

如何使用 IEEEtran 文档类

Michael Shell, *Member, IEEE*, 翻译: L^AT_EXStudio

有问题来这里: wenda.latexstudio.net

摘要—本文介绍在 L^AT_EX 下如何使用 IEEEtran 类制作适合提交给电气和电子学会工程师 (IEEE) 的高质量排版文稿。IEEEtran 类提供了一系列的可选参数来制作会议, 期刊和技术说明 (通信) 论文。本文档使用 IEEEtran 生成在期刊的模板模式下制作而成。

欢迎大家反馈我们翻译问题, 我们及时矫正。

扫一扫, 关注我们微信公众号: L^AT_EX 工作室

<http://www.latexstudio.net>

QQ 群: 91940767 | 淘宝: shop240512713.taobao.com



精致您的学术成果, 助力您的科研工作

| 书籍期刊排版 | 简历排版 | 培训咨询 | 模板定制 | 矢量绘图 |

公众号回复: 免费视频 或 入门电子书, 获取精品入门教程和 7G 免费视频。一键获取, 轻松入门。

Index Terms—IEEE, IEEEtran, journal, L^AT_EX, paper, template.

I. 简介

使用最新的 IEEEtran 类文件, 一个装好 L^AT_EX 的计算机以及对 L^AT_EX 语言的基本理解, 作者就可以快速, 不费力气地制作专业品质的排版研究论文。本文的目的是指导用户使用 IEEEtran 类所特有的功能和使用方法。

本文适用于 IEEEtran.cls 1.8b 版本及以后更新版本。之前的版本可能有些功能是缺失的。具体版本是在

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see <http://www.michaelshell.org/contact.html>).

J. Doe and J. Doe are with Anonymous University.

Manuscript received August 11, 2019; revised August 14, 2019.

L^AT_EX 编译的时候会在用户控制台提示给用户具体的版本。例如本文编译的版本信息如下:

```
Document Class: IEEEtran 2015/08/26 V1.8b by Michael Shell
-- See the "IEEEtran_HOWTO" manual for usage information.
-- http://www.michaelshell.org/tex/ieeetran/
```

August 26, 2015

最新版本的 IEEEtran 类及其支持文件可以从 IEEE 的网站 [1] 或 CTAN [2] 获得。后一个网站可能有一些其他材料, 例如 beta 测试版和相关文件非 IEEE 用户使用 IEEEtran。请参阅 IEEEtran 主页 [3] 的 IEEEtran 常见问题和最新消息。

与本文配套的文件有¹: bare_conf.tex, bare_jrnl.tex, bare_jrnl_comsoc.tex, bare_conf_compsoc.tex, bare_jrnl_compsoc.tex 和 bare_jrnl_transmag.tex 这分别是对应的是 conference, journal, IEEE Communications Society journal, IEEE Computer Society conference, IEEE Computer Society journal 和 IEEE TRANSACTION ON MAGNETICS 的示例。作者可以通过使用这些基础示例文件作为自己编写论文的起点, 快速制作成一个功能文档。可以在 bare_adv.tex 找到更完整的示例, 其中包括使用可用宏包以及更复杂的使用方法。

阅读本文档前提是假设作者具备基本的 L^AT_EX 基础知识。对于缺乏基础知识的用户, 强烈建议其阅读参考文献 [4], [5], [6]。尤其是, Tobias Oetiker 的《The Not So Short Introduction to L^AT_EX 2_ε》[5], 介绍了 L^AT_EX 整体使用上的基本知识, 其中文版本在这里下载: Stefan M. Moser 的《HOW TO TYPESET EQUATIONS IN L^AT_EX》还重点介绍了使用 IEEEtran 的 IEEEeqnarray 命令输出 IEEE-风格的公式。这两本书都是可以在线免费获得。

可以在 Internet 新闻组 comp.text.tex 中获得对 L^AT_EX 相关问题的一般支持。还有一个关于 L^AT_EX 常见问题的可搜索列表 [7]。请注意, 附录部分介绍了安装 IEEEtran 类文件的信息以及如何处理使用过程中的常见错误。

¹请注意, 本文档的惯例是不用连字符命令或文件名并以打字机字体显示它们。在这样的结构中, 空格不会隐含在换行符中, 而是显式地进入下一行的开头。此行为不是 IEEEtran 的功能, 但在此用于逐字说明计算机命令。

II. 模板类选项

有许多类选项可用于控制 IEEEtran 文档类的模式, 这都和一般 L^AT_EX 类使用方式是一样的。例如,

```
\documentclass[9pt,technote]{IEEEtran}
```

这是用于通信/简介/技术说明文件的选项。后面我将对这些选项逐一进行说明。对于每个类别选项, 默认选项都以粗体显示。如果默认的选项不能满足需求, 用户必须指定适合自己的选项。各种选项之间是完全独立的——其中一个类别的更改不会影响其他参数的默认值。

A. 9pt, 10pt, 11pt, 12pt

文本字体大小有 4 个类型可选。绝大多数论文使用 10pt。值得注意的例外是技术文件, 它使用 9pt 文本, 最初提交给一些会议的时候使用 11pt 字体大小。

值得注意的是, IEEE 计算机协会出版物使用的字体大小单位是“PostScript”(即“大点”, bp)点大小(即 72bp = 1in)而不是传统排版的点(即 72.27pt = 1in)。此外, “10pt” IEEE 计算机学会期刊论文实际上使用了一个略小的 9.5bp 字体大小(可能是为了弥补 Palatino 字体略宽的特性)。IEEEtran 将根据模式自动调整所选的字体大小。

B. draft, draftcls, draftclsnofoot, final

IEEEtran 提供三种草稿模式以及终稿模式。草稿模式提供更大(双)行间距, 以允许编辑注释以及纸张四边的一英寸边距。标准草案选项将文档中使用的每个包放入草稿模式。对于大多数图形包, 这具有禁用图像渲染的效果。如果不希望这样, 可以使用 draftcls 选项来产生将在 IEEEtran 类中定义的草稿模式, 以便将图像正常包含在内。draftclsnofoot 就像 draftcls, 但不显示单词“DRAFT”以及每页脚下的日期。draft 和 draftclsnofoot 模式都意味着 draftcls(这是其他两个的子集)。使用其中一种草稿模式时, 大多数用户还希望选择 onecolumn 选项。

C. conference, journal, technote, peerreview, peerreviewca

IEEEtran 提供五种主要模式, 包括会议, 期刊, 通信(简要/技术)和同行评审论文。期刊和技术模式将产生非常类似于许多 IEEE TRANSACTIONS 期刊中出现的论文。使用 technote 时, 大多数用户还应选择 9pt 选项。peerreview 模式非常类似于日志模式, 但会生成单列封面(带有标题, 作者姓名和摘要)以便于匿名同行评审。在封面页后的第一页上重复标题(没有作者姓名或摘要)。² 使用同行评审选项的论文

²使用 twoside(双面打印)选项时, 可以在封面页后插入空白页, 以使封面背面不显示纸张的开头。

需要 \IEEEpeerreviewmaketitle 命令(除了传统的 \maketitle 之外)在封面要结束的地方执行, 通常就在摘要之后。使用非 peerreview 模式将以静默方式忽略此命令。有关此命令放置的示例, 请参阅裸模板文件。peerreviewca 模式类似于 peerreview, 但允许输入和格式化作者姓名信息, 如在会议模式中所做的那样(有关详细信息, 请参阅 IV-B2), 以便编辑更加可见作者联系和联系信息。

1) Conference 模式的细节: Conference 模式会使 IEEEtran 行为方式的重大变化数量:

