

自然科学引文和参考文献*

(作者-年份格式与数字格式)

Patrick W. Daly

翻译: [Marlin](#)

2015 年 2 月 14 日

摘要

natbib 宏包重定义了 L^AT_EX 命令 `\cite`, 可以采用作者年份格式或者数字格式引用文献, 适用于 plain 等标准的参考文献格式, 也与 harvard, apalike, chicago, astron, authordate 以及 natbib 等兼容。

与上述宏包相比, natbib 宏包不仅支持众多的作者年份格式, 也支持标准的数字格式引用。事实上, 它还可以在作者年份的文献格式下产生数字格式引用, 而且很容易在两种引用模式间切换。为此, 它也提供了替代标准 L^AT_EX 文献格式的专用格式 (.bst)。

narbib 宏包可以定义引用格式 (如括号以及不同引用条目间标点的类型等), 甚至可以关联文献格式名以自动激活不同引用格式, 也可以通过当前的配置文件 natbib.cfg 为 .bst 文件定义引用格式。

natbib 宏包与 babel, index, citeref, showkeys, chapterbib, hyperref, koma 等宏包以及 amsbook, amsart 等文档类兼容, 也能实现 cite 宏包的排序与压缩功能, 还能实现 Thorsten Ohl 写的 mcite 宏包的多个引用的合并功能。然而, natbib 宏包本身与 cite 或 mcite 宏包不兼容。

应该注意的是实现文献列表中增加引用页码功能的 citeref 宏包必须在 natbib 宏包之后调用。(调用 hyperref 宏包时设定 pagebackref 选项也有此功能, 而且提供了超链接。)

此外, natbib 宏包为大多常见的参考文献格式提供了统一而灵活的接口。

*本文档适用于 natbib 宏包 8.31b 版 (2010/09/13)。

目录

1	简介	3
2	如何使用本宏包	4
2.1	新的参考文献格式	4
2.2	thebibliography 的语法	5
2.3	基本的引用命令	5
2.4	扩展的引用命令	7
2.5	强制姓名大写	8
2.6	引用别名	8
2.7	缺少作者和年份信息的文献	9
2.8	plainnat 系列格式的扩展	10
2.9	引文输出格式的选择	10
2.10	预定义引用格式	12
2.11	格式命令的优先权	13
2.12	其他的格式选项	14
2.13	引用的自动索引	15
2.14	与 HyperTeX 的兼容性	15
2.15	一个文档内有多个参考文献	16
2.15.1	为 natbib 和 chapterbib 的特殊考虑	16
2.16	数字格式引用的排序与压缩	17
2.17	数字格式文献的合并	17
2.18	初次引用时使用全部作者列表	18
3	作者年份格式下的数字格式引用	18
3.1	选择数字模式	19
4	当前配置文件	19
5	宏包选项	19
6	natbib 宏包使用简要介绍	21
7	docstrip 选项	21
8	作者年份格式的其他解决方案	22
8.1	natsci.bst 格式	22
8.2	apalike.bst 格式	22

8.3 newapa 格式	22
8.4 Harvard 系列	23
8.5 Astronomy 格式	23
8.6 作者日期格式	24

1 简介

natbib 宏包扩展了 L^AT_EX，不仅提供数字格式引用，也提供作者年份格式引用。标准 L^AT_EX 只允许数字格式引用，而 1993 年 natbib 宏包发布之前，所有实现作者年份格式的扩展都局限于此。因为这些扩展一般都增加了新的命令（natbib 宏包也是这样），使用了这些命令的文档在扩展编辑后只能用于数字格式引用。

natbib 宏包作了调整，切换作者年份格式与数字格式只需设置一个选项，不需要改动源文档。此宏包现在已经是标准 L^AT_EX 安装包的一部分，符合很多期刊的要求。同时，由于它的易用性以及可配置性，此宏包已经在众多 L^AT_EX 社区作为引用宏包的首选。

跟所有宏包一样，natbib 宏包可以附带可选项在导言区调用，例如

```
\usepackage[sectionbib,square]{natbib}
```

可选项 sectionbib 指明，当调用宏包 chapterbib 宏包时，参考文献作为一节出现在各章最后（参见第2.15节）。可选项 square 指明参考文献的引用放在一对方括号之内，而不是放在一对圆括号之内。调用 natbib 宏包时可使用的全部选项参见第5节。

在文档正文开始处可以指定参考文献格式，如

```
\begin{document}  
\bibliographystyle{plainnat}
```

这里 plainnat 指明参考文献格式，B_BT_EX 程序使用它从参考文献数据库生成实际的参考文献。plainnat 是标准的 plain 格式（仅限数字格式）在 natbib 下的对应版本。别的可用参考文献格式详见第2.1节或在 T_EX 系统安装目录查找.bst 文件。

命令\bibliographystyle 可以放置在文档的任何位置，但还是建议放在文档开始处，以便于辨认、修改。

在正文中引用文献，形如

```
\citep{jon90} 生成括号引用 (Jones et al., 1990)  
\citet{jon90} 生成文本引用 Jones et al. (1990)
```

这里\citep 和\citet 并不是 L^AT_EX 的标准命令，而是 natbib 宏包定义的专用命令。L^AT_EX 标准命令\cite 应尽量避免使用，因为 natbib 宏包重定义了此命令，使它在作者年份格式下等同于\citet，在数字格式下等同于\citep。natbib 宏包还定义了很多其他命令产生一些特别的效果（参见第2.4节）。

上述例子中的 jon90 是文献的引用标签，在 B_BT_EX 文献数据库中定义，或者在 thebibliography 环境中定义（参见第2.2节），如

```

\begin{thebibliography}{1}
  \bibitem[Jones et al.(1990)]{jon90}
  . . . . .
\end{thebibliography}

```

此环境生成真正的参考文献，`\bibitem` 命令将文献条目与引用通过引用标签关联起来，这里的 `jon90` 就是引用标签。引用标签可以任意指定，只要不重复即可。方括号中内容为文献的引用信息，作者为 Jones et al.，年份为 1990。注意这两部分内容根据引用命令能以不同的方式结合在一起。事实上，如果使用了数字格式的引用，方括号中的引用信息会被忽略，只有顺序数字作为引用。

`thebibliography` 环境可以手工书写，但更好更安全的方式是由 `BIBTEX` 来生成。为此，我们需要使用前文提到的 `\bibliographystyle` 命令，以及在文档末尾放置

```

\bibliography{mybib}
\end{document}

