

The `parskip` package*

Frank Mittelbach,virhuiai 翻译

December 16, 2025

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 L^AT_EX in finding the best place for page breaks.

摘要

`parskip` 宏包用于实现段落布局，其中段落之间通过垂直间距分隔，而不是（或除了）缩进。

该宏包可用于任何文档类和任何字号。默认情况下，它产生以下段落布局：零 `\parindent` 和非零 `\parskip`。`\parskip` 中的可拉伸胶水有助于 L^AT_EX 找到最佳的分页位置。

1 Introduction

1 引言

Many L^AT_EX constructs are internally built by using the paragraph mechanism even if technically there aren't text paragraphs. In most such cases the L^AT_EX 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.

许多 L^AT_EX 结构在内部使用段落机制构建，即使从技术上讲它们不是文本段落。在大多数情况下，L^AT_EX 代码会处理缩进，并在必要时抑制它。但不幸的是，对于 `\parskip` 通常不会这样做（因为在默认布局中它为零），因此更改它会导致在意外位置出现垂直间距。

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.

此宏包尝试修复目录结构、列表环境以及标题周围的间距，这些间距会因正的 `\parskip` 值而混乱。

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.

然而，这只不过是一个快速修复；要实现如此深远的效果，‘正确’的方法是创建一个新的文档类。

1.1 History

1.1 历史

This file was originally developed by Hubert Partl in 1989 (i.e., for L^AT_EX 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.

此文件最初由 Hubert Partl 于 1989 年开发（即用于 L^AT_EX 2.09），用于在没有适当的文档类（当时称为文档样式）支持的情况下提供一个略显粗糙的解决方案。

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

大约十年后，Robin Fairbairns 接手了这个被遗弃的宏包，他的版本在接下来的 15+ 年里一直是 L^AT_EX 2_E 可用的版本。

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

Finally, while working on the next edition of the L^AT_EX 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 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.

最后，在编写《L^AT_EX 伴侣》下一版时，当前作者进行了重新实现，添加了对目录数据和标题结构的支持。还添加了一些额外的键/值选项，使宏包更加有用。与精心设计的文档类相比，它仍然是并将继续是一个次优选择。但如果没有任何其他选择，它提供了一个起点。

2 The user interface

2 用户界面

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

`parskip` 宏包不提供任何文档用户命令，只需使用 `\usepackage` 加载即可。

2.1 Options to customize the package

2.1 自定义宏包的选项

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

所有宏包选项都实现为键/值选项。

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.

使用宏包选项 `skip` 可以显式指定段落之间的垂直间距。如果未给出该选项（或给出时没有值），则默认使用 `.5\baselineskip` 加上 `2pt` 的拉伸量。

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.

默认情况下，`\tableofcontents` 和类似列表中的 `\parskip` 为零，无论其在其他地方的值如何。使用选项 `tocskip` 可以为其指定不同的值。如果使用时没有显式值，则在这些列表中获得与其他地方相同的 `\parskip`。

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.

使用宏包选项 `indent` 可以显式设置段落缩进。使用此选项时不带值可保持文档类缩进不变，如果指定了值则使用该值。如果不加载此选项，缩进将被设置为零。

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.

使用宏包选项 `parfill`，宏包还会调整 `\parfillskip`，以在段落最后一行末尾施加最小间距。如果指定时没有值，则默认使用 `30pt`，如果给出了值，则该值作为最小值。

3 Differences to the original package

3 与原始宏包的区别

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:

如果宏包在不使用任何选项或仅使用 `parfill` 选项的情况下使用，其行为与早期版本类似，除了现在也会调整标题周围的间距（不添加额外的 `\parskip`）。如果在处理旧文档时不希望这样，可以通过显式请求版本 v1 来避免，如下所示：

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

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

当然，这样一些新的选项等也将不可用。

4 Sources, bugs and issues

4 源代码、错误和问题

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

官方生产版本可从 CTAN 获取。最新（开发）源代码在 GitHub 上维护，地址为：

<https://github.com/FrankMittelbach/fmitex-parskip>

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

如果宏包出现问题，您可以在以下地址报告：

<https://github.com/FrankMittelbach/fmitex-parskip/issues>

Please provide a minimal test example that can be run and doesn't use packages not in a standard L^AT_EX distribution (and only those that are needed to show the issue).

请提供一个可以运行的最小测试示例，不要使用标准 L^AT_EX 发行版中没有的宏包（并且只使用显示问题所需的宏包）。

5 The Implementation

5 实现

1 (*package)

5.1 The main implementation part

5.1 主要实现部分

```
2 \NeedsTeXFormat{LaTeX2e}[2018-04-01]
3
4 \DeclareRelease      {v1}{2001-04-09}{parskip-2001-04-09.sty}
5 \DeclareCurrentRelease{v2}{2018-08-24}
6 \ProvidesPackage{parskip}[2021-03-14 v2.0h non-zero parskip adjustments]
```

5.1.1 Option handling

5.1.1 选项处理

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

在这里，我们定义所有用作宏包选项的选项键：

```
7 \RequirePackage{kvoptions}
8 \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.

键 `indent` 定义每个段落的缩进量。如果未给出，缩进将为零（默认值）；如果给出时没有值，则使用文档类的外部值；否则使用给定值。

```
9 \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.