- 随着文本高度降低到约 9.25 英寸, 边距增加。特别是由于 IEEE 希望在底部有额外的间隙, 因此底部边距将变得比顶部边距更大。文本高度不会是精确的 9.25 英寸, 但会与正常字体大小略有不同, 以确保列中的行数是整数。
- 标题和页码不会显示在页眉或页脚中。这与对称的水平边距相结合意味着单面和双面选项之间不会有明显的差异。
- \author 文本放置在表格环境中, 以允许作者姓名和从属关系的多列格式。启用了几个命令以促进此格式化(有关详细信息, 请参阅 IV-B2)。
- 作者姓名后的间距减少了, section 名称周围的间距也是如此。
- 特殊纸质通知(如果使用)将出现在作者姓名和标题之间(而不是在期刊之后)。
- 图片的标题居中。
- 禁用以下命令: \thanks, \IEEEPARstart, \IEEEbiography, \IEEEbiographynophoto, \IEEEpubid, \IEEEpubidadjcol, \IEEEmembership 和 \IEEEaftertitletext。如果需要, 可以通过发出命令重新启用它们: \IEEEoverridecommandlockouts。
- 启用了各种提醒(最终版论文)和警告通知。

使用会议模式时, 大多数用户还希望均衡最后一页上的列(参见 XIV 节)。

D. comsoc, compsoc, transmag

这些互斥选项调用特殊模式, IEEEtran 分别通过这些模式产生 IEEE 通信协会, IEEE 计算机协会和 IEEE TRANSACTIONS ON MAGNETICS 的出版物的格式。默认情况下, 这些都不会启用。

1) Comsoc 模式: Comsoc 模式仅影响数学字体, 因此它将更接近 Times Roman 主文本。可以接受 Michael Sharpe 免费提供的 newtxmath 包 [8](推荐版本 1.451, 2015 年 7 月 28 日或更晚)或商业 MathTime [9] 数学字

体(如 `mtpro2.sty`, `mt11p.sty` 或 `mathtime.sty`)。在 `comsoc` 模式下, 如果用户在文档开头没有加载其中一个宏包, `IEEEtran` 将尝试根据系统上可用的内容强制使用它们。

建议的加载程序和 `newtxmath` 的顺序是:

```
\usepackage[T1]{fontenc}% optional
\usepackage{amsmath}
\usepackage{cmintegrals}{newtxmath}
\usepackage{bm}% optional
```

其中 `cmintegrals` 选项 `IEEEtran` 在加载 `newtxmath` 时设置为默认选项, 是为了获得 IEEE 通信协会使用的特定整数符号样式。可选的 `bm` 宏包 [10] 提供了选择性粗体数学符号。请注意, 不需要加载 AMS Math `amssymb.sty` 包 [11], 因为该功能是由 `newtxmath` 和 `MathTime` 构建和提供的。另外, 不要加载 `newtxtext.sty` 包, 因为这样做会改变主文本字体。

a) *Comsoc* 会议模式: *Comsoc* 会议论文目前的工作方式与传统会议论文 (`bare_conf.tex`) 相同, 因此不需要额外的示例文件。除非正在提交的会议另有明确指示, 否则不要在会议论文中使用 `comsoc` 选项。

2) *Compsoc* 模式: 值得注意的 *compsoc* 模式格式功能包括:

- 默认文本字体从 Times Roman 更改为 Palatino/Palladio (仅限非会议 *compsoc* 模式);
- 修改边距;
- 阿拉伯数字作为节的编号;
- 启用 `\IEEEcompsocitemizethanks` 和 `\IEEEcompsocthanksitem` 命令以提供用于作者从属关系的 `\thanks` (第一个脚注) 逐项列表;
- 启用 `\IEEEtitleabstractindextext` 命令以提供单栏摘要和索引术语 (参见 V 节);
- 各种其他样式更改 (大多数仅适用于非会议 *compsoc* 模式), 例如标题页眉等使用衬线 (Helvetica) 字体; 第一个脚注区域上方带横线; 左对齐参考标签等。

a) *Compsoc* 会议模式: `IEEEtran` 遵循 IEEE 计算机协会会议论文的指导原则。也许令人惊讶的是, 这种格式使 *compsoc* 期刊的许多独特功能无效, 并且与传统会议模式没有太大区别。但是, 保留了阿拉伯语数字作为节的编号。应该提到的是, Scott Pakin 的 `IEEEconf LATEX` 类文件 [12] 也产生了这种格式。请注意, 许多 IEEE 计算机协会会议使用传统会议格式, 并且不应使用 *compsoc* 模式。

3) *Transmag* 模式: 对于 *transmag* 模式:

- `\author` 中的文本应作为会议模式下的长格式输入;

- 启用 `\IEEEtitleabstractindextext` 命令以提供单栏摘要和索引术语 (参见 V 节);
- `\IEEEauthorrefmark` 将产生阿拉伯数字的从属符号;
- 节和小节标题它们的间距略有不同;
- 标题使用比正常情况更小的粗体字体。

transmag 模式 (以及标准日志模式) 也可以提交给 IEEE *Magnetics Letters*。希望在论文末尾出现图片和表格的作者可以使用 `endfloat.sty` [13] 包来实现这一目标。

E. *letterpaper, a4paper, cspaper*

`IEEEtran` 完全支持美国信函 (8.5 英寸 × 11 英寸) 和 A4 纸张尺寸 (210 毫米 × 297 毫米)。由于 IEEE 主要使用美国信函, 因此作者通常应在将作品提交给 IEEE 之前选择信纸选项, 除非另有说明 (通常是在美国境外举行的会议)。在标准日记和会议模式下更改纸张大小不会改变文档的排版, 只会影响页边距。特别地, 使用 `a4paper` 选项的文档将减少侧边距 (A4 比美国信函窄) 和较长的底边距 (A4 比美国信函长)。对于这两种情况, 顶部边距将相同, 文本将水平居中。

对于 *compsoc* 会议和草稿模式, 边距将保持不变, 因此文本区域大小将随纸张大小的变化而变化。

`cspaper` 选项是 IEEE 计算机学会期刊实际出版物中使用的特殊“修剪”纸张尺寸 (7.875 英寸 × 10.75 英寸)。在 *compsoc* 日志模式下, 此选项不会更改文档的排版。只有在他们提交的特定期刊的编辑要求时, 作者才应调用此选项。

请注意, 作者应确保所有后处理 (PS, PDF 等) 使用与 `.tex` 文档相同的纸张规范。这里的问题是迄今为止边距不正确的首要原因。有关详细信息, 请参阅附录 B 节。

对于特殊的 *cspaper* 纸张尺寸, 请注意尽管 `IEEEtran` 会自动为 `pdfLATEX` 的 PDF 模式 (适用于所有纸张尺寸), `dvips` (用于 DVI 到 PS 转换的应用程序) 系统配置正确的纸张尺寸除非在 `dvips` 的 `config.ps` 配置文件中有这样的条目, 否则不会识别特殊的“*ieeecs*”文件:

```
% Special paper size for the IEEE Computer Society J
ournals
@ ieeecs 7.875in 10.75in
@+ ! %%DocumentPaperSizes: ieeecs
@+ %%BeginPaperSize: ieeecs
@+ /setpagedevice where
@+ { pop << /PageSize [567 774] >> setpagedevice }
@+ if
@+ %%EndPaperSize
```

如果未明确要求转换过程使用不同的特定纸张尺寸, 则大多数现代 PS 到 PDF 转换软件将正确处理此类自定义纸张尺寸。

F. *oneside,twoside*

这些选项控制布局是否遵循单面或双面打印的布局。因为侧边缘通常居中, 所以主要的显著差异在于运行标题的格式。

G. *onecolumn,twocolumn*

这些选项允许用户在单栏还是双栏文本格式之间进行选择。由于 IEEE 始终使用双栏文本, 因此 *onecolumn* 是在草稿中才用的。

H. *romanappendices*

IEEEtran 默认按字母顺序编号附录 (例如, A, B 等)。调用此选项以获取罗马编号。

I. *captionsoff*

调用此选项将禁止在图表和表格中显示 `caption`。这样做是为了在 `\caption` 中保留 `\label` 的操作。此选项适用于期刊如 IEEE TRANSACTIONS ON POWER ELECTRONICS(TPE), 它要求在文档末尾的页面上放置无标题的图形和表格。这种图片放置可以在 `endfloat.sty` 宏包 [13] 的帮助下实现:

```
\usepackage[nomarkers]{endfloat}
```