```

这里 `mybib` 是 `BIBTEX` 文献数据库名，后缀为 `.bib`。`mybib.bib` 中包含了文档中引用到的文献的相关信息。

本文档其余部分介绍了 `natbib` 宏包可能的全部细节。

2 如何使用本宏包

本文中，我们区分了引用模式（引用文献时的标点类型等）与引用格式（作者年份格式与数字格式）。引用模式不同于参考文献格式，并不是在 `.bst` 文件中指定。

2.1 新的参考文献格式

`natbib` 宏包提供了三种文献格式 `.bst` 文件来代替标准 `LATEX` 数字格式：

```
plainnat.bst    abbrvnat.bst    unsrtnat.bst
```

它们可以用来生成与相应标准格式风格相同的文献列表，但更适于 `natbib` 宏包。其优势在于它们在数字格式与作者年份格式下都可以使用。

这些 `.bst` 文件并没有穷尽所有格式，还有很多适于 `natbib` 宏包的格式。当然也可以使用本宏包作者提供的 `custom-bib` 程序（称为 `makebst`）生成自己的 `.bst` 格式文件。

2.2 thebibliography 的语法

thebibliography 环境中的 `\bibitem` 命令提供所引用文献的作者姓名及年份等信息。natbib 宏包希望这些信息能以上面提到的格式文件 .bst 指定的样式呈现出来。（也可以由早期的 Harvard 和 Chicago 等宏包使用。）如果不使用 BibTeX，就必须自己设定 thebibliography 环境，使它符合 natbib 宏包要求。

需要使用如下形式的语法：

```
\bibitem[Jones et al.(1990)]{jon90}...
```

或者

```
\bibitem[Jones et al.(1990)Jones, Baker, and Williams]{jon90}...
```

方括号中的内容包含了引用信息，缩略作者列表 Jones et al.，年份 1990，可选的完整作者列表 Jones, Baker, and Williams。如果省略完整作者列表，则用缩略作者列表代替。将年份括起来的括号并不是引用信息必须的，只是用来界定作者列表与年份。即使在引用模式中使用方括号，这里也必须使用圆括号。在年份括号前后不能有空格，否则它将被看成是作者列表的一部分。

注：如果某一条 `\bibitem` 条目不符合 natbib 的语法，将被强制转换成数字格式，因为此时 natbib 宏包无法确定作者与年份信息。

2.3 基本的引用命令

natbib 宏包也可以与为别的宏包甚至陈旧的宏包如 Harvard 等设计的文献格式配合使用。然而，这里与下节介绍的命令是由 natbib 宏包定义的，也可以由其他文献格式使用。

natbib 宏包提供了两个基本的引用命令，`\citet` 与 `\citep` 分别实现文本引用与括号引用。还提供了带 * 的版本 `\citet*` 与 `\citep*`，区别在于生成完整的作者列表，而不是缩略作者列表。这些命令都可以通过可选参数在文献引用标记前后增加说明信息。

<code>\citet{jon90}</code>	⇒ Jones et al. (1990)
<code>\citet[chap.~2]{jon90}</code>	⇒ Jones et al. (1990, chap. 2)
<code>\citep{jon90}</code>	⇒ (Jones et al., 1990)
<code>\citep[chap.~2]{jon90}</code>	⇒ (Jones et al., 1990, chap. 2)
<code>\citep[see] []{jon90}</code>	⇒ (see Jones et al., 1990)
<code>\citep[see] [chap.~2]{jon90}</code>	⇒ (see Jones et al., 1990, chap. 2)
<code>\citet*{jon90}</code>	⇒ Jones, Baker, and Williams (1990)
<code>\citep*{jon90}</code>	⇒ (Jones, Baker, and Williams, 1990)

带 * 的版本只能在 .bst 格式支持的情况下生成完整作者列表，否则生成缩略作者列表。

在标准 L^AT_EX 中，`\cite` 命令只能有一个可选参数提供引用标记后的说明信息，这里一个可选参数提供引用标记后的说明信息，两个可选参数分别提供引用标记前和引用标记后的说明信息。如果只需要引用标记前的说明信息，像上述例子那样，必须提供一个对应引用标记后信息的空的可选参数。

更复杂的文本与引用的混合使用可以由用途更多的 `\citetext` 命令实现，参见第 2.4 节。

多个引用可以由一个 `\cite` 命令使用多个引用标签实现。如果相邻的引用文献有相同的作者但年份不同，则作者信息不会重复出现。

```

\citet{jon90,jam91}    ⇒ Jones et al. (1990); James et al. (1991)
\citep{jon90,jam91}   ⇒ (Jones et al., 1990; James et al. 1991)
\citep{jon90,jon91}   ⇒ (Jones et al., 1990, 1991)
\citep{jon90a,jon90b} ⇒ (Jones et al., 1990a,b)

```

这些例子是针对作者年份格式的，如果是数字格式，结果将是：

```

\citet{jon90}          ⇒ Jones et al. [21]
\citet[chap.~2]{jon90} ⇒ Jones et al. [21, chap. 2]
\citep{jon90}          ⇒ [21]
\citep[chap.~2]{jon90} ⇒ [21, chap. 2]
\citep[see] []{jon90}  ⇒ [see 21]
\citep[see] [chap.~2]{jon90} ⇒ [see 21, chap. 2]
\citep{jon90a,jon90b} ⇒ [21, 32]

```

只有 .bst 文件支持作者年份格式，作者信息才会显示。标准的 .bst 文件如 plain.bst 只支持数字格式，不会将作者年份信息传递给 L^AT_EX。这种情况下，`\citet` 将生成形如“(author?) [21]”的引用标记。

在 natbib 的早期版本中，`\cite` 命令可以用于文本或括号引用，放在方括号中的内容表示插入说明信息。出于兼容性考虑，保留了这种用法，但不再提倡这么用。

因为不带说明信息的 `\cite` 命令在作者年份格式下相当于 `\citet`，在数字格式下相当于 `\citep`。带 * 的版本以及带一个或两个说明信息的命令，也是类似使用。

有时候需要多条引用信息按照参考文献列表中的顺序排列，而不是按照 `\cite` 参数中的排列顺序。宏包选项 `sort` 实现此功能。详情参见第 2.16 节。

有些出版机构要求所有文献在第一次被引用时给出全部作者列表，之后的引用只需给出缩略作者列表。调用 natbib 宏包时加 `longnamesfirst` 选项即可实现此功能。详情参见第 2.18 节。