键 `parfill` 定义了最后一行应保留的最小空白量。默认情况下，最后一行可以完全填满。如果给出时没有值，默认（与之前一样）是要求最小 30pt，否则使用给定值。

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

键 `skip` 定义段落之间的垂直分隔。如果未给出，默认（与之前一样）是使用半个 `\baselineskip` 加上 2pt 的拉伸量以增加一些灵活性。如果给出，需要提供一个显式值，该值随后用作分隔（如果需要，它需要包含任何额外的拉伸，即在此情况下不添加额外的拉伸）。

键 `skip` 定义段落之间的垂直分隔。如果未给出，默认（与之前一样）是使用半个 `\baselineskip` 加上 2pt 的拉伸量以增加一些灵活性。如果给出，需要提供一个显式值，该值随后用作分隔（如果需要，它需要包含任何额外的拉伸，即在此情况下不添加额外的拉伸）。

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

键 `tocskip` 定义列表 `\tableofcontents`、`\listoffigures` 和 `\listoftables` 内部的垂直分隔。默认情况下没有额外的分隔（即 `Opt`）。如果指定时没有值，则使用标准 `\parskip`，否则使用给定值。

键 `tocskip` 定义列表 `\tableofcontents`、`\listoffigures` 和 `\listoftables` 内部的垂直分隔。默认情况下没有额外的分隔（即 `Opt`）。如果指定时没有值，则使用标准 `\parskip`，否则使用给定值。

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

Execute any package options:

执行任何宏包选项：

```
13 \ProcessKeyvalOptions*
```

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

现在我们可以评估给定的选项并调整参数设置：attoc

```
14 \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.

如果未给出 `skip`（或它为空），则将 `\parskip` 设置为半个 `.5\baselineskip` 加上 2pt 的拉伸量。在此情况下，`\baselineskip` 内部的拉伸或收缩将被忽略。

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

Otherwise set it to the specified value: 否则将其设置为指定值：

```
17 \setlength\parskip\parskip@skip  
18 \fi
```

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

设置 `\parfillskip` 是 Donald Arseneau 在 comp.text.tex 上的某个时候建议的：

```
19 \setlength\parfillskip\parskip@parfill  
20 \advance\parfillskip Opt plus 1fil\relax
```

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

`\parindent` 将获得指定的任何值。如果给出键时没有选项，这实际上将重新分配现在的“当前”值。

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

5.2 Handling document elements

5.2 处理文档元素

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.

设置非零的 \parskip 在文档元素（如列表或标题等）中会产生一些副作用。在这里，我们尝试将这些副作用控制在一定范围内。

We make use of the etoolbox package to do patching.

我们使用 etoolbox 包进行修补。

```
22 \RequirePackage{etoolbox}
```

5.2.1 Lists

5.2.1 列表

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.

为此，列表环境中的垂直间距在所有相关位置（仅适用于 \normalsize）更改为使用与 \parskip 相同的值，即

```
\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 parsip (via the skip option) we shouldn't do this but rather keep the default list settings, so we better check for this.

但是，如果用户明确要求零 parsip（通过 skip 选项），我们不应该这样做，而应该保持默认的列表设置，因此我们最好检查这一点。

```
23 \ifdim \parskip > 0pt
24   \def\@listI{\leftmargin\leftmargini
25     \topsep\z@ \parsep\parskip \itemsep\z@}
26   \let\@listi\@listI
27   \@listi
28   \def\@listIi{\leftmargin\leftmarginii
29     \labelwidth\leftmarginii\advance\labelwidth-\labelsep
30     \topsep\z@ \parsep\parskip \itemsep\z@}
31   \def\@listIii{\leftmargin\leftmarginiii
32     \labelwidth\leftmarginiii\advance\labelwidth-\labelsep
33     \topsep\z@ \parsep\parskip \itemsep\z@}
34 %
35 % and finally ...
36 %
37 % 最后...
38 % \partopsep = \z@ % don't even add anything before first item (beyond
39 %                     % \parskip) even if the list is preceded by a blank line
40 %                     % 即使列表前面有一个空行，也不要在第一个项目之前添加任何内容（除了 \parskip）
41   \partopsep=\z@
42 \fi
```

5.2.2 TOCs and similar lists

5.2.2 目录和类似列表

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.