请注意, TPE 还有其他不寻常的格式要求, 这些要求还需要 `draftclassnofoot` 和 `onecolumn` 选项以及在第一部分和参考书目之前插入分页符 (`\newpage`)。这些命令可以通过 `\ifCLASSOPTIONcaptionsoff` 有条件地启用 (III-A)。

J. *nofonttune*

IEEEtran 通常会将默认的字间距更改为 IEEE 出版物中使用的间距。结果是文本需要较少的连字并且通常看起来更美观, 尤其是对于双栏文本。`nofonttune` 选项将禁用这些字体参数的调整。只有那些使用专门设计或修改用于双栏工作的字体的人才应该对此选项感兴趣。

III. CLASSINPUT, CLASSOPTION 和 CLASSINFO 的控制

IEEEtran 提供了三个特殊命令类别, 允许在类文件和用户文档之间传递信息:

- `CLASSINPUTs` 是输入, 提供了一种通过覆盖某些默认设置 (加载 IEEEtran 时) 来自定义 IEEEtran 操作的方法;

- `CLASSOPTIONs` 是允许根据选择的 IEEEtran 类选项进行条件编译的输出;
- `CLASSINFOs` 是允许用户访问有关 IEEEtran 运行时环境的其他信息的输出。

A. *CLASSINPUTs*

可用的 `CLASSINPUTs` 包括:

- `\CLASSINPUTbaselinestretch` 用于设置文档的行间距;
- `\CLASSINPUTinnersidemargin` 设置内部 (装订) 边缘的边距;
- `\CLASSINPUToutersidemargin` 设置外边缘的边距;
- `\CLASSINPUTtopt\extmargin` 设置上边距;
- `\CLASSINPUTbotto\mtextmargin` 设置底部边距。

当然, 这些参数可以通过传统的 LATEX 接口 (`\oddsidemargin`, `\topmargin` 等) 设置。但是, 使用 `CLASSINPUT` 方法的优点是它允许 IEEEtran 调整其他内部参数并根据需要执行任何其他计算。例如, 在 LATEX 中设置边距需要仔细设置 `\oddsidemargin`, `\evensidemargin` 和 `\textwidth`, 还有是否双面打印。

要调用 `CLASSINPUT`, 只需在加载 IEEEtran 之前根据需要定义相关性 `CLASSINPUT`。例如:

```
\newcommand{\CLASSINPUTinnersidemargin}{17mm}
\documentclass{IEEEtran}
```

将产生一个具有 17mm 页边距的文档。如果只指定了一个内部/外部 (或 `toptext/bottomtext`) 边距对, IEEEtran 将假定用户想要对称边距 (或顶部/底部) 并将设置两个值相关对到 (单个) 用户指定的值。

IEEEtran 分别对 `\headheight` 和 `\headsep` 使用 12pt 和 0.25in 的固定值。加载 IEEEtran 后, 可以更改标题的位置, 只要保证 `\topmargin`, `\headheight` 和 `\headsep` 的总和不变, 就不会更改边距。例如, 标题可以使用以下方式向上移动 0.2in:

```
\addtolength{\headsep}{0.2in}
\addtolength{\topmargin}{-0.2in}
```

同样, 可以很容易地更改 `\footskip`, 其默认值为 0.4in, 以改变页脚在底部边距内的位置。

使用 `\CLASSINPUTbaselinestretch` 时, IEEEtran 将自动 “数字化” `\textheight`, 以便整数的行适合页面 (就像在草稿模式中一样)。通过 `CLASSINPUT` 设置顶部或底部边距时, 不会进行数字化。用户应注意, 使用 `CLASSINPUT` 控件可能会导致文档不符合 IEEE 标准。预期的应用包括:

- 具有不寻常格式要求的会议或社团;

- 制作非标准边距的复印件，例如在个人装订使用；
- 非 IEEE 相关工作。

B. CLASSOPTIONS

CLASSOPTIONS 主要是根据正在使用的 IEEEtran 选项 TeX \if 条件自动设置的。因此，例如，诸如的构造

```
\ifCLASSOPTIONconference
\typeout{in conference mode}
\else
\typeout{not in conference mode}
\fi
```

可用于提供条件代码执行。请注意，如 II-B 所述，draft 和 draftsnofoot 选项意味着 draftcls。因此，大多数用户都希望测试 \ifCLASSOPTIONdraftcls 来检测草稿模式。

对于文档的磅值选项，\CLASSOPTIONpt 被定义为一个宏，它扩展到所选点值的数字部分 (例如 9,10,11 或 12)。对于纸张尺寸选项，\CLASSOPTIONpaper 将是包含纸张规格的宏 (例如 letter,a4)。要将这些用作条件，需要进行字符串宏比较：

```
\newcommand{\myninestring}{9}
\ifx\CLASSOPTIONpt\myninestring
\typeout{document is 9pt}
\fi
```

用户应将 CLASSOPTIONS 视为“只读”，而不是尝试手动更改其值，因为 IEEEtran 在内部将它们用作标志以确定选择了哪些选项，更改这些标志可能会导致格式不正确。

C. CLASSINFOS

可用的 CLASSINFOS 包括 \ifCLASSINFOpdf，它与 Heiko Oberdiek 的 ifpdf.sty 宏包 [14] 非常相似，表明 PDF 输出 (来自 pdfLaTeX) 是否有效：

```
\ifCLASSINFOpdf
\typeout{PDF mode}
\fi
```

IEEEtran.cls 还提供了长度为 \CLASSINFOnormalsizebaselineskip，它是 normalsize 字体的 \baselineskip，以及 \CLASSINFOnormalsizeunitybaselineskip，它是 unity \baselinestetch 下 normalsize 字体的 \baselineskip。

最后，有字符串宏 (这些不是条件或长度) \CLASSINFOpaperwidth 和 \CLASSINFOpaperheight，其中包含原始规格中的纸张尺寸，包括单位 (例如 8.5 英寸，22 毫米等)。与 CLASSOPTIONS 一样，用户不应尝试更改 CLASSINFO。

IV. 标题页

标题区域特有的文档部分是使用标准 TeX 命令 \maketitle 创建的。在调用此命令之前，作者必须声明将出现在标题区域中的所有文本对象。

A. 文章标题

论文标题声明如下：

```
\title{A Heuristic Coconut-based Algorithm}
```

以标准 TeX 的方式。标题通常是大写的，除了 a, an, and, as, in, but, by, for, in, no, on, to, to, to 通常不是大写，除非它们是标题的第一个或最后一个字。换行符 (\) 可用于均衡标题行的长度。不要在标题中使用数学符号或其他特殊符号。

B. 作者姓名

使用 \author 命令声明姓名和相关信息。不同的文档模式 \author 的行为略有不同。

1) *Journal/Technote* 模式中的名称: 期刊或技术文件的典型 \author 命令看起来像这样：

```
\author{Michael~Shell,~\IEEEmembership{Member,~IEEE,}
↪ John~Doe,~\IEEEmembership{Fellow,~OSA,}
↪ and~Jane~Doe,~\IEEEmembershi{Life~Fellow,~IEEE}}%
\thanks{Manuscript received January 20, 2002; revised August
↪ 26, 2015. This work was supported by the IEEE.}%
\thanks{M. Shell was with the Georgia Institute of
↪ Technology.}}
```