2.4 扩展的引用命令

引用的另一种形式，`\citealt` 命令与`\citet` 命令的差别只在于引用标记中没有括号。类似地，`\citealp` 命令是`\citep` 命令没有括号的版本。

`\citenum` 命令生成数字编号引用，即使在作者年份格式下，但不产生括号，也不会生成上标形式。这样设计是为了更容易的以普通文本的方式对引用编号进行引用。 `\citenum`

除了`\citenum` 之外，这些命令也可以一次引用多条引文献，添加说明信息，以及带 * 版本等用法。

```

\citealt{jon90}           ⇒ Jones et al. 1990
\citealt*{jon90}         ⇒ Jones, Baker, and Williams 1990
\citealp{jon90}          ⇒ Jones et al., 1990
\citealp*{jon90}         ⇒ Jones, Baker, and Williams, 1990
\citealp{jon90,jam91}    ⇒ Jones et al., 1990; James et al., 1991
\citealp[pg.~32]{jon90} ⇒ Jones et al., 1990, pg. 32
\citenum{jon90}          ⇒ 11
\citetext{priv.\ comm.} ⇒ (priv. comm.)

```

`\citetext` 命令可以在当前引用标记括号内插入任意文本内容。此命令经常与`\citealp` 命令配合使用。例如： `\citetext`

```
\citetext{see \citealp{jon90}, or even better \citealp{jam91}}
```

生成如下形式的引用：

```
(see Jones et al., 1990, or even better James et al., 1991).
```

在作者年份格式下，有时候只需要引用作者而不需要引用年份，或者是只需要年份而不需要作者。此宏包提供了一系列扩展的引用命令。 `\citeauthor`

```

\citeauthor{jon90} ⇒ Jones et al.
\citeauthor*{jon90} ⇒ Jones, Baker, and Williams
\citeyear{jon90} ⇒ 1990
\citeyearpar{jon90} ⇒ (1990)

```

`\citeyear`
`\citeyearpar`
`\citefullauthor`

这里的`\citefullauthor` 命令等价于`\citeauthor*` 命令。

如果文献条目中缺少完整的作者列表，`\citeauthor*` 命令等同于`\citeauthor` 命令，只生成缩略作者列表。这种情形也适用于`\citet` 命令`\citep` 命令。

跟使用标准 L^AT_EX 格式文件.bst 一样，如果文献条目中缺少作者或者年份信息，这些命令会产生警告信息。

注：即使使用了作者年份格式.bst，这些命令也可以用于数字模式引用。

注：这些`\cite..` 命令有相同的语法，允许多条文献引用，可以有附加说明信息（`\citeyear` 和`\citenum` 命令没有带 * 版本的用法）。为`\citeyear` 和`\citeauthor`

命令添加说明信息没有实际意义，尤其是多条文献引用时，但是可以这么使用，也不会报错，只是显示结果会有些奇怪而已。例如，在数字引用模式下，这些说明信息完全被忽略；在作者年份引用模式下，只接受引用标记后的说明信息。也不推荐在`\citet`命令中引用多条文献（在我看来这没有什么意义），如果这么用了，而且添加了说明信息，引用标记前的说明信息会出现在每一个年份前，而引用标记后的说明信息只出现在最后一个年份后。这些都是显而易见的问题，但没必要为了解决不合理的使用造成的问题而花费太多精力。

总的来说，只提倡在`\citep`命令中使用附加说明信息，但是也可以在作者年份引用模式下，在`\citet`命令引用单条文献时使用附加说明信息。在别的情形下使用附加说明信息会产生不可预期的结果。

2.5 强制姓名大写

如果首作者的姓名中含有 `von` 部分，例如“della Robbia”，即使在句首，`\citet{dRob98}`也只能产生形如“della Robbia (1998)”的引用。此时，我们可以使用`\Citet`命令强制首字母大写。还有一些其他的强制首字母大写的命令。

when	<code>\citet{dRob98}</code>	⇒	della Robbia (1998)
then	<code>\Citet{dRob98}</code>	⇒	Della Robbia (1998)
	<code>\Citep{dRob98}</code>	⇒	(Della Robbia, 1998)
	<code>\Citealt{dRob98}</code>	⇒	Della Robbia 1998
	<code>\Citealp{dRob98}</code>	⇒	Della Robbia, 1998
	<code>\Citeauthor{dRob98}</code>	⇒	Della Robbia

这些命令也都有相应的显示完整作者列表的带 * 版本的用法。

注：实现首字母大写命令的写法有些讨巧，可以作用于写在`\bibitem`文献条目中作者信息，甚至使用了旧的字体设置命令也可以使用，但是 \LaTeX 2_ϵ 字体设置命令会产生冲突。例如

```
\bibitem[{\it della Robbia}(1998)]{dRob98} is okay, but
\bibitem[\textit{della Robbia}(1998)]{dRob98} crashes.
```

2.6 引用别名

有时候我们希望通过某个特殊的名称来引用文献，而不是作者信息等方式，如“Paper I”、“Paper II”等。可以定义并使用这些别名，不论是文本引用还是括号引用，例如：

```
\defcitealias
\citetalias
\citepalias
```

```

\defcitealias{jon90}{Paper~I}
\citetalias{jon90}           ⇒ Paper I
\citepalias{jon90}          ⇒ (Paper I)

```

这些命令的使用类似于`\citet`和`\citep`，可以在参数中引用多条文献，可以包含附加说明信息，可以标记为超链接，等等。

如果在定义之前就使用了别名，或者重定义一个已经使用了的别名，会产生警告信息。如果给一个不存在的文献条目定义别名，并不会产生警告信息，直到此别名被使用时才会产生警告信息！

第2.7节给出了通过代码名称引用的方法。

2.7 缺少作者和年份信息的文献

对于没有作者信息的文献如何引用？这是困扰了我很久的问题，但现在我有了解决办法。在标准的`BIBTEX`格式中，当作者或编者信息缺失时，在文献条目中使用一个称为“KEY”的域来按字母顺序排序。在作者年份格式中，更进一步将“KEY”域插入到作者信息的位置。我们可以想象此时为文献指定了代码名称。例如

```

@MANUAL{handbk98,
  title = {Assembling Computers},
  year = 1998,
  organization = {MacroHard Inc.},
  key = "MH-MAN"
}

```

使用`plain`文献格式时，`key`的值“MH-MAN”只是用于文献列表中对文献排序；使用`plainnat`或别的作者年份文献格式时，`key`的值被直接用于作者信息。我们可以通过`\citeauthor{handbk98}`引用此文献得到MH-MAN，或者通过`\citetext{\citeauthor{handbk98}}`得到括号引用(MH-MAN)。

