of the following characters in *any* order:

- `!` indicates that some of the restrictions that normally apply should be ignored (discussed later)
- `h` indicates that the float is allowed to be placed in-line ("here")
- `t` indicates that the float is allowed to go into a top area
- `b` indicates that the float is allowed to go into a bottom area
- `p` indicates that the float is allowed to go on a float page or column area

The order in which these characters are put in the optional argument does *not* influence how the algorithm tries to place the float! The precise order is discussed in section 3.2. This is one of the common misunderstandings, for instance when people think that `bt` means that the bottom area should be tried first.

However, if a letter is not present then the corresponding area will not be tried at all.

2.4 Float algorithm parameters

There are about 20 parameters that influence the placement. Basically they define

- how many floats can go into a certain area,
- how big a float area can become,
- how much text there has to be on a page (in other words, how much the top and bottom float areas can occupy), and
- how much space will be inserted
  - between consecutive floats in an area and
  - between the float area and the text above or below it.

2.5 Float reference point

A point in the document that references the float (e.g., "see figure X") is called a "call-out" and the float body should be placed close to the (main) call-out, as its placement in the document affects the placement of the float in the output, because it determines when LaTeX sees the float for the first time. It's important to understand that if a float is placed in the middle of a paragraph, the reference point for the algorithm is the next line break, or page break, in the paragraph that follows the actual placement in the source.

For technical and practical reasons it is usually best to place all floats between paragraphs (i.e., after the paragraph with the call-out), even if that makes the call-out and reference point slightly disagree.

3 Basic behavioral rules of LaTeX's float mechanism

With this knowledge, we are now ready to delve into the algorithm's behavior.

First we have to understand that all of LaTeX's typesetting algorithms are designed to avoid any sort of backtracking. This means that LaTeX reads through the document source, formats what it finds and (more or less) immediately typesets it. The reasons for this design choice were to limit complexity (which is still quite high) and also to maintain reasonable speed (remember that this is from the early eighties).

For floats, this means that the algorithm is greedy, i.e., the moment it encounters a float it will immediately try to place it and, if it succeeds, it will never change its decision. This means that it may choose a solution that could be deemed inferior in light of data received later on.

For example, if a figure is allowed to go to the top or bottom area, LaTeX may decide to place this figure in the top area. If this figure is followed by two tables which are only allowed to go to the top, these tables may not fit anymore. A solution that could have worked in this case (but wasn't tried) would have been to place the figure in the bottom area and the two tables in the top area.

3.1 The basic sequence

So here is the basic sequence the algorithm runs through:

- If a float is encountered, LaTeX attempts to place it immediately according to its rules (detailed later);
- if this succeeds, the float is placed and that decision is never changed;
- if this does not succeed, then LaTeX places the float into a holding queue to be reconsidered when the next page is started (but not earlier).
- Once a page has finished, LaTeX examines this holding queue and tries to empty it as best as possible. For this it will first try to generate as many float pages as possible (in the hope of getting floats off the queue). Once this possibility is exhausted, it will next try to place the remaining floats into top and bottom areas. It looks at all the remaining floats and either places them or defers them to a later page (i.e., adding them once more to the holding queue).
- After that, it starts processing document material for this page. In the process, it may encounter further floats.
- If the end of the document has been reached or if a `\clearpage` is encountered, LaTeX starts a

new page, relaxes all restrictive float conditions, and outputs all floats in the holding queue by placing them on float page(s).

In two-column mode the same algorithm is used, except that it works on the level of columns, e.g., when a column has finished LaTeX will look at the holding queue and generate float columns, etc.

3.2 Detailed placement rules

Whenever LaTeX encounters a float environment in the source, it will first look at the holding queue to check if there is already a float of the same class in the queue. If that happens to be the case, no placement is allowed and the float immediately goes into the holding queue.

If not, LaTeX looks at the float placement specifier for this float, either the explicit one in the optional argument or the default one from the float class. The default per float class is set in the document class file (e.g., `article.cls`) and very often resolves to `tbp`, but this is not guaranteed.