\IEEEmembership 命令用于生成斜体字体，指示作者的 IEEE 成员资格状态。 \thanks 命令产生“第一个脚注”。因为 TeX \thanks 不是为了包含多个段落³，所以作者必须对每个段落使用单独的 \thanks。但是，如果需要，可以在 \thanks 中使用常规换行符 (\)。为了获得正确的换行符和间距，正确使用和控制 \author 中的空格非常重要。使用不间断空格 (~) 确保名称/成员对保持在一起。一个小而容易犯的错误就是忘记在使用分隔 (}) 参数的命令之间进行防止不必要的空间。注意这两个 % 用于防止以结尾的行上的代码行中断成为不需要的空白。这样的空白不会被视为行尾空白，因为从技术上讲，最后的 \thanks 是行上的最终命令。像这样的“幻影”空格会附加到最后一个作者姓名的末尾，导致其他居中的名称行稍微向左移动。

2) *Conference* 模式中的名称: 在会议模式下，作者姓名区域更复杂，因为它还包含作者的隶属关系。因此，在会议模式下，\author\{} 的内容将被放入已修改的表格环境中。还提供了命令 \IEEEauthorblockN\{} 和 \IEEEauthorblockA{}，以便分别正确地格式化作者姓

³尽管 IEEEtran.cls 确实支持它，但标准类文档却不支持它。

名和从属关系。对于具有三个或更少隶属关系的论文，首选多列格式：

```
\author{\IEEEauthorblockN{Michael Shell}
\IEEEauthorblockA{School of Electrical and\
Computer Engineering\
Georgia Institute of Technology\
Atlanta, Georgia 30332--0250\
Email: mshell@ece.gatech.edu}
\and
\IEEEauthorblockN{Homer Simpson}
\IEEEauthorblockA{Twentieth Century Fox\
Springfield, USA\
Email: homer@thesimpsons.com}
\and
\IEEEauthorblockN{James Kirk\
and Montgomery Scott}
\IEEEauthorblockA{Starfleet Academy\
San Francisco, California 96678-2391\
Telephone: (800) 555--1212\
Fax: (888) 555--1212}}
```

使用 `\and` 分隔从属关系列。列将自动相对于彼此和侧边距居中。

如果有超过三位作者或文本太宽而无法跨页面，请使用备用的长格式：

```
\author{\IEEEauthorblockN{Michael Shell\IEEEauthorrefmark{1},
↪ Homer Simpson\IEEEauthorrefmark{2}, James
↪ Kirk\IEEEauthorrefmark{3}, Montgomery
↪ Scott\IEEEauthorrefmark{3} and Eldon
↪ Tyrell\IEEEauthorrefmark{4}}
\IEEEauthorblockA{\IEEEauthorrefmark{1}School of Ele
ctrical and Computer Engineering\
Georgia Institute of Technology, Atlanta, Georgia 30
332--0250\
Email: mshell@ece.gatech.edu}
\IEEEauthorblockA{\IEEEauthorrefmark{2}Twentieth Cen
tury Fox, Springfield, USA\
Email: homer@thesimpsons.com}
\IEEEauthorblockA{\IEEEauthorrefmark{3}Starfleet Academy, San
↪ Francisco, California 96678-2391\
Telephone: (800) 555--1212, Fax: (888) 555--1212}
\IEEEauthorblockA{\IEEEauthorrefmark{4}Tyrell Inc.,123
↪ Replicant Street, Los Angeles, California 90210--4321}}
```

`\IEEEauthorrefmark{}` 命令将生成与其参数中的数字对应的脚注符号。使用此链接将作者姓名链接到各自的附属机构。没有必要阻止位于 `\IEEEauthorblock` 之间的空格，因为每个块都会启动一组新的行，而 \LaTeX 将忽略行的最后和开头的空格。

3) *Compsoc* 期刊模式中的姓名: IEEE 计算机学会期刊的一个独特特征是作者从属关系在第一个 (`\thanks`) 脚注中用列表格式化。在 *compsoc* 模式下, *IEEEtran* 提供了一种特殊形式的 `\thanks`, 即 `\IEEEcompsocitemizethanks` 以获得此效果:

```
\author{Michael~Shell,~\IEEEmembership{Member,~IEEE,}
↪ John~Doe,~\IEEEmembership{Fellow,~USA,}
↪ and~Jane~Doe,~\IEEEmembership{Life~Fellow,~IEEE}}%
\IEEEcompsocitemizethanks{\IEEEcompsocthanksitem M.Shell is
↪ with the Georgia Institute of Technology.
```

```
\IEEEcompsocthanksitem J. Doe and J. Doe are with Anonymous
↪ University.}%
\thanks{Manuscript received January 20, 2002; revised August
↪ 26, 2015.}}
```

在 `\IEEEcompsocitemizethanks` 中, `\IEEEcompsocthanksitem` 的工作方式与 `\item` 类似, 以提供项目符号的联属组。为了便于双重编译, 在非 *compsoc* 模式下, *IEEEtran* 将 `\IEEEcompsocitemizethanks` 视为 `\thanks` 并设置 `\IEEEcompsocthanksitem` 以生成带缩进的换行符。然而, 这并不完全令人满意, 因为 IEEE 计算机学会期刊将作者附属机构放在“手稿收到”行之前, 而传统的 IEEE 期刊使用相反的顺序。如果需要正确的双重编译, 可以使用 `CLASSOPTION` 条件来根据需要交换顺序。

4) *Compsoc* 会议模式中的姓名: *compsoc* 会议模式中的名称与传统会议模式相同。

5) *Transmag Journal Mode* 中的名称: IEEE TRANSACTIONS ON MAGNETICS 论文通常使用会议长格式, 但尝试将每个姓名和地址对保留在一行, 而不使用任何电子邮件地址或电话号码。此外, `\thanks` 在 *transmag* 期刊模式下可用, 即使输入的名称很像会议模式下的长格式。有关 *transmag* 模式下作者条目的示例, 请参阅文件 `bare_jrn1_transmag.tex`。

C. 流水标头

使用 `\markboth{}{}` 命令声明流水标头。第一个参数包含期刊名称信息, 第二个参数包含作者姓名和论文标题。例如:

```
\markboth{Journal of Quantum Telecommunications,~Vol.~1,
↪ No.~1~January~2025}{Shell \MakeLowercase{\textit{et al.}}:
↪ A Novel Tin Can Link}
```

请注意, 由于流水标头中的文本会自动大写, 因此必须使用 `\MakeLowercase{}` 命令来获取小写文本。第二个参数用作页面标题, 仅用于双面期刊论文标题页之后的奇数页。这个页面就是这样一个例子。技术论文不需要第二个参数。会议论文没有流水标头, 因此 `\markboth{}{}` 在会议模式下无效。作者不应在匿名同行评审文章的标题 (如果使用) 中加入任何姓名信息。

D. 出版物 ID 标记

出版物 ID 标记可以通过 `\IEEEpubid{}` 命令放在期刊和技术论文的标题页上:

```
\IEEEpubid{0000--0000/00\$\$00.00-\copyright-2015 IEEE}
```

尽管作者在提交论文时尚未拥有有效的出版物 ID, 但 `\IEEEpubid{}` 非常有用, 因为它提供了一种方法, 可以查看最终出版物中有多少标题页文本区域不可用。这在技术期刊中尤其重要, 因为在某些期刊中, 发布 ID

空间可能会占用多个文本行。如果使用 `\IEEEpubid{}`，则必须在标题页的第二列中的某处发出第二个命令 `\IEEEpubidadjcol`。这是必需的，因为 \LaTeX 会重置每列开头的文本高度。`\IEEEpubidadjcol` “拉起”第二列中的文本，以防止它盲目地运行到发布 ID。

作者不会在就绪会议论文上放置出版物 ID，因此 `\IEEEpubid{}` 在会议模式下被禁用。相反，在会议模式下，`IEEEtran` 会自动增加底部边距，以便在发布时为此类标记提供 IEEE 空间。在草稿模式下，发布者 ID 标记不会打印在标题页的底部，但会为其清除空间。