如果参考文献格式将文献条目里可选项中的日期信息留空，`natbib`宏包处理起来更简单，例如

```

\bibitem[MH-MAN()] {handbk98}

```

引用标记中会取消日期，标点以及括号，就像`\citet`命令的效果一样。这意味着`natbib`宏包会像上面的例子那样，自动匹配`\citet`或`\citep`的格式。而文献列表中的文献条目中仍然会显示日期信息。

当缺失的作者或编者信息由 key 域的值替换后，natbib 宏包根据省略了日期信息的文献条目对文献格式做了相应修改。

类似地，如果年份信息缺失，在文献条目中也会将年份域留空，引用这样的文献将只生成作者姓名。

注：这种情形有各种可能的处理。比如可以将引用代码放置到文献条目的作者位置，将年份域留空，然后像 alpha 格式（字母顺序排列）那样生成引用。第二组代码（甚至就是作者姓名）放置到一般情况下完整作者姓名列表出现的位置，以便 `\cite*` 命令使用。例如

```
\bibitem[MH-MAN()MacroHard Inc.]{handbk98}
```

2.8 plainnat 系列格式的扩展

第2.1节中提到的 natbib 宏包特有的 .bst 格式文件与标准的格式相比，有一些扩展的域：

ISBN 为书籍类文献添加 ISBN 号

ISSN 为期刊类文献添加 ISSN 号

URL 为在线文献添加网址链接

DOI 现在很多期刊采用的比 URL 更可靠的数字对象标识符

EID 在线期刊和纸质期刊用来代替页码的电子身份证，也用作期刊的序列号

DOI 和 URL 往往都比较长，这会导致难看的断行或者突出到页边界。这个问题可以通过调用 Donald Arseneau 编写的 url 宏包解决，此宏包允许文本在不带连字符的标点处断行。该宏包的调用可由 natbib 宏包自动检测到，并重定义一些适当的命令。URL 显示为打字机字体，DOI 显示为罗马字体。如果没有调用 url 宏包，URL 和 DOI 这些内容将不会自动断行。

像第2.7节提到的那样，plainnat 格式和 plain 格式对 key 域的值采用不同的处理方式。plain 格式下，只是将 key 域的值用于没有作者信息时对文献字母顺序排序。plainnat 格式下，将 key 域的值实际插入到文献信息与引用标记的作者位置。进一步，如果文献条目中年份信息也留空，`\citep` 命令只能生成“作者”内容，此时即显示 key 域的值。这应该是为此文献设定的代码。

2.9 引文输出格式的选择

上面的例子显示了默认的引用风格的样式。不论是数字格式还是作者年份格式，这些样式都可以通过 `\setcitestyle` 命令改变。此命令采用一组逗号分隔的 `\setcitestyle` 关键词作为参数。（这个命令是版本 8 以上的 natbib 宏包新增加的。）

- 引用模式: `authoryear` (作者年份格式)、`numbers` (数字格式)、`super` (数字上标格式)。(对应于`\bibpunct` 命令的第四个参数)
- 括号: `round` (圆括号)、`square` (方括号)、`open={左括号}` 和 `close={右括号}`。(对应于`\bibpunct` 命令的第一和第二个参数)
- 引用标记间的分隔符: `semicolon` (分号)、`comma` (逗号)、`citesep={分隔符}`。(对应于`\bibpunct` 命令的第三个参数)
- 作者与年份间的分隔符: `aysep={分隔符}`。(对应于`\bibpunct` 命令的第五个参数)
- 同一个作者不同年份间的分隔符: `yysep={分隔符}`。(对应于`\bibpunct` 命令的第六个参数)
- 引用命令可选参数说明信息前的分隔符: `notesep={分隔符}`。(对应于`\bibpunct` 命令的可选参数)

默认设置相当于:

```
\setcitestyle{authoryear,round,comma,aysep={;},yysep={,},notesep={,}}
```

例 1, 在作者年份格式下,

```
\setcitestyle{square,aysep={},yysep={;}}
```

```
\citep{jon90,jon91,jam92}
```

将生成 [Jones et al. 1990; 1991, James et al. 1992] 形式的引用。

例 2, 在作者年份格式下,

```
\setcitestyle{notesep={; },round,aysep={},yysep={;}}
```

```
\citep[and references therein]{jon90}
```

将生成 (Jones et al. 1990; and references therein) 形式的引用。

注:

- 没有指明修改的参数保持不变;
- 参数的次序无关紧要;
- 作者与年份间的分隔符只用于作者年份格式的引用, 数字格式下无效;
- 不同年份间的分割用于一次性引用的多条文献有相同作者的情形, 而且总会插入一个空格; 如果年份也相同, 则生成形如 ‘2007a,b’ 的引用, 此时不会插入空格; 如果希望此时插入空格, 参数可以设置为`yysep={,~}`;

- 在数字格式下，相同作者多条文献的引用，例如`\citet{jon90,jon91}` 会生成形如 ‘Jones et al. [21, 22]’ 的引用，数字间插入标点；而且自动加入空格，但在数字上标格式下不加入空格；
- 除逗号外，单个符号不需要放在 `{ }` 内，例如，`yysep=;` 是可以接受的参数。

原有的设置引用格式的命令是`\bibpunct`，此命令带有六个必选参数与一个可选参数：

`\bibpunct`

1. 引用标记的左括号，默认是 (
2. 引用标记的右括号，默认是)
3. 多条引用间的分隔符，默认是;
4. 字母 n 代表数字格式，字母 s 代表数字上标格式，别的字母代表作者年份格式，默认是作者年份格式
5. 作者与年份之间的分隔符
6. 相同作者的文献连续引用时，年份或数字之间的分隔符，默认是,

可选参数是说明信息前的符号，默认是逗号加一个空格。如果重定义了分隔符，需要其后的空格的话，必须显式指定。

上述`\setcitestyle` 命令的两个例子，可以由如下命令分别实现：

```
\bibpunct{[ ]}{,}{a}{;}
\bibpunct[,~]{( )}{,}{a}{;}

```

2.10 预定义引用格式

如果要经常使用一种特定的引用记号，可以将这些记号放置在当前的配置文件 `natbib.cfg` 中，然后通过`\citestyle{name}` 命令调用。这里 `name` 是通过`\bibstyle@name` 命令为想要使用的引用记号定义的名称。