- If the specifier contains a `!`, the algorithm will ignore any restrictions related either to the number of floats that can be put into an area or the maximum size an area can occupy. Otherwise the restrictions defined by the parameters apply.
- As a next step it will check if `h` has been specified.
- If so, it will try to place the float right where it was encountered. If this works, i.e., if there is enough space, then it will be placed and processing of that float ends.
- If not, it will look next for `t` and if that has been specified the algorithm will try to place the float in the top area. If there is no other restriction that prevents this, then the float is placed there and float processing stops.
- If not it will finally check if `b` is present and, if so, it will try to place the float into the bottom area (again obeying any restrictions that apply if `!` wasn't given).
- If that doesn't work either or is not permitted because the specifier wasn't given, the float is added to the holding queue.
- A `p` specifier (if present) is not used during the above process. It will only be looked at when the holding queue is being emptied at the next page or column boundary.

This ends the processing when encountering a float in the document.

3.3 Emptying the holding queue at the column or page boundary

After a column or page has been finished, LaTeX looks at the holding queue and attempts to empty it out as best as possible. For this it will first try to build float pages.[1]

Any floats participating in a float page (or column) must have a `p` as a float specifier in its float placement specification. If not, the float cannot go on a float page and, in addition, will also prevent any further deferred float of the same class from being placed onto the float page!

If the float can go there, it will be marked for inclusion on the float page, but the processor may still abort the attempt if the float page will not get filled "enough" (depending on the parameter settings for float pages). Only at the very end of the document, or when a `\clearpage` has been issued, are these restrictions lifted, and a float will then be placed on a float page even if it has no `p` and would be the only float on that page.

Creation of float pages continues until the algorithm has no further floats to place or when it fails to produce a float page due to parameter settings. In the latter case, all floats that have not been placed so far, are then considered for inclusion in the top and bottom areas of the next page (or column).

The process there is the same as the one described above, except that

- the `h` specifier no longer has any meaning (as we are, by now, far away from the original "here") and is therefore ignored,
- and the floats at this time are not coming from the source document but are taken one after the other from the holding queue.

Any float that couldn't be placed is then put back into the holding queue, so that when LaTeX is ready to look at further textual input from the document the holding queue may already contain floats. A consequence of this is that a float encountered in the document may immediately get deferred just because an earlier float of the same float class is already on hold.

3.4 Parameters influencing the placement

There are four counters that control how many floats can go into areas:

`totalnumber` (default 3) is the maximum number of floats on a text column; it is not used for float pages;

`topnumber` (default 2) is the maximum number of floats in the top area;

`bottomnumber` (default 1) is the maximum number of floats in the bottom area;

[1] In two-column mode LaTeX will build float columns (when finishing a column) and also attempt to generate float pages when finishing a page. In the remainder of the article "float page" will denote either depending on the context.

`dbltopnumber` (default 2) is the maximum number of full-width floats in two-column mode going above the text columns.

The size of the areas are controlled through parameters (to be changed with `\renewcommand`) that define the maximum (or minimum) size of the area, expressed as a fraction of the page height:

`\topfraction` (default 0.7) maximum size of the top area

`\bottomfraction` (default 0.3) maximum size of the bottom area

`\dbltopfraction` (default 0.7) maximum size of the top area for double-column floats

`\textfraction` (default 0.2) minimum size of the text area, i.e., the area that must not be occupied by floats

The space that separates floats within an area, as well as between float areas and text areas, is defined through the following parameters (all of which are rubber lengths, i.e., can contain some stretch or shrink components). Their defaults depend on the document font size and change when class options like 11pt or 12pt are used. We show only the 10pt defaults:

`\floatsep` (default 12pt plus 2pt minus 2pt) the separation between floats in top or bottom areas

`\dblfloatsep` (default 12pt plus 2pt minus 2pt) the separation between double-column floats on two-column pages

`\textfloatsep` (default 20pt plus 2pt minus 4pt) the separation between top or bottom float area and the text area

`\dbltextfloatsep` (default 20pt plus 2pt minus 4pt) the analog of `\textfloatsep` for two-column floats

For in-line floats (that have been placed "here") the separation to the surrounding text is controlled by

`\intextsep` (default 12pt plus 2pt minus 2pt)

In the case of float pages or float columns (i.e., a page or a column of a page containing only floats) parameters like `\topfraction` etc. do not apply. Instead the creation of them is controlled through

`\floatpagefraction` (default 0.5) minimum part of the page (or column) that needs to be occupied by floats to be allowed to form a float page (or column).

4 Consequences of the algorithm

4.1 A float may appear in the document earlier than its location in the source

The placement of the float environment in the source determines the earliest point where it can appear in the final document. It may move visually backward to some degree as it may be placed in the top area on the current page; see section 6.3 on how to change this. It can, however, not end up on an earlier page than the surrounding text due to the fact that LaTeX does no backtracking and the earlier pages have already been typeset.