在目录或图列表中，我们不希望仅仅因为此类列表中的单行是作为单行段落实现的而产生任何额外的垂直间距。因此，我们默认在本地将 \parskip 设置为零。这真的应该是 LaTeX 已经完成的事情。

```
43 \patchcmd\@starttoc
```

```

44  {\begingroup \makeatletter
45  {\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.

像宏包原始版本那样仅将 `\parskip` 设置为零并不总是有效。如果列表以普通段落开始（而不是像通常那样以 `\addvspace` 开始），我们将获得零 `\parskip`，但上面的标题假设我们获得正常的 `\parskip`，因此已从其自身的垂直间距中移除了该数量。只要 `parskip` 值不是太大，人们就不会注意到标题和列表彼此靠近，但如果使用，例如，`[skip=20pt]`，您甚至会看到重叠。

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

因此，我们执行以下操作：我们查看最后一个间距，取消它，然后发出一个等于 `\par skip + \lastskip` 的间距。这样，任何后续代码看到的间距都具有正确的值，这对于 `\addvspace` 计算很重要。只有这样，我们才会在本地将 `\par skip` 设置为零，或者更确切地说，设置为 `\par skip@tocskip`，这是用户可以通过选项设置的参数。

```

46  \skip@\lastskip
47  \advance\skip@\par skip
48  \vskip-\lastskip
49  \vskip\skip@
50  \par skip\par skip@tocskip}
51 {}{\typeout{Couldn't patch \string@\starttoc}}

```

5.2.3 Standard headings

5.2.3 标准标题

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

出于同样的原因，我们不希望在显示标题之前和之后看到额外的 `\par skip` 被添加，因此我们减去其值（在两个地方）：

```

52 \patchcmd@\startsection
53   {\addvspace@\tempskipa}
54   {\advance@\tempskipa-\par skip\addvspace@\tempskipa}
55   {}{\typeout{Couldn't patch \string@\startsection}}
56 \patchcmd@\xsect
57   {\vskip@\tempskipa}
58   {\advance@\tempskipa-\par skip\vskip@\tempskipa}
59   {}{\typeout{Couldn't patch \string@\xsect}}

```

5.2.4 titlesec headings

5.2.4 titlesec 标题

If `titlesec` is used then headings are built using different commands and we have to cancel the `\par skip` 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:

如果使用 `titlesec`，则标题使用不同的命令构建，我们必须在那里取消 `\par skip`。原理相同。当然，只有当该包真正被加载时才应该进行修补，因此我们将其推迟到文档开始时并进行测试：

```

60 \AtBeginDocument{%
61 \ifx\ttl@straight@ii\undefined\else % titlesec got loaded
62 \patchcmd@\ttl@straight@ii
63   {\addvspace{\tempskipa}}%
64   {\advance@\tempskipa-\par skip\addvspace@\tempskipa}%
65   {}{\typeout{Couldn't patch \string\ttl@straight@ii}}%

```

```

66 \patchcmd\ttl@straight@ii
67   {\vspace{\@tempskipb}}%
68   {\advance\@tempskipb-\parskip \vspace{\@tempskipb}}%
69   {}{\typeout{Couldn't patch \string\ttl@straight@ii}}%
70 \patchcmd\ttl@part@ii
71   {\vspace*{\@tempskipa}}%
72   {\advance\@tempskipa-\parskip \vspace*{\@tempskipa}}%
73   {}{\typeout{Couldn't patch \string\ttl@part@ii}}%
74 \patchcmd\ttl@part@ii
75   {\vspace{\@tempskipb}}%
76   {\advance\@tempskipb-\parskip \vspace{\@tempskipb}}%
77   {}{\typeout{Couldn't patch \string\ttl@part@ii}}%
78 \patchcmd\ttl@page@ii
79   {\vspace*{\@tempskipa}}%
80   {\advance\@tempskipa-\parskip \vspace*{\@tempskipa}}%
81   {}{\typeout{Couldn't patch \string\ttl@page@ii}}%
82 \patchcmd\ttl@page@ii
83   {\vspace{\@tempskipb}}%
84   {\advance\@tempskipb-\parskip \vspace{\@tempskipb}}%
85   {}{\typeout{Couldn't patch \string\ttl@page@ii}}%
86 \fi}

```

5.2.5 **amsthm** theorems

5.2.5 **amsthm** 定理

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.

amsthm 包是少数几个对 **\parskip** 进行显式校正的包之一，如果加载了此 **parskip** 包，这种校正就不再足够。因此，如果加载了该包，我们将从该包中移除该设置。

```

87 \AtBeginDocument{%
88 \ifx\deferred@thm@head\@undefined\else % amsthm got loaded
89 \patchcmd\deferred@thm@head
90   {\addvspace{-\parskip}}{}%
91   {}{\typeout{Couldn't patch \string\deferred@thm@head!}}%
92 \fi}

```

5.3 Closing shop

5.3 结束工作

```
93 (*package)
```