`\citestyle{name}`

`\bibstyle@name`

例如，美国地球物理学会（American Geophysical Union, AGU）旗下出版物要求引用标记使用方括号，用分号分隔。有专门的 `agu.bst` 完成大部分格式的设置，但没有包括这些引用标记的要求。`natbib` 宏包有如下定义以及相应的命令

```
\newcommand{\bibstyle@agu}{\bibpunct{[ ]}{;}{a}{,}{,}{~}}
\citestyle{agu}

```

来实现这些要求。

这样预定义引用风格的方法特别之处在于：`natbib` 宏包在文档开始处调用参考文献格式名称来执行`\citestyle`，就像使用`\bibliographystyle` 命令一样（储存在辅助文件`.aux` 中）。这意味着引用风格可以直接与参考文献格式`.bst` 文件关联。这种实现方式可以由指定的文献格式立即改写引用风格，就像使用了宏包参数或`\setcitestyle`，`\bibpunct`，`\citestyle` 等命令一样。

`natbib` 宏包为以下文献格式预定义了引用风格：

plain 等 4 种标准格式： 方括号，数字格式，逗号分隔

plainnat 等格式： 方括号，作者年份格式，逗号分隔

agu（美国地球物理学会）： 方括号，作者年份格式，分号分隔

egu（欧洲地球科学学会）： 圆括号，作者年份格式，逗号分隔

agms, dcu, kluwer（Harvard 系列）： 圆括号，作者年份格式

cospar（太空研究委员会）： 斜线，数字格式，逗号分隔

nature（自然期刊）： 上标

还有很多出于我个人便利考虑的设置。上面列出了大多数主要的改变，可以满足各种一般要求。可以将这些定义放到当前的配置文件 `natbib.cfg` 实现与别的参考文献格式的自动关联。

注意 `plain` 和 `plainnat` 的预定义中指定了方括号，从而改变了 `natbib` 宏包默认的圆括号。

除了`\bibpunct` 和`\setcitestyle` 命令外，还有很多定义引用风格的方法。一些数字格式的引用模式常希望有更多的改变，有时会要求文献列表中只生成数字而没有常规的方括号。为适应这样的要求，`natbib` 宏包可以这样定义风格：

```
\newcommand{\bibstyle@nature}%
  {\bibpunct{}{}{,}{s}{}{\textsuperscript{,}}}%
  \renewcommand\bibnumfmt[1]{##1.}}
```

重定义的`\bibnumfmt` 命令指明了文献列表中文献数字编号的显示格式。

2.11 格式命令的优先权

引用风格（标点以及引用模式）可以由`\setcitestyle`，`\bibpunct`，`\citestyle` 等命令以及通过`\bibliographystyle{bst}` 调用预定义的`\bibstyle@bst` 等方式

来指定。还可以通过调用 `natbib` 宏包时指定可选参数来实现，第5节。如果同时有几种相冲突的选择会发生什么呢？

`\bibliographystyle` 命令具有最低的优先级，因为这种设定引用风格方式对用户并不透明。接下来是宏包调用时指定可选参数的方法。最后是 `\setcitestyle`，`\bibpunct`，`\citestyle` 等命令，可以改变其他方式指定的风格。

2.12 其他的格式选项

参考文献列表通常以 `\section*` 或 `\chapter*` 的形式出现，取决于文档所使用的类型。如果想设定自己特定的标题，比如要改成带编号的 `\section` 形式，可以由用户自行重定义 `\bibsection` 来实现。

`\bibsection`

`\bibpreamble` 命令指定对文献列表的说明信息，其内容会在文献列表标题之后，列表之前插入。这些内容通常以正文字体显示，除非在此命令参数中设定特定字体。`\bibfont` 命令设定整个文献列表的总体字体，但不设定列表前说明信息的字体。

参考文献列表通常采用与正文相同的字体尺寸与风格。然而，可以通过重定义 `\bibfont` 来设定文献列表中说明信息后的字体。例如

```
\renewcommand{\bibfont}{\small}
```

数字格式引用时编号可以使用不同的字体。重定义 `\citenumfont` 可以声明字体如 `\itshape`，也可以用带参数的命令如 `\textit`。

`\citenumfont`

```
\renewcommand{\citenumfont}[1]{\textit{#1}}
```

上例中的用法优于使用 `\itshape`，因为它可以自动进行斜体校正。

参考文献列表中数字编号的格式可以由重定义 `\bibnumfmt` 命令来设定。例如

```
\renewcommand{\bibnumfmt}[1]{\textbf{#1}:}
```

将默认的数字编号格式 [32] 改成粗体不带方括号的编号 **32**。

作者年份格式的文献列表通常采用悬挂缩进的格式：每条文献条目的第一行左对齐，其余行设置距离页边界一定的缩进量。此缩进量默认为 `1em`，可以通过 `\setlength` 命令重定义 `\bibhang` 的值来进行调整。

不论是作者年份格式还是数字格式，参考文献列表中文献条目之间的行间距由 `\bibsep` 的值确定。如果此值设成 `0pt`，则文献条目间没有额外行间距。其默认值取决于文档类设定的字体尺寸，基本上就是一条空行。可以通过 `\setlength` 命令重定义 `\bibsep` 的值来进行调整。

`\bibsep`

2.13 引用的自动索引

如果要文献引用放到索引文件.idx 中，只需要将 `\citeindextrue` 命令写在文档任意位置即可。之后所有的 `\cite` 命令及延伸命令都会将相应条目插入索引文件。写上 `\citeindexfalse` 命令，则相应条目不再插入索引文件。

`\citeindextrue`
`\citeindexfalse`

`thebibliography` 环境中的文献条目命令 `\bibitem` 也会产生索引条目。如果不希望产生索引，只需将 `\citeindexfalse` 命令放置到 `\bibliography` 或 `\begin{thebibliography}` 之前即可。

当然了，跟通常用法一样，必须在导言区写上 `\makeindex` 命令才会激活索引功能。否则，不会生成任何索引。

要正常生成索引，在运行 `makeindex` 之前，需确保运行 `BiBTeX` 后编译文档两次以上。

索引条目的样式由内部命令 `\NAT@idxtxt` 设置，如果要更改样式，可以在当前配置文件 `natbib.cfg` 中重定义此命令。默认情况下，索引形式为当前括号形式与引用模式下，将缩略作者列表或文献编号数字写入括号内。