Thus normally a float is placed in the source near its first call-out (i.e., text like "see figure 5") because this will ensure that the float appears either on the same page as this text or on a later page. However, in some situations you may want to place a float on the preceding page (if that page is still visible from the call-out). This is possible only by moving the float to an earlier position in the source.

4.2 Double-column floats are always deferred first

When LaTeX encounters a page-wide float environment (indicated by a `*` at the end of the environment name, e.g., `figure*`) in two-column mode, it immediately moves that float to the deferred queue. The reason for this behavior again lies in the "greedy" behavior of its algorithm: if LaTeX is currently assembling the second column of that page, the first column has already been assembled and stored away; recall that because LaTeX does not backtrack there is no way to fit the float on the current page. To keep the algorithm simple, it does the same even if working on the first column (where it could in theory do better even without backtracking).

Thus, in order to place such a float onto the current page, one has to manually move it to an earlier place in the source — before the start of the current page. If this is done, obviously any further change in the document could make this adjustment obsolete; hence, such adjustments are best done (if at all) only at the very last stage of document production — when all material has been written and the focus is on fine-tuning the visual appearance.

Also note that the base algorithm has a bug[2] in this area: it maintains two independent holding queues: one for single-column and one for double-column floats. As a result the float order is not necessarily preserved and floats may get typeset out of sequence. If this happens one either has to manually move the double-column float to an earlier (or later) place in the document or load the fixltx2e package that implements a correction for this issue.

[2] As this is the documented behavior in the LaTeX manual [1] it is perhaps more correctly called an undesired feature than a bug.

4.3 There is no bottom float area for double-column floats

This isn't so much a consequence of the algorithm but rather a fact about its implementation. For double-column floats the only possible placements offered are the top area or a float page. Thus if somebody adds an `h` or a `b` float placement specifier to such a float it simply gets ignored. As a special important case `{figure*}[b]` implies that this float will not get typeset at all until either a `\clearpage` is encountered or the end of the document is reached.

4.4 All float parameters (normally) restrict the placement possibilities

This may be obvious but it is worth repeating: any float parameter defines a restriction on LaTeX's ability to place the floats. How much of a restriction depends on the setting: there is always a way to set a parameter in such a way that it does not affect the placement at all. Unfortunately, in doing so one invites rather poor-looking placements.

By default LaTeX has settings that are fairly liberal. For example, for a float page to be accepted the float(s) must occupy at least half of the available page. Expressed differently, this means that such a page is allowed to be half empty (which is certainly not the best possible placement in most cases).

What often happens is that users try to improve such settings and then get surprised when suddenly all floats pile up at the end of the document. To stay with this example: if one changes the parameter `\floatpagefraction` to require, say, 0.8 of the float page, a float that occupies about 0.75 of the page will not be allowed to form a float page on its own. Thus, if there isn't another float that could be added and actually fits in the remaining space, the float will get deferred and with it all other floats of the same class. But, even worse, this specific float is too big to go into the next top area as well because there the default maximum permissible area is 0.7 (from `\topfraction`). As a result all your floats stay deferred until the next `\clearpage`.

For this reason it is best not to meddle with the parameters while writing a document or at least not to do so in a way that makes it more difficult for the algorithm to place a float close to its call-out. For proof-reading it is far more important to have a figure next to the place it is referenced then to avoid half-empty pages. Possibilities for fine-tuning an otherwise finished document are discussed below.

Another conclusion to draw here is that there are dependencies between some of the float parameters; it is important to take these dependencies into account when changing their values.

4.5 "Here" just means "here if it fits"

... and often it doesn't fit. This is somewhat surprising for many people, but the way the algorithm has been designed the `h` specifier is not an unconditional command. If an unconditional command is needed, extension packages such as the float package offer `H` as an alternative specifier that really means "here" (and starts a new page first if necessary).

4.6 Float specifiers do not define an order of preference

As mentioned above, the algorithm tries to place floats into available float areas in a well-defined order that is hard-wired into the algorithm: "here", "top", "bottom" and — on page boundaries — first "page" and only if that is no longer possible, "top" followed by "bottom" for the next page.

Thus specifying `[bt]` does not mean try bottom first and only then top. It simply means allow this float to go into top or bottom area (but not onto a float page) just like `[tb]` would.

