The `parskip` package[*]

Frank Mittelbach

June 15, 2020

Abstract

The parskip package helps in implementing paragraph layouts where the paragraphs are separated by a vertical space instead of (or in addition to) indenting them.

The package can be used with any document class at any size. By default it produces the following paragraph layout: Zero `\parindent` and non-zero `\parskip`. The stretchable glue in `\parskip` helps LaTeX in finding the best place for page breaks.

1 Introduction

Many LaTeX constructs are internally built by using the paragraph mechanism even if technically there aren't text paragraphs. In most such cases the LaTeX code handles indentation and suppressed it if necessary. But unfortunately this is normally not done for `\parskip` (as that is zero in the default layouts) and thus changing it will result in vertical spaces in unexpected places.

This package attempts to fix the spacing in table of contents structures, list environments, and around display headings that would get screwed up by a positive `\parskip` value.

It is, however, is no more than quick fix; the 'proper' way to achieve effects as far-reaching as this is to create a new class.

1.1 History

This file was originally developed by Hubert Partl in 1989 (i.e., for LaTeX 2.09) to provide a somewhat crude solution to an existing problem in case no proper document class (back then called document style) support was available.

About ten years later Robin Fairbairns picked up the orphaned package and his version was then the one available for LaTeX 2_ε during the next 15^+ years.

Finally, while working on the next edition of the LaTeX Companion the current author did a reimplementation, that added support for TOC data and heading structures. Also a few additional key/value options were added to make

---

[*] This is a reimplementation of a package originally written by Hubert Partl in 1989 and later maintained by Robin Fairbairns.

the package more useful. It still is and will remain an inferior choice compared to a properly designed document class. But it offers a starting point if nothing is around.

2 The user interface

The parskip package doesn't offer any document user commands and just needs loading with `\usepackage`.

2.1 Options to customize the package

All of the package options are implemented as key/value options.

`skip` With the package option `skip` it is possible to explicitly specify the vertical space between paragraphs. If the option is not given (or given without a value) then `.5\baselineskip` plus `2pt` of stretch is assumed.

`tocskip` By default the `\parskip` is zero within `\tableofcontents` and similar lists, regardless of its value elsewhere. With the option `tocskip` it can be given a different value. If used without an explicit value you get the same `\parskip` as elsewhere within these lists.

`indent` With the package option `indent` it is possible to explicitly set the paragraph indentation. Using this option without a value keeps the document class indentation unchanged, if it is specified with a value then that value is used. If the package is loaded without this option the indentation is set to zero.

`parfill` With package option `parfill`, the package also adjusts `\parfillskip` to impose a minimum space at the end of the last line of a paragraph. If specified without a value then `30pt` are assumed, if a value is given that forms the minimum.

3 Differences to the original package

If the package is used without any options or just with the option `parfill` it behaves like the earlier version, except that now the spacing around headings is also adjusted (not adding extra `\parskip`). If this is not desirable when processing an old document it can be avoided by explicitly requesting version `v1` as follows:

```
\usepackage{parskip}[=v1]
```

Of course, the new options, etc. are then also not available.

4 Sources, bugs and issues

The official production version is available from CTAN. The latest (development) sources are maintained at GitHub at:

`https://github.com/FrankMittelbach/fmitex/tree/parskip/parskip`

In case of problems with the package you can report them at

`https://github.com/FrankMittelbach/fmitex/issues`

Please provide a minimal test example that can be run and doesn't use packages not in a standard LaTeX distribution (and as little as possible to show the issue).

5 The Implementation

```
⟨*package⟩
```

5.1 The main implementation part

```
\NeedsTeXFormat{LaTeX2e}[2018-04-01]

\DeclareRelease       {v1}{2001-04-09}{parskip-2001-04-09.sty}
\DeclareCurrentRelease{v2}{2018-08-24}
```

```
\ProvidesPackage{parskip}[2020-06-15 v2.0f non-zero parskip adjustments]
```

5.1.1 Option handling

Here we define all option keys for use as package options:

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{family=parskip,prefix=parskip@}
```

The key `indent` defines the amount of indentation for each paragraph. If not given the indentation will be zero (default) and if given without a value then the outer value from the document class will get used, otherwise the given value is used.