`\NAT@idxt`

`natbib` 宏包也可以与 David M. Jones 编写的 `index` 宏包一起使用。两个宏包的调用次序没有影响。

在 `index` 宏包中，可以由 `\newindex` 命令生成多个索引列表。例如，可以将所有文献引用索引放到单独的一个索引列表内。首先，这样的列表需要类似于如下形式的初始化：

```
\newindex{cite}{ctx}{cnd}{List of Citations}
```

然后，使用 `natbib` 宏包中的如下命令将自动引用索引关联到此索引列表：

```
\renewcommand{\citeindextype}{cite}
```

详情参见 `index.sty` 宏包文档。

2.14 与 HyperTeX 的兼容性

`natbib` 宏包与 Sebastian Rahtz 和 Heiko Oberdiek 编写的 `hyperref` 宏包兼容，可用于 `LaTeX`→`HTML` 转换，`pdfTeX`，`pdfmark` 等的实现。兼容是共有的性质，涉及两个宏包的编码及相互配合方面。

`hyperref` 宏包有个特殊的选项 `breaklinks`，允许链接文本自动断行到下一行，避免产生行溢出信息。默认情况下，链接文本是被限制在一行内的。对数字格式引用没有问题，但是作者年份格式引用常常会有较长的引用文本，容易产生行溢出问题。

2.15 一个文档内有多个参考文献

`natbib` 宏包与 Donald Arseneau 和 Niel Kempson 编写的 `chapterbib` 宏包兼容，该宏包允许在一个文档内有多个独立的参考文献列表。通常用法是一本书的各章有独立的参考文献列表，尤其是在各章由不同作者独立编写时。

`chapterbib` 宏包使各章作者可以一种很自然的方式使用参考文献，只有将各章汇集在一本书里的作者或编者才需要一些额外的工作。

宏包使用 `\include` 命令，每个被包含进来的文件有自己的参考文献。对大规模的书籍，这样的处理有很大的优势。

回顾 `\include` 命令的使用，主文档形如：

```
\documentclass{...}
\includeonly{ch2}
\begin{document}
  \include{ch1}
  \include{ch2}
  \include{ch3}
\end{document}
```

只导入 `ch2.tex` 文件，即 `ch1.tex` 和 `ch2.tex` 都存在。如果以前曾经完整地编译过整个文档，那么原来的所有计数器，特别是页码、章节编号、交叉引用等任然会保留为前一次编译的效果。这里的诀窍是每一个被包含进来的文件有自己单独的辅助 `.aux` 文件，其中包含了这些信息，即使相应的 `.tex` 文件没有被导入，每次编译时这些辅助文件也会被调用。辅助 `.aux` 文件中也包含了 `BIBTEX` 使用的引用信息，同样会被 `chapterbib` 宏包使用。

如果在导言区调用了 `chapterbib` 宏包 (`\usepackage{chapterbib}`)，每一个 `\cite` 命令或 `bibitem` 命令中的引用标签会关联到当前导入的文件，与其他文件中相同的引用标签区别开来。每一个这样被包含进来的文件必须包含自己的 `\bibliography` 和 `\bibliographystyle` 命令。在使用 `LATEX` 编译文档（至少两次）之前，要用 `BIBTEX` 分别编译各个子文档。

2.15.1 为 `natbib` 和 `chapterbib` 的特殊考虑

调用 `chapterbib` 和 `natbib` 宏包的次序是无关紧要的。

`chapterbib` 宏包提供了一个选项 `sectionbib`，可以在每章都有自己的参考文献时，使参考文献列表以 `\section*` 的格式替代 `\chapter*` 的格式排版。当 `natbib` 宏包也被调用时，该选项不起作用了，相应地，`natbib` 宏包提供了这一选项来实现此功能。（调用 `natbib` 宏包时总是可以使用 `sectionbib` 选项，但它只在文档类是 `book` 或者 `report` 以及它们的扩展类型时才有效。）

如果每一个被导入的子文档要显示自己的参考文献列表，就必须包含自己的`\bibliography`命令。当然，各子文档的这条命令调用的文献数据库可以不同。然而，特别之处在于每个子文档必须包含自己的`\bibliographystyle`命令，各子文档的这条命令还可以指定不同的文献格式。

`natbib`宏包 8.0 以上版本中，各章可以有不同的引用风格和模式（作者年份格式或数字格式）。`\setcitestyle`, `bibpunct`, `\citestyle`等命令可以在文档任何位置指定，尤其是在各章的子文档中。（这是唯一需要指明的地方。）

2.16 数字格式引用的排序与压缩

另一个由 Donald Arseneau 编写的宏包 `cite.sty` 也改写了数字格式引用文献时的标点以及模式，它的所有功能都可以由 `natbib` 宏包实现。然而，它也可以实现数字格式引用标记的排序和压缩，这是很多期刊要求的格式。

当一条引用命令中引用多条文献条目时，引用标记的正常顺序就是引用条目时的顺序。这通常会生成较长的引用标记列表，形如 `[4,2,8,3]`。调用了 `cite` 宏包后，会显示为 `[2-4,8]`。

使 `cite` 宏包与 `natbib` 宏包兼容是不可能的，因为它们都改写了原始的`\cite`命令。相应地，我参考了 `cite` 宏包的某些代码，使它可用于 `natbib` 宏包。这些代码可以在调用 `natbib` 宏包时使用 `sort` 或者 `sort&compress` 选项激活其功能。

在作者年份格式下，`sort` 选项使`\citet`或`\citep`命令中的引用标记按文献列表中的顺序排版。通常按姓名的字母顺序，然后是年份顺序。这样可以避免生成形如“James et al. (1994b,a)”的引用标记。在作者年份格式下，选项 `sort&compress` 等同于 `sort`。

2.17 数字格式文献的合并

注：合并数字格式文献的代码是 Arthur Ogawa 为美国物理学会设计的。

Thorsten Ohl 编写的 `mcite` 宏包不能跟 `natbib` 宏包一起使用。实现文献合并功能，需要在调用 `natbib` 宏包时使用 `merge` 选项。

使用 `merge` 选项，可以把`\citep`引用的几个不同文献放在文献列表中的同一项，为此需要使用 `*` 指明主要的和次要的文献。例如，`\citep{feynmann,*salam,*epr}` 生成一个引用标记，三篇文献在文献列表中被列在一条文献条目下，共用一个文献编号。

Thorsten Ohl 编写的 `mcite` 宏包的一个限制被突破：`*` 可以用于任何引用标签。此宏包语法方面的限制也不再适用。

另一个特色是文献条目中可以插入说明信息：在`\citep`命令的参数中，可以在引用标签前加两个可选参数，形如`\citep{[pre][post]key}`，这里 `pre` 为引用