4.7 Relation of floats and footnotes

This is not exactly a consequence of the algorithm but one of its implementation: Whenever LaTeX tries to decide on a placement for a float (or a `\marginpar`!) it has to trigger the output routine to do this. And as part of this process all footnotes on the page are removed from their current place in the galley and are collected together in the `\footins` box as part of TeX's preparation for page production.

But after placing the float (or deferring it) LaTeX then returns the page material to the galley, and because of TeX's output routine behavior the galley has now changed: all the footnotes have been taken out from their original places. So LaTeX has to put the footnotes back, but it can only place them in a single place (not knowing the origin anymore). What it does is reinsert the footnotes (the footnote text to be precise) at the end of the galley. There are some good reasons for doing this, one of which is that LaTeX expects that all of the returned material still fits on the current page.

However, if for some reason a page break is finally taken at an earlier point then the footnotes will show up on the wrong page or column. This is a fairly unlikely scenario and LaTeX works hard at making it a near-impossibility, but if it happens check if there is a float near the chosen page break and either move the float or guide the algorithm by using explicit page breaks. An example of this behavior can be found in another question on TeX.stackexchange [4]. In fact the particular case discussed in the question is worth highlighting: Do *not* place a float directly

after a heading, unless it is a heading that always starts a page. The reason is that headings normally form very large objects (as a heading prevents a page break directly after it). However placing a float in the middle of this means that the output routine gets triggered before LaTeX makes its decision where to break and any footnotes get moved into the wrong place.

5 Documentation of the algorithm

As requested, here is some information on existing documentation. The algorithm and its implementation are documented in the file `ltoutput.dtx` as part of the LaTeX kernel source. This can be typeset standalone or as part of the whole kernel (i.e., by typesetting `source2e.tex` — ignore the checksum error if it is still there,[3] sorry).

This documentation is an interesting historical artifact. Parts of it show semi-formatted pseudo-code which dates back to LaTeX 2.09; in other words it is from the original documentation by Leslie Lamport. The actual code is documented using doc style and in parts is more or less properly documented (from scratch) and dates back to 1994 or thereabouts when Chris Rowley and myself adjusted and extended the original algorithm for LaTeX 2_ε (the current version). It also fairly openly documents the various issues with the algorithm and/or its implementation — in many cases we didn't dare to alter it because of the many dependencies and, of course, because of the danger to screw up too many existing documents that implicitly rely on the current behavior for good or ill.[4] Near the end you'll find a list of comments compiled on the algorithm back then, but there are also comments, questions, and tasks (?:-) sprinkled throughout the documentation of the code.

One interesting aspect of this file (that I forgot all about) is that it contains all the code necessary to trace the behavior of the algorithm in real life. It is fairly raw and detailed output and probably for that reason I didn't make this publicly available back then. But even in its current form it does give some interesting insight into the behavior of the algorithm and how certain decisions come about.

Thus while writing this article I had second thoughts and now the most recent distribution of LaTeX (May 2014)[5] offers the package fltrace that you can load to trace some strange float placement decisions, or simply to understand the algorithm a bit better. It offers the commands `\tracefloats` and `\tracefloatsoff` to start or stop tracing the algorithm and `\tracefloatvals` to display the current values of various float parameters that are discussed in this article.

As the package is identical to the kernel code with tracing added, it may or may not work if you load any other package that manipulates that part of the kernel code. In such a case your best bet is to load fltrace first.

6 How to address specific issues

In the final section we discuss a few strategies to circumvent or resolve common issues. It is by no means comprehensive and you may find further information in other publications, e.g., *The LaTeX Companion* [2] that devotes a whole chapter to the topic of floats.

6.1 Ensure that floats appear "here"

Sometimes it is necessary to ensure that floats appear in-line at certain points in the document text even if that results in some partially empty pages. As discussed above the `h` specifier doesn't provide this functionality but there are extensions that do, such as the float package which offers an `H` specifier for this purpose.

An alternative is the `\captionof` command from the caption package that generates a normal float caption (including its entry in the list of figures or tables, etc.) but without the need for a surrounding float environment.

6.2 Provide a bottom float area for two-column floats

As discussed above, the standard algorithm doesn't support double-column floats at the bottom of pages. This missing functionality is added, except for the first page[6], if you load the stfloats package.