```
\DeclareStringOption[0pt]{indent}[\parindent]
```

The key `parfill` defines a minimum amount of white space that should be left in the last line. By default the last line can get completely fill up. If given without a value the default (as before) is to require a minimum of `30pt`, otherwise the given value is used.

```
\DeclareStringOption[0pt]{parfill}[30pt]
```

The key `skip` defines the vertical separation between paragraphs. If not given the default (as before) is to use half a `\baselineskip` plus a stretch of `2pt` to add some flexibility. If given, one need to provide an explicit value which is then used as a separation (and it needs to contain any extra stretch if that is wanted, i.e., there is no extra stretch added in this case).

```
\DeclareStringOption{skip}[]
```

The key `tocskip` defines the vertical separation inside the lists `\tableofcontents`, `\listoffigures` and `\listoftables`. By default there is no extra separatation (i.e., `0pt`). If specified without a value the standard `\parskip` is used, otherwise the given value.

```
\DeclareStringOption[0pt]{tocskip}[\parskip]
```

Execute any package options:

```
\ProcessKeyvalOptions*
```

So now we can evaluate the given options and adjust the parameter settings:

```
\ifx\parskip@skip\@empty
```

If no `skip` was given (or it was empty) set `\parskip` to half of `.5\baselineskip` plus `2pt` stretch. Stretch or shrink inside `\baselineskip` is ignored in this case.

```
  \parskip=.5\baselineskip plus 2pt\relax
\else
```

Otherwise set it to the specified value:

```
  \setlength\parskip\parskip@skip
\fi
```

Setting `\parfillskip` was suggested by Donald Arseneau at some point on comp.text.tex:

```
\setlength\parfillskip\parskip@parfill
\advance\parfillskip 0pt plus 1fil\relax
```

`\parindent` gets whatever was specified. If the key was given without an option this will essentially reassign the now "current" value.

```
\setlength\parindent\parskip@indent
```

5.2 Handling document elements

Setting up a non-zero `\parskip` has some side-effects in document elements such as lists or headings etc. Here we try to keep these side-effects somewhat under control.

We make use of the etoolbox package to do patching.

```
\RequirePackage{etoolbox}
```

5.2.1 Lists

To accompany this, the vertical spacing in the list environments is changed to use the same as `\parskip` in all relevant places (for `\normalsize` only), i.e.

```
\parsep = \parskip
\itemsep = \z@ % add nothing to \parskip between items
\topsep = \z@ % add nothing to \parskip before first item
```

However, if the user explicitly asked for a zero parskip (via the `skip` option) we shouldn't do this but rather keep the default list settings, so we better check for this.

```
\ifdim \parskip > 0pt
```

```
  \def\@listI{\leftmargin\leftmargini
     \topsep\z@ \parsep\parskip \itemsep\z@}
  \let\@listi\@listI
  \@listi

  \def\@listii{\leftmargin\leftmarginii
     \labelwidth\leftmarginii\advance\labelwidth-\labelsep
     \topsep\z@ \parsep\parskip \itemsep\z@}

  \def\@listiii{\leftmargin\leftmarginiii
      \labelwidth\leftmarginiii\advance\labelwidth-\labelsep
      \topsep\z@ \parsep\parskip \itemsep\z@}
%
% and finally ...
%   \partopsep = \z@ % don't even add anything before first item (beyond
%                    % \parskip) even if the list is preceded by a blank line
  \partopsep=\z@
\fi
```

5.2.2 TOCs and similar lists

Within a table of contents or a list of figures we don't want any additional vertical spacing just because the individual lines in such a list are implemented as one-line paragraphs. So we locally set the `\parskip` to zero by default. Should be really something that is done already in LaTeX.