标记前的说明信息，post 为引用标记后的说明信息。

因此，用户可以在文档中使用如下形式的引用：

```
text \citep*[{See, e.g., }][ for a simpler explanation]ablebaker}
```

上例演示了带 * 模式与两个参数的使用。

警告：因为逗号 (,) 是 `\citep` 命令语法的一部分，方括号 ([]) 只是文本记号而不是真正的定界符（像大括号那样），如果在参数中使用了逗号，必须将参数放在一个分组（一对大括号 {}）内。当然，如果在参数中使用了方括号，也是一样处理。

`elide` 选项也可以激活 `natbib` 的文献合并功能，但此时被合并文献的公共部分（如作者等信息）不重复出现，在文献条目中只生成一次。

`mcite` 选项关闭文献合并与省略功能，接受特殊的语法，* 以及说明信息等可选参数被忽略。

这些功能只在数字格式且使用括号模式时有效，就像排序与压缩功能一样受限。

它们也需要特定的 `.bst` 文件，就像美国物理学会提供给它们的 REVTeX 文档类的格式文件之类的。

2.18 初次引用时使用全部作者列表

另一个经常被 `natbib` 用户使用的功能，就像 `harvard` 宏包的标准格式那样：任何文献首次被引用时，生成完整的作者列表，其后的引用生成缩略的作者列表。当然可以通过首次引用时使用 `\citet*`，以后的引用使用 `\citet` 或 `\citep` 命令来实现此格式。然而，还是希望能自动实现此功能。

这项功能可以在调用 `natbib` 宏包时使用 `longnamesfirst` 可选参数来激活。

对一些有很多作者的文献，我们希望只是自动缩略显示这些文献的作者列表。这时候，我们可以在首次引用之前指明

```
\shortcites{<key-list>}
```

那些包含在 `key-list` 列表内的文献在首次引用时也会显示缩略作者列表。

完整的作者列表仍然可以在任何时候使用带 * 的引用命令来生成。

3 作者年份格式下的数字格式引用

在使用作者年份格式 `.bst` 文件时，也可以生成数字模式引用，只是在内容上有少许改动。不需要任何改动，使用 `\citet` 或 `\citep` 命令就可以生成合理的引用。很明显，反过来是不可能的。使用了数字格式的 `.bst` 文件时，绝不会生成作者年份模式的引用，原因很简单，此时没有作者年份信息传递给辅助 `.aux` 文件。

3.1 选择数字模式

默认情况下，natbib 宏包使用作者年份格式。这可以通过以下方式更改：

1. 选择预定义了引用模式的数字格式文件，可以是宏包文件也可以是当前的配置文件；
2. 在调用宏包时使用 numbers 或 super 选项；
3. 指明`\setcitestyle{numbers}`；
4. 使用`\citestyle` 命令指定预定义了数字模式的格式名称（像 plain 格式可以使用 plainnat 指定）。

如果`\bibitem` 命令指定的文献条目不能适用作者年份格式时，natbib 宏包会自动切换到数字格式。这项功能不能被抑制，因为这样的文献条目在作者年份格式下会有问题。

有一些特殊的“数字”模式，像标准的 alpha.bst 那样，在数字标记的位置使用了非数字标记，例如

```
\bibitem[ABC95]{able95}
```

调用 natbib 后，这个文献标签不适用作者年份格式，因此会被视为数字格式。引用模式会切换到数字模式，引用命令`\cite{able95}` 将生成形如 [ABC95] 的引用标记。

然而，像第2.7节末尾提到的那样，还有别的处理方式。上面的结果可以由如下代码获得：

```
\bibitem[ABC95()]{able95}
```

4 当前配置文件

也可以使用当前配置文件 natbib.cfg，如果存在此文件，会被读入到宏包末尾。配置文件里的代码会取代宏包里的相应代码，尽管它的主要目的是方便用户预定义自己的`\bibstyle@bst`，来适应当前的文献格式的标点等，或者供`\citestyle` 命令使用。

5 宏包选项

当宏包被调用时，可以通过可选参数来选择不同的模式，形如

```
\usepackage[options]{natbib}
```

可用的选项提供了另一种选择引用标点的方法：

round 默认值，引用标记使用圆括号；

square 方括号；

curly 大括号；

angle 尖括号；

semicolon 默认值，多条引用间用分号分隔；

colon 分号，早期宏包误用的选项；

comma 逗号；

authoryear 默认值，引用标记为作者年份格式；

numbers 数字格式引用；

super 上标数字格式引用，像 Nature 期刊的要求；

sort 多条引用按文献列表中的顺序排序；

sort&compress 像 sort 一样排序，如果可能就提供压缩的引用标记（像 3-6, 15）；

compress 只压缩不排序，因此压缩只在引用顺序为升序的数字标记时生效；

longnamesfirst 使任何文献在首次引用时生成完整作者列表，其后的引用生成缩略的作者列表；

sectionbib 将参考文献环境从“章”格式重定义为“节”格式，只在文档类包含“章”条目时有效，效果就跟使用 chapterbib 宏包一样；

nonamebreak 一条文献的所有作者排在一行内，会导致行溢出，但有助于解决某些超链接问题；

merge 允许在文献引用标签前使用 *，使之与前一条文献合并；

elide 省略被合并文献的共同部分，如作者、年份等；

mcite 识别（并忽略）合并文献的语法。

选项 `curly` 和 `angle` 并无多大意义，我增加它们只是为了完善常见括号类型。据我所知，其他的常见引用标记有斜线，如 `/21/`，或者形如 (Ref. 21) 的样式。这可以通过 `\setcitestyle{open={/},close={/}}` 之类的命令来实现。

宏包选项设定的模式可以通过 `\setcitestyle`，`\bibpunct` 或 `\citestyle` 等命令来改变。如果使用这些命令或宏包选项，可以取消 `\bibliographystyle` 命令自动设定的模式。

6 natbib 宏包使用简要介绍

可以通过 L^AT_EX 编译 `natnotes.tex` 文档获得使用 `natbib` 宏包的简要说明，此文档可以从 `natbib.dtx` 中通过 `docstrip` 使用选项 `notes` 来获得。这可以作为方便的 `natbib` 宏包使用手册。