6.3 Ensure that floats are always placed after their call-out

By default the LaTeX float algorithm allows for floats to move before their call-out as long as float and call-out are on the same page; more precisely, it allows floats to appear in the top area of the column in which the float has been encountered.

[3] But this also means you are running an older release of LaTeX.

[4] This is, for example, the reason that the correction of the issue discussed in section 4.2 was placed into the fixltx2e package and not made part of the kernel algorithm.

[5] If you have an earlier version of LaTeX installed, you can still extract this code yourself, by writing a short installation file `fltrace.ins` with the following content:

```
\input docstrip
\generateFile{fltrace.sty}{t}{%
  \from{ltoutput.dtx}{fltrace,trace}}
\endbatchfile
```

and run this through LaTeX.

[6] See [5] in this issue to manually lift even this restriction.

This practice offers a better chance that the float is visible from the call-out position and doesn't end up on a later page. For some journals, however, this is too liberal and they require that floats are strictly placed after their call-out, i.e., that in the call-out column only the bottom area forms a valid placement option. To accommodate this requirement, this strategy is implemented by the flafter package.

This may work well if your document has only a few floats. For documents with lots of floats, placement obviously becomes much more difficult, and you may find that all your floats appear together at the end of the document or chapter, or you may receive a "Too many unprocessed floats" error.

6.4 Prevent floats on certain pages

Sometimes it is helpful to prevent floats from appearing on a certain page, for example, to prevent a float in a new section from moving into the top area on the current page without disallowing a placement in the top area of a later page. For this type of fine-tuning LaTeX offers the command `\suppressfloats[`*placement*`]`. The optional argument can be either `t` or `b` and prevents any further placement into the respective area(s) on the current page. Without an argument, all remaining floats on the current page are deferred.

6.5 Implement float barriers

Standard LaTeX already implements a float barrier called `\clearpage`. Floats on either side will never appear on the other. It works by outputting all deferred floats, if necessary by generating float pages, and then starting a new page. While this is suitable to keep floats within one chapter (as chapters typically start on a new page) there are cases where one would wish for a less intrusive barrier, i.e., one that works without forcing a new page or is partially porous.

This functionality is offered by the placeins package, which implements a `\FloatBarrier` command that doesn't introduce a page break. Through package options you can alter the behavior to allow for floats to migrate from one side to the other as long as they still appear on the same page.

6.6 Overwrite placement restrictions

If a given float is (slightly) too large to fit into a certain area or if an area already contains the maximum number of floats but you nevertheless want to force the current float into this place then adding `!` to the optional argument of the float is a good choice. It results in ignoring all restrictions implemented through parameters for this particular float, so that it will always be placed unless there are already deferred floats with the same float class or the allowed areas get bigger than the available space when adding the float.

As the order of attempts is still the same (first top then bottom), you may have to use `[!b]` to force a float into the bottom area as `[!tb]` would normally already succeed in placing it into the top area. The downside is of course that if the float doesn't fit, it will only appear in the bottom area of a following page. Thus any later text change may create havoc on your placement decisions.

6.7 Final tuning advice

There are many ways to fine-tune the behavior of the float placement algorithm; most of them have been discussed in this article. However, there is one more "tuning" possibility and in fact the biggest of all: changes in your document text.

Therefore, as final advice: do not start manipulating parameters or change placement specifiers or move floats within your document until after you have fully written your text and your document is close to completion. It is a waste of effort and it may even result in inferior placements as your initially provided restrictions may no longer be adequate after a text change.

References

[1] Leslie Lamport. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1994. Reprinted with corrections in 1996.

[2] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX Companion*. Tools and Techniques for Computer Typesetting. Addison-Wesley, Reading, MA, USA, second edition, 2004. Also available as an eBook, see `http://www.latex-project.org/site-news.html#2013-11-02`.

[3] Marco Daniel. How to influence the position of float environments like figure or table in LaTeX?, 2012. `http://tex.stackexchange.com/q/39020`.

[4] Martin Hermann. "thanks" note (footnote) placed below right column even though there is enough space on the left, 2012. `http://tex.stackexchange.com/q/43294`.

[5] Barbara Beeton. Placing a full-width insert at the bottom of two columns. *TUGboat*, 35(3):255–255, 2014. `http://tug.org/TUGboat/35-3/tb111beet-banner.pdf`.

◇ Frank Mittelbach
LaTeX3 Project
`http://www.latex-project.org`