出版商 ID 标记对于 `compsoc` 论文可能不太重要，因为 IEEE 计算机学会期刊将出版商 ID 标记放在底部边距内，以免影响可用于文本的页面空间量。

E. 特殊文章告知

特殊文章告知，例如邀请文章，可以声明：

```
\IEEEspecialpapernotice{(Invited Paper)}
```

期刊和技术文章中的特殊文章告知出现在作者姓名和正文之间。本文档的标题页有一个例子。对于会议论文，特殊文章告知放在标题和作者姓名之间。

更为罕见的是，有时需要在主文本正上方的两列上更多空格。例如，一篇论文可能有献辞 [15]。`IEEEtran` 提供了命令 `\IEEEaftertitletext{}`，可用于插入文本或更改标题区域和主文本之间的间距：

```
\IEEEaftertitletext {\vspace{-1\baselineskip}}
```

作者应该知道 `IEEEtran` 仔细计算标题区域和主文本之间的间距，以确保第一页的主文本高度始终等于正常大小的整数行（除非顶部或底部边距已被 `CLASSINPUTs` 覆盖）。如果在该列中没有任何弹性长度（例如章节标题周围的那些），不这样做可能会导致第一页的第二列中的不饱满的 `vbox` 错误和段落被“拉开”。有意允许 `\IEEEaftertitletext{}` 的内容绕过这个“动态确定的标题间距”机制，因此作者可能必须手动调整 `\IEEEaftertitletext{}` 内容（如果使用）的高度（几个点）避免不完整的 `vbox` 警告。

V. 摘要和关键词

摘要通常是 `\maketitle` 之后的论文的第一部分。摘要文本放在 `abstract` 环境中：

```
\begin{abstract}
We propose...
\end{abstract}
```

通常不应在摘要中使用数学，特殊符号或引用。⁴

⁴也就是说，如果允许或需要，请注意，为了保持构造（如矢量和标量形式）之间的区别，`IEEEtran` 默认在摘要中使用非粗体数学符号。但是，粗体数学符号号更好地匹配用于摘要文本的粗体文本字体。如果需要粗体数学字体，只需在摘要的开头发出 `\boldmath` 命令。

期刊和技术文章还有一个关键词（索引词）列表，可以用以下方式声明：

```
\begin{IEEEkeywords}
Broad band networks, quality of service, WDM.
\end{IEEEkeywords}
```

要从 IEEE 获取有效关键字列表，只需发送一封空白电子邮件至 `keywords@ieee.org`。可以从 `http://www.computer.org/mc/keywords/keywords.htm` 获取 IEEE 计算机协会批准的关键字列表。不要在关键字中使用数学或特殊符号。

IEEE 计算机学会和 IEEE TRANSACTIONS ON MAGNETICS 格式存在一个困难，即 `compsoc` 和 `transmag` 期刊（但不是 `compsoc` 会议）论文将摘要和关键词部分放在作者名称下面的单列格式中，但其他 IEEE 格式放置它们在第一部分之前的正文的第一列。为了解决这个问题，`IEEEtran` 提供了一个命令，`\IEEEtitleabstractindextext`，它将在 `\maketitle` 之前声明，其单个参数包含在作者名称后以单列格式显示的文本/节：

```
\IEEEtitleabstractindextext{%
\begin{abstract}
We propose ..
\end{abstract}
\begin{IEEEkeywords}
Broad band networks, quality of service, WDM.
\end{IEEEkeywords}}
```

为了便于双重编译，`IEEEtran` 提供了另一个命令 `\IEEEdisplaynontitleabstractindextext`，它将“成为”在非 `compsoc`，非 `transmag` 或会议模式下在 `\IEEEtitl\abstractindextext` 中声明的任何内容（因为 `compsoc` 会议使用与传统会议相同的摘要和关键词的位置）。也就是说，摘要和关键词部分可以根据文档模式自动“远程传送”到他们需要的适当位置。`\IEEEdisplaynontitleabstractindextext` 通常应放在 `\maketitle` 之后（如果使用的话，在 `\IEEEpeerreviewmaketitle` 之前）。

VI. SECTIONS

章节及其标题通过 `\section`，`\subsection`，`\subsubsection` 和 `\paragraph` 以通常的 \LaTeX 方式声明。在 `noncompsoc` 模式中，这些部分的编号分别为大写罗马数字，大写字母，阿拉伯数字和小写字母。在 `compsoc` 模式下，阿拉伯数字专门用于 (sub)section 编号。

技术文章或 `compsoc` 会议不允许使用 `\paragra`，因为它通常不允许具有如此深的嵌套深度。如果需要，可以通过在文档前言中发出命令 `\setcounter{\secnumdepth}{4}` 来恢复 `\paragraph`。

请注意, IEEE 计算机协会期刊 (但不是会议!) 是不寻常的, 因为它们将第一节开头的文本 (引言) 提升。IEEEtran 提供了产生此效果的命令:

```
\IEEEraisesectionheading{\section{Introduction}}
\label{sec:introduction}
```

除 compsoc 期刊模式中的介绍部分外, 此命令不适用于任何其他用途。请注意, 需要保留 \label, 以引用上面 \section 之后的部分, 作为 \IEEEraisesectionheading 把 \section 在一个盒子中。

A. 首字母下沉

期刊论文的第一个字母是较大的大写字母, 它在基线以下一行下降。这封信被称为“下沉”字母。第一个单词中的其他字母以大写字母呈现。使用 IEEEtran 命令 \IEEEPARstart{}{} 可以准确地生成此效果。第一个参数是第一个单词的第一个字母, 第二个参数包含第一个单词的剩余字母。本文件的下沉是通过以下方式生成的:

```
\IEEEPARstart{W}{ith}
```

请注意, 某些期刊也会以大写形式呈现第二个单词, 特别是如果第一个单词非常短。有关更多用法示例, 请参阅 bare_jrn1.tex 示例文件。

VII. 引用

引用是像往常一样使用 \cite 命令创建的。IEEEtran 将生成以 IEEE 格式单独括起来的引用号。(“[1], [5]” 而不是更常见的 “[1,5]” 形式。) 当有三个或更多相邻引文数时, 基本 IEEEtran 不排序或产生压缩的“范围”。但是, IEEEtran 预先定义了一些格式控制宏, 以便于使用 Donald Arseneau 的 cite.sty 包 [16]。所以, 作者所要做做的就是引用 cite.sty :

```
\usepackage{cite}
```

并且相邻的引用号将自动进行分类和压缩 (范围) IEEE 样式。(当然, 多个相邻的引用应该总是在单个 \cite 中声明, 以逗号分隔, 这样才有效。) 调用 cite.sty 的 noadjust 选项, 以防止在引用需要包含在内时出现不必要的前导空格插入语。

compsoc 模式中的一个复杂因素是 IEEE 计算机协会不会压缩相邻的引用数字, 而是对它进行排序。cite.sty 4.0 及更高版本提供了一个禁用压缩的 nocompress 选项, 但保留了排序。从而,

```
\ifCLASSOPTIONcompsoc
% requires cite.sty v4.0 or later (November 2003)
\usepackage[nocompress]{cite}
\else
\usepackage{cite}
\fi
```

可以普遍适用。

请注意, 如果需要 (例如, 在非全角, 非空格字符旁边), cite.sty 的 \cite 命令将自动添加前导空格。即, “(\cite\{mshell01})” 将变为 “([1])”。如果不需要这种行为, 请使用 cite 包的 noadjust 选项 (cite.sty V3.8 及更高版本), 这将关闭添加的空格:

```
\usepackage[noadjust]{cite}
```

\cite 还允许选择注释 (例如, \cite\{Th.7.1\}{mshell01})。如果带有注释的 \cite 有多个引用, 则该注释将应用于列出的最后一个引用。通常希望如果给出一个注释, 那么只应该在该引用中列出一个引用。

VIII. 公式

使用传统的 equation 环境创建公式:

```
\begin{equation}
\label{eqn_example}
x=\sum\limits_{i=0}^z{z}2{i}Q
\end{equation}
```

产生的公式是:

$$x = \sum_{i=0}^z 2iQ \quad (1)$$

如果不需要公式编号, 请使用 displaymath 环境。在提到公式时, IEEE 出版物中的文章通常不使用“equation”一词, 而只是将公式编号括在括号中, 例如, .. as can be seen in(\ref{eqn_example}).