这个文档也可以由 `natbib.ins` 安装宏包时自动提取出来。

7 docstrip 选项

源文件 `.dtx` 文件是要用文件分解工具 `docstrip` 来处理的，此命令有如下可选参数：

all 包含了所有其他选项；

apalike 允许 `\bibitem` 形式的最小 `apalike` 说明；

newapa 允许配合 `newapa.sty` 在 `\bibitem` 命令的可选参数中使用 `\citeauthoryear`；

chicago 跟 `newapa` 一样；

harvard 包含 `\harvarditem` 形式的说明；

astron 允许在 `\bibitem` 命令的可选参数中使用 `\astroncite`；

authordate 增加 `\citename` 命令的语法；

package 生成去掉大多数注释的 `.sty` 宏包文档；

notes 生成宏包使用说明的简要介绍，最终的文档是 L^AT_EX 编译好的；

driver 生成源 `.ltx` 文件，此文件可经 L^AT_EX 编译后得到宏包文档；还可以修改后得到各种变化后的文档（页码、字体、使用手册或带说明的代码等）。`natbib.dtx` 也是这样的源文档，但不能也不应该被用户修改。

`natbib.dtx` 本身也是源文档，可以直接由 L^AT_EX 编译。

8 作者年份格式的其他解决方案

本节内容仅为追溯历史考虑。

在 1993 年 natbib 宏包发布之前，有一些尝试提供作者年份格式的宏包，正是其中某些启发了 natbib 宏包的编写。这些宏包中还有一些至今仍在使用，鉴于此，natbib 宏包试图包含它们的 `\bibitem` 语法，从而与那些 .bst 文件兼容。

这些宏包大都是 L^AT_EX 格式的，不适合现在 L^AT_EX 2_ε 格式的风格。

8.1 natsci.bst 格式

在使用 Stephen Gildea 编写的 agujgr.sty 时，它的格式 natsci.bst 文件给了我最初的灵感。它告诉我问题是可以解决的。然而，Gildea 提供的 `\bibitem` 命令只有缩略作者列表，只能生成带括号的引用标记。

名称 natsci 代表自然科学 (natural sciences)，受此启发，本宏包命名为 natbib。(这是个公认的丑陋的名字，但现在已经广泛使用，不容易再修改了。)

8.2 apalike.bst 格式

B_BT_EX 程序和标准 .bst 文件的创始人 Oren Patashnik 也开发过作者年份格式的 apalike.bst，并有相应的 apalike.sty 宏包支持它。同样，它也只提供了带括号的引用标记。它的功能与 natsci 文件相同。

在此格式下，参考文献条目形如：

```
\bibitem[Jones et al., 1990]{jon90}...
```

这是可以考虑的最起码的形式。在 Patashnik 发布 apalike.bst 和 apalike.sty 后，我称之为 apalike 类型。然而，有很多独立的 .bst 文件沿袭了这种风格，比如 natsci 格式。

沿袭这种风格的参考文献格式包括：

```
apalike, apalike2, cea, cell, jmb, phapalik, phppcf, phrmp
```

8.3 newapa 格式

随后，Stephen N. Spencer 和 Young U. Ryu 做了主要的改进，开发了 newapa.bst 格式，编写了宏包 newapa.sty。在他们的格式中，在 `\bibitem` 命令标签中引入了三个必选的项目：完整作者列表、缩略作者列表、年份。这是通过在标签中引入 `\citeauthoryear` 命令实现的，形如：

```
\bibitem[\protect\citeauthoryear{Jones, Barker,
and Williams}{Jones et al.}{1990}]{jon90}...
```

实际上,这只是说明了`\citeauthoryear`命令的最基本用法;`newapa`文件更进一步使用命令替换了一些词与标点。例如,上例中的‘and’实际上是`\betweenauthors`,可以在`.sty`文件中定义。当然,`\citeauthoryear`也是在那个文件中定义的。一系列不同的`\cite`命令用来生成各种形式的引用,如完整作者列表,缩略作者列表,有没有日期,文本格式还是括号格式等等。

因此,文献条目中的`\citeauthoryear`命令是很灵活的,允许用户生成各种想要的引用格式。此命令被很多其他的格式文件以及宏包使用。它们都受到了`newapa.sty`的启发,尽管缺少额外的标点命令。

沿袭这种风格的参考文献格式包括:

```
newapa, chicago, chicagoa, jas99, named
```

注:上面提到的`named.bst`使用`\citeauthoryear`命令时稍有变化,只有两个参数:缩略作者列表和年份。

8.4 Harvard 系列

Harvard 系列格式用不同的方法实现了同样的效果。对应于`\bibitem`,它们使用如下形式:

```
\harvarditem[Jones et al.]{Jones, Baker, and  
Williams}{1990}{jon90}...
```

这是由 Peter Williams 和 Thorsten Schnier 编写的 `harvard.sty` 宏包。它定义了`\harvarditem`命令以及括号引用命令`\cite`和文本引用命令`\citeasnoun`。如果在`\harvarditem`命令中设定好参数,在第一次引用时使用完整的作者列表,随后的引用使用缩略的作者列表。

Harvard 系列的参考文献格式包括:

```
agsm, dcu, kluwer
```

这个宏包已经更新到适用于 \LaTeX 2_ϵ ,并增加了一些易用的功能,使之成为功能强大的宏包,能满足众多引用需要。(它不能抑制重复作者信息,然而,`natbib`宏包可以。)

8.5 Astronomy 格式

显然, Oren Patashnik 意识到了他的 `apalike` 格式的限制性,继续开发出真正的‘apa’参考文献格式,使用了已经在天文学期刊使用的技术。这实际上很像 `newapa` 格式的引用标签,但只有缩略作者列表:


```
\bibitem[\protect\astroncite{Jones et al.}{1990}]{jon90}...
```

它需要调用 `astron.sty` 或其他适当定义了 `\astroncite` 命令的格式文件。

Astronomy 系列的参考文献格式包括：

```
apa, astron, bbs, cbe, humanbio, humannat, jtb
```

这就像 `\citeauthoryear` 命令一样好用，尽管缺少了一些灵活性，因为它缺少了完整的作者列表。

8.6 作者日期格式

最后，我发现了一些使用了称为 `\citename` 的标签命令的宏包，形如

```
\bibitem[\protect\citename{Jones et al., }1990]{jon90}...
```

这不是一种好的实现方式，作者列表和年份没有作为独立参数清晰地分开，因为在标签文本中包含了标点。最好是将标点完全去除，而将其作为 `.sty` 文件中定义的一部分，从而实现完全的灵活性。

沿袭这种风格的参考文献格式包括：

```
authordate1, authordate2, authordate3, authordate4, aaai-named
```

以及相应的宏包 `authordate1-4.sty` 文件。