```
\patchcmd\@starttoc
    {\begingroup \makeatletter}
    {\begingroup \makeatletter
```

Just setting `\parskip` to zero as it was done in the original version of the package, does not always work. If the list starts out with an ordinary paragraph (and not with `\addvspace` as it usually does) we will get a zero `\parskip` but the heading above assumes we get the normal `\parskip` and has therefore removed that amount from its own vertical skip. As long as the parskip value is not too large people didn't notice that heading and list moved closer to each other but if you use, say, `[skip=20pt]` you will even see an overlap.

We therefore do the following: we look at the last skip, undo it and then issue a skip that is equal to `\parskip` + `\lastskip`. This way the skip seen by any following code has the right value which is important for `\addvspace` calulations. Only then we locally set `\parskip` to zero or rather to `\parskip@tocskip`, the parameter that the user can set through an option.

```
      \skip@\lastskip
      \advance\skip@\parskip
      \vskip-\lastskip
      \vskip\skip@
      \parskip\parskip@tocskip}
    {}{\typeout{Couldn't patch \string\@starttoc}}
```

5.2.3 Standard headings

For the same reason we don't want to see an additional `\parskip` being added before and after a display heading, so we subtract its value (in two places):

```
\patchcmd\@startsection
    {\addvspace\@tempskipa}
    {\advance\@tempskipa-\parskip\addvspace\@tempskipa}
    {}{\typeout{Couldn't patch \string\@startsection}}
```

```
\patchcmd\@xsect
    {\vskip\@tempskipa}
    {\advance\@tempskipa-\parskip\vskip\@tempskipa}
    {}{\typeout{Couldn't patch \string\@xsect}}
```

5.2.4 titlesec headings

If titlesec is used then headings are built using different commands and we have to cancel the `\parskip` there. The principle is the same. Of course, the patching should only happen if that package really got loaded, so we defer it to the start of the document and test for it:

```
\AtBeginDocument{%
\ifx\ttl@straight@ii\@undefined\else  % titlesec got loaded
\patchcmd\ttl@straight@ii
    {\addvspace{\@tempskipa}}%
    {\advance\@tempskipa-\parskip \addvspace\@tempskipa}%
    {}{\typeout{Couldn't patch \string\ttl@straight@ii}}%
\patchcmd\ttl@straight@ii
    {\vspace{\@tempskipb}}%
    {\advance\@tempskipb-\parskip \vspace\@tempskipb}%
    {}{\typeout{Couldn't patch \string\ttl@straight@ii}}%
\patchcmd\ttl@part@ii
    {\vspace*{\@tempskipa}}%
    {\advance\@tempskipa-\parskip \vspace*\@tempskipa}%
    {}{\typeout{Couldn't patch \string\ttl@part@ii}}%
\patchcmd\ttl@part@ii
    {\vspace{\@tempskipb}}%
    {\advance\@tempskipb-\parskip \vspace\@tempskipb}%
    {}{\typeout{Couldn't patch \string\ttl@part@ii}}%
\patchcmd\ttl@page@ii
    {\vspace*{\@tempskipa}}%
    {\advance\@tempskipa-\parskip \vspace*\@tempskipa}%
    {}{\typeout{Couldn't patch \string\ttl@page@ii}}%
\patchcmd\ttl@page@ii
    {\vspace{\@tempskipb}}%
    {\advance\@tempskipb-\parskip \vspace\@tempskipb}%
    {}{\typeout{Couldn't patch \string\ttl@page@ii}}%
\fi}
```

5.2.5 **amsthm theorems**

The amsthm package is one of the few packages that make an explicit correction for `\parskip` which isn't any longer adequate if this parskip package is loaded. We therefore remove that setting from the package if it was loaded.

```
\AtBeginDocument{%
\ifx\deferred@thm@head\@undefined\else  % amsthm got loaded
\patchcmd\deferred@thm@head
  {\addvspace{-\parskip}}{}%
  {}{\typeout{Couldn’t patch \string\deferred@thm@head!}}%
\fi}
```

5.3 Closing shop

```
⟨*package⟩
```