IEEE 的双栏模式严重限制了公式的范围。因此, 构建公式的相当一部分努力通常必须用于正确地分解它们。作者有责任确保所有公式符合给定的列宽。在极少数情况下, 可能会有一些跨栏的公式 (参见 X-D1), 但绝大多数超长公式必须在多行上分开。

IX. 多行公式

也许最方便和最流行的方法来产生多行公式是 L^AT_EX 2_ε 的 eqnarray 环境。然而, eqnarray 有几个严重的缺点:

- 1) 在列分隔空间中使用 2× \arraycolsep 不会在默认配置中提供自然的数学间距;
- 2) 列定义不能改变;
- 3) 它限于三个对齐列;
- 4) 不能在单个单元格内覆盖列对齐。

有许多非常优秀的包用于排版多行公式。也许最受欢迎的是 amsmath 包 [11]。amsmath 是一项综合性工作, 除了增强的多行对齐环境外, 还包含许多有用的工具。因此, 所有作者都应该认真考虑其使用, 无论他们使用什么来生成对齐公式。需要注意的一点是, 在加载

表 1
MATH SPACINGS USED BY LATEX

Size	Width	Cmd.	Used for	Example
small	1/6 em	\,	symbols	ab
medium	2/9 em	\:	binary operators	$a + b$
large	5/18 em	\;	relational operators	$a = b$
negative small	-1/6 em	\!	misc. uses	ab

时, `amsmath` 将配置 LATEX 以禁止多行公式的分页 (即使在非 `amsmath` 定义的环境中)。这里的理念是作者应该在需要的地方手动插入断点, 以确保断点仅在可接受的点发生。要恢复 IEEEtran 在多行公式自动分解的能力, 请加载 `amsmath`, 如:

```
\usepackage{amsmath}
\interdisplaylinepenalty=2500
```

另一套极其强大的对齐工具, 其中一个是完全重写的 `eqnarray` 环境, 由 `mathenv.sty` 提供, 它是 Mark Wooding 的 MDW Tools 的一部分 [17]。

最后, IEEEtran 提供了一个完全集成的自定义 `\IEEEeqnarray` 命令系列 (见附录 F 节), 这些命令几乎可以适用于许多不同类型的对齐情况。

尽管如此, 使用标准的 `eqnarray` 显示一个简单的例子是有益的, 以便解释 LATEX 下数学间距的一些细节。如表 1 所示, LATEX 通常在排版数学公式时从四个不同的间距绘制。为了产生精确 (和正确) 的数学比对, 理解如何控制这种间距是至关重要的。

考虑一个多行公式:

$$\begin{aligned}
 Z &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
 &+ a + b & (2) \\
 &+ a + b & (3) \\
 &+ a + b & (4) \\
 &+ a + b & (5)
 \end{aligned}$$

(以典型的 IEEE 风格) 由生产

```
\setlength{\arraycolsep}{0.0em}
\begin{eqnarray}
Z&{=}{{}&x_1 + x_2 + x_3 + x_4 + x_5 + x_6\nonumber\\
&&+ a + b\\
&&+ a + b\\
&&+ a + b\\
&&+ a + b
\end{eqnarray}
\setlength{\arraycolsep}{5pt}
```

第一到第四行显示了 $+a + b$ 可以实现的一些可能方式。⁵ 对于大多数 IEEE 而言, 只有第四行才是正确的方法。在 LATEX 的数学模式中, 可以通过将它们括在大

⁵在该示例中, 等式编号系统 (ab) 用于识别线。

括号 (例如 `{=}`) 中来强制禁止运算符周围的空格, 或者通过用 “empty ords” (例如 `{}= {}`) 将它们包围来限制它们。重要的是要占位符本身没有宽度。但是, 它们的存在会导致 LATEX 在运算符周围放置空格, 好像它们 “紧挨着某些东西”。考虑到这一点, 示例中的第一步是将 `\arraycolsep` 设置为零以防止 `eqnarray` 放入不需要的, 人为的列间距。在等号周围放置占位符然后强制正确的自然间距。或者 `\arraycolsep` 可能已设置为 0.14em 和等号周围的空格消除了。⁶重要的是, 要记住在 `eqnarray` 完成后将 `\arraycolsep` 恢复到其默认值 5pt, 因为其他环境 (例如 `array`) 依赖于它。(或者, 结构可以包含在一组大括号中以保持更该是局部的, 这具有额外的优点, 即不要求用户记住正确的默认值。)

第一行是不正确的, 因为 a 表示为正数而不是必须添加到前一行的数字。(即 $+$ 被视为一元, 而不是二元运算符。) 在第二行中, 在加号的右侧添加一个占位符什么都不做, 除了证明占位符的宽度为零。在加号的左侧添加一个占位符 (第三行) 确实会使二者间距接合, 但会导致不需要的右移。⁷ 最后, 只在第四行的加号手动添加一个中间空间就可以了。在这种情况下, 不需要抑制加号 (`{+}`) 周围的自动间距, 但在默认情况下 “扩展” 此类运算符的其他对齐环境中可能需要抑制。

间距问题的另一种方法是仅使用两个对齐列 (如 `amsmath.sty` 的 `\align` 所做)。例如, 在前面的例子中, “ $Z =$ ” 将包含在第一列中。

A. 分段结构

顺便说一句, Donald Arseneau 的 `case.sty` 包 [18] 中的 `numcases` (或 `subnumcases`) 环境应该用于 “case” 结构, 其中每个分支可以用不同的公式 (或子公式) 编号引用:

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \\ -x, & \text{for } x < 0 \end{cases} \quad (6)$$

因为从 `array` 或 `amsmath` 案例环境构建的那些将具有包含两个分支的单个等式编号。

请注意, `amsmath` (如果用户没有这样, 将在 `comsoc` 模式下自动加载) 将与 `cases.sty` 一起使用, 后者应该在前者之后加载, 否则会出现错误 “Command `\subequations` already defined”。

X. 浮动结构

在为图片和表格环境选择适当的可选放置参数时, 作者应该记住, 大多数 IEEE 期刊都强烈支持将浮动位

⁶这假设文本字体中的 1em 与数学字体中的 1em 具有相同的宽度。对于标准字体, 情况确实如此。

⁷IEEE 通常希望所有的行保持左对齐, 但是有些情况下可能需要这样的缩进。

置定位到页面顶部，而且很少 (如果有的话) 使用底部浮动。IEEE 计算机学会期刊也赞成浮动体放在顶部，但偶尔会使用底部浮动体。此外，IEEE 期刊从不在第一页的第一列中放置浮点体，并且很少 (如果有的话) 在第一页的第二列中放置浮点体。文本中间放置 (“here”) 通常不用于 IEEE，但有一个值得注意的例外 IEEE 计算机协会会议。

请注意 $\LaTeX 2_{\epsilon}$ 的浮动例来将脚注放在底部浮动上方。要更改这一点，以便脚注出现在底部浮动下方 (如 IEEE 所做的那样)，请按照 Sigitas Tolušis 的 `stfloats` 包 [19] 提供的命令调用 `\fnbelowfloat` (有关 `stfloats` 包的更多功能，请参阅 X-D)。

A. 图片

图片以标准 \LaTeX 方式处理。例如：

```
\centering
\includegraphics [width=2.5in]{myfigure}
\caption{Simulation results for the network.}
\label{fig_sim}
\end{figure}
```

注意：

- 1) 图片应该通过 \LaTeX `\centering` 命令居中，这比使用增加不需要的垂直间距的 `center` 环境更好；
- 2) 标题放在图片下方；
- 3) 任何标签必须在 `caption` 命令之后 (或之内) 声明。

当引用典型 IEEE 论文中的图片时，作者应使用缩写 “Fig”，但在 IEEE 计算机协会会议论文中，他们应使用完整的单词 “Figure”。IEEEtran 提供字符串宏 `\figurename`，其中包含用于给定格式模式的正确名称。

`\includegraphics` 命令是插图的现代首选方式，它提供了一个灵活的界面，可以轻松地将图形缩放到大小。要使用它，必须首先加载 `graphics` 或 `graphicx` (推荐后者)。

强烈建议作者熟悉插图宏包 [20] 以及 Keith Reckdahl $\LaTeX 2_{\epsilon}$ 中的优秀地使用插图 [21]。提醒读者必须选择 `draftcls` 或 `draftclsnofoot` 而不是 `draft` 类选项才能获得带有可见图片的草稿。

如附录 D 节中所述，封装的 PostScript (EPS) 或可移植文档格式 (PDF) 是 \LaTeX 工作的首选图片格式。此外，用户的绘图/绘图应用程序应该能够直接以 EPS (或 PDF) 矢量形式输出 (放大时不会降低或像素化)，尽管照片很可能必须是 (EPS/PDF/JPEG/PNG) 位图形式。请注意，EPS 以外的图像格式需要使用 $\pdf\LaTeX$ 。

`psfrag` 包 [22] 也可能是有用的。`psfrag` 允许用户 “进入” EPS 图片并用真正的 \LaTeX 代码替换其中包

含的文本字符串。通过这种方式， \LaTeX 对数学符号和字体的广泛支持可以扩展到使用具有更适度的字形支持的应用程序所做的图片。使用 `psfrag` 确实需要使用 `dvips` DVI 到 PostScript 转换步骤 (而不是 $\pdf\LaTeX$ 的 PDF 模式)，因为必须使用 PostScript 语言的一些功能。⁸ $\pdf\LaTeX$ 用户可以使用 `psfrag` “预处理” 他们的图片，方法是使用 `psfrag` 将其导入到虚拟文档中，运行 \LaTeX 后跟 `dvips`，然后将 PostScript 输出转换为 PDF 图片，以便直接导入到主文件中然后由 $\pdf\LaTeX$ 处理的文档。

1) 子图：可以通过使用 Steven Douglas Cochran 的 `subfigure` [23] 或 `subfig` [24] 插入图片。提前说明前者不再被维护，虽然自包含且与 IEEEtran 兼容，但与越来越多的其他 \LaTeX 宏包 (包括 `fixltx2e.sty`) 不兼容。因此，不建议将 `subfigure` 用于新工作，此处不予介绍。

值得注意的是，通常需要 `subfig.sty` 包选项才能获得符合 IEEE 标准的子图标题。此外，`compsoc` 格式需要比传统 IEEE 格式中使用的衬线脚注大小字体更大的无衬线字体。`subfig.sty` 还有一个复杂因素，因为这个包依赖于 `caption.sty`，它在默认配置下会覆盖 IEEEtran 对 `caption` 的处理，导致非 IEEE 样式的主要标题。为了防止这种情况，请务必调用 `subfig.sty` 的 `caption=false` 选项，该选项自版本 1.3(2005/06/28) 开始提供。因此，加载 `subfig.sty` 的推荐方法是：

```
\ifCLASSOPTIONcompsoc
\usepackage [caption=false,font=normalsize,labelfont=sf,textfont=sf]{subfig}
\else
\usepackage [caption=false,font=footnotesize]{subfig}
\fi
```

因为多个子图通常需要比单列中更多的宽度，所以它们通常在双栏图形环境中使用 (X-D)：

```
\begin{figure*}[!t]
\centering
\subfloat [Case I]{\includegraphics [width=2.5in]{subfigcase1}}
\label{fig_first_case}}
\hfil
\subfloat [Case II]{\includegraphics [width=2.5in]{subfigcase2}}
\label{fig_second_case}}
\caption{Simulation results for the network.}
\label{fig_sim}
\end{figure*}
```

请注意如何通过 `\subfloat` 命令的可选参数将标题标记到每个子图以及整个图中。但是，大多数 IEEE 作者/期刊不使用子图标题，而是参考/描述主标题内的所有子图 (a), (b) 等。请注意，对于 `subfig.sty` 生成 (a), (b) 等，子图标签必须存在 `\subfloat` 的可选参数。如果不需要子标题，只需将其内容留空 (例如 `\subfloat []`)。`\hfil` 用作子图分隔符，以实现图形周围的相等间距。

⁸PDF 很像 PostScript 的一个子集，后者是图灵完整的编程语言，前者不是。

表 II
A SIMPLE EXAMPLE TABLE

First	Next
1.0	2.0

更复杂的实现是可能的。请注意，一行上所有子图的总宽度必须小于文本宽度，否则会出现不需要的换行符。如果需要，可以在图中使用多行子图。

Axel Sommerfeldt 现代且积极维护的 `subcaption.sty` 包 [25] 目前不建议使用，因为它没有提供防止底层 `caption.sty` 控制主标题格式，远离 IEEEtran 的选项。

B. 算法

IEEE 出版物使用 `figure` 环境来包含不属于主文本流的算法。Peter Williams 和 Rogerio Brito 的算法包 `algorithmic.sty` [26] 或 SzászJános 的算法包 `algorithmicx.sty` [27](后者的设计为比前者更可定制) 可能有助于产生类似算法的结构 (尽管作者当然可以免费使用他们在这方面最熟悉的 L^AT_EX 命令。但是，不要使用 `algorithm.sty` (也是 Williams 和 Brito) 或 `algorithm2e.sty` (作者 Christophe Fiorio) 的浮动算法环境，因为 IEEE 使用的浮动体只有是图片和表格。此外，IEEEtran 将无法控制 `algorithm.sty` 或 `algorithm2e.sty` 浮动环境生成的 (非 IEEE) 标题样式。

C. 表格

表格以类似的方式处理，但有一些显着的差异。例如，代码

```
\begin{table}[!t]
\renewcommand{\arraystretch}{1.3}
\caption{A Simple Example Table}
\label{table_example}
\centering
\begin{tabular}{c|c}
\hline
\bfseries First & \bfseries Next\
\hline\hline
1.0&2.0\
\hline
\end{tabular}
\end{table}
```

结果列于表 II 中。请注意，IEEE 将表格标题放在表格之前，并且它们的样式与标题非常相似，通常大写，除了诸如 a, an, and, as, at, but, by, for, in, of, 之类的单词之外，on, or, to, to 和 up, 它们通常不是大写的，除非它们是标题的第一个或最后一个字。

请注意，为了防止因更改大小写而导致意义发生变化，IEEE 通常在渲染单位时使用标准文本字体，而不是小型大写字体，以及表格标题中的数学字母。这可以通过使用 `\upshape` 来实现

```
\caption{Diagnosis of Rotor Faults in a DRFOC Drive Using the
↪ VCT(Flux Loop Bandwidth (FLB)=10 {\upshape eHz};75%
↪ Load;1450 {\upshape r/min})}
```

感谢 ZhaowenHou 提供有关此主题的信息以及上述示例。

在表格环境中，默认文本大小是 `footnotesize`，这是 IEEE 通常用于表的内容。当使用表格环境构造表格时，通常最好将 `\arraystretch` 的值增加到 1 以上，以“打开”表格行。此外，IEEE 经常使用具有“开放侧”的表 (沿着每一侧没有垂直线)，尽管“封闭侧”形式 (例如，表 I) 更常用于本文档中的表。不幸的是，标准 L^AT_EX 2_ε 表格环境有许多缺点。两个值得注意的问题是

- 1) 线路相交的角落形成不当;
- 2) 在用户控制方面不够灵活。

出于这些原因，敦促作者研究制作表格的其他一些包。为表格和阵列环境提供修订的“直接替换”的好方法是 Frank Mittelbach 和 David Carlisle 的 `array` 包 [28]。更强大 (也更复杂) 的是 `mdwtab.sty` 包提供的表格和数组环境，它是 Mark Wooding 的 MDW Tools [17] 的一部分。

作为替代方案，IEEEtran 提供 `IEEEeqnarraybox` 命令，该命令也可用于生成高质量的表格。⁹有关详细信息，请参阅附录 F。

1) 表格内的脚注：脚注通常不能直接放在某些命令和环境中，例如 `\parbox`, `tabular` 等，因为它们会被“困”在里面。解决这个问题的一种方法是将脚注标记 (`\footnotemark`) 所在的位置 (在表格内) 与脚注文本本身声明的位置分开 (使用 `\footnotetext` 在表格之外)。

另一种方法是使用 `footnote.sty` 包 (它是 Mark Wooding 的 MDW Tools [17] 的一部分)，它允许配置环境以便不捕获脚注：

```
\usepackage{footnote}
\makesavenoteenv{tabular}
```

请注意，在浮动结构 (如表) 中使用脚注可能不是一个好主意，因为每个脚本的位置可以相对于彼此移动。要将脚注放在表格的末尾而不是页面的底部，只需将表格等包含在小型内容中 (不需要 `footnote` 包)。在表格中处理脚注 (包括那些浮动的脚注) 的一个非常好的方法是使用 Donald Arseneau 的 `threeparttable` 包 [29]，它用于生成表 III (其代码是 `threeparttable.sty` 文件中的一个示例)。

⁹表 I 是用这个命令生成的。

表 III
THE SKEWING ANGLES (β) FOR $Mu(H) + X_2$
AND $Mu(H) + HX^A$

	$H(Mu) + F_2$	$H(Mu) + Cl_2$
$\beta(H)$	80.9° ^b	83.2°
$\beta(Mu)$	86.7°	87.7°

^a for the abstraction reaction, $Mu + HX \rightarrow MuH + X$.

^b 1 degree = $\pi/180$ radians .

D. 双栏浮动体

L^AT_EX 的 `figure*` 和 `table*` 环境生成跨栏的图片和表格。对于单栏而言太宽的结构有时需要此功能。

L^AT_EX 2_ε 内核的限制是双栏浮动体不能放在页面的底部。也就是说 `\begin{figure*}[!b]` 通常不会按预期生效。需要此功能的作者应获取并加载 Sigita Tolušis 的 `stfloats` 包 [19], 该宏包修补 L^AT_EX 2_ε 输出例程, 以允许它处理页面底部的双栏浮动体。请注意, `stfloats` 是一个非常激进的宏包, 除了标准的 L^AT_EX 2_ε 版本之外, 它可能无法与 L^AT_EX 版本一起使用, 并且可能会导致修改输出和/或浮动例程的其他宏包出现问题 (例如那些平衡列, 改变浮动体图片的位置等)。IEEE 作者被警告不要使用允许内容放在两个文本列中间的包 (例如 `cuted.sty`, `midfloat.sty` 等)。

另一个 L^AT_EX 2_ε 限制 (使用 `stfloats` 修补或不修补) 是双栏浮动不会出现在定义它们的同一页面上。因此, 用户必须在它们将要 (可能) 出现的页面之前定义这样的事物。

L^AT_EX 2_ε (用 `stfloats` 修补或不修补) 不会尝试将双栏和单列浮动体依次保持在一起。这可以通过加载 Frank Mittelbach, David Carlisle 和 Chris Rowley 的 `fixltx2e` 宏包 (已安装在大多数 L^AT_EX 系统上) 来解决 [30]。请注意, `fixltx2e.sty` 是旧版 `fix2col.sty` [30] 的替代 (和超集)。但是, `fixltx2e/fix2col` 不应与 `stfloats` 包一起使用, 因为它们都以不同的方式修改了一些相同的浮动体例程。

请注意, 2015 年及以后的 L^AT_EX 2_ε 内核已经在系统中内置了 `fix2col.sty` 的更正, 在这种情况下, 如果尝试加载 `fixltx2e.sty`, 则会发出警告, 因为不再需要它。

Morten Høgholm 的 `dblfloatfix` 宏包 [31] 提供了 `fixltx2e` 和 `stfloats` 宏包的组合功能, 现在是获得这些功能的推荐方法。

最后, 作者还应该意识到 L^AT_EX 2_ε 内核 (用或不用 `stfloats` 修补) 有一个长期的限制, 因为它不允许跨越双栏的弹性空间根据需要拉伸或收缩两个主要文本列。因此, 双栏浮动体可能会导致完整的 `vbox` 错误, 因为

剩余的文本高度可能不等于正常大小的整数行。问题可能出现在没有垂直弹性间距的主文本列 (在具有双栏浮动的页面上)(例如在章节标题, 公式等周围), 并导致完整的 `vbox` 警告以及“拉开”的段落间距。为了纠正这个问题, 用户可以通过插入一个命令来手动调整双栏结构和主文本之间的空间:

```
\vspace*{-3pt}
```

(在需要时调整) 在双栏结构内。顺便提一下, IEEEtran 在形成文章标题时自动补偿这个问题。

1) 双栏公式: 使用 `figure*` 来获得双栏公式是可能的, 但不是令人满意的。IEEE 很少使用双栏公式, 因为它们会浪费空间, 因此这种功能很容易被滥用。正在考虑使用双栏公式的作者应该验证以前在他们计划提交的期刊中发表的论文中有一些例子。

这是很复杂的。尽管 IEEE 没有对双栏公式相对于主文本公式的顺序进行约束 (也就是说, 一组双栏公式可以位于页面的顶部或底部, 它们通常会出现在页面的顶部或底部在中间他们是常规公式), 双栏公式数必须随着向下进展而增加 (即, 页面底部的双栏公式必须比顶部的公式更高)。此外, 双栏公式应出现在它们被引用的同一页面上 (如果它们是常规公式, 它们将出现在页面上)。更加困难的是, L^AT_EX 2_ε 不会在定义它们的同一页面上放置双栏公式。最后, IEEE 通常不允许其他图片或表格出现在双栏公式和主文本之间 (通过规则彼此分开)。所有这些意味着定义双栏公式的位置必须与文本中最终将被引用的位置“断开”, 并且用户须手动干预公式编号。

因此, 用户必须

- 1) 在双栏公式要出现的前一页定义它;
- 2) 当定义双栏公式时, 重置公式计数器, 以免扰乱普通公式的计数器;
- 3) 手动设置双栏公式计数
- 4) 在文本中引用双栏公式的点处增加公式计数器, 以便在该点之后的常规公式的编号中考虑它们。

要做到这一切, 可以方便地使用“便笺簿”计数器来临时保存公式编号。这可以通过诸如文档的前言中的命令来完成:

```
\newcounter{MYtempeqncnt}
```

现在, 双栏公式在页面上定义, 它们出现在它们之前 (在这个例子中假设它们是 6 和 7):

```
\begin{figure*}[!t]
% ensure that we have normalsize text
\normalsize
% Store the current equation number.
\setcounter{MYtempeqncnt}{\value{equation}}
% Set the equation number to one less than the one
% desired for the first equation here.
% The value here will have to changed if equations
```

```
% are added or removed prior to the place these
% equations are referenced in the main text.
\setcounter{equation}{5}
\begin{equation}
\label{eqn_dbl_x}
x = 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19 + 21+ 23 + 25
+ 27 + 29 + 31
\end{equation}
\begin{equation}
\label{eqn_dbl_y}
y = 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20+ 22 + 24
+ 26 + 28 + 30
\end{equation}
% Restore the current equation number.
\setcounter{equation}{\value{Mytempeqcnt}}
% The IEEE uses as a separator
\hrulefill
% The spacer can be tweaked to stop underfull vboxes.
\vspace*{4pt}
\end{figure*}
```

其结果显示在本页面的顶部。该技术允许根据需要任意定位公式的定义，使得 (浮动) 公式将出现在期望的位置。“[t]”选项强制 L^AT_EX 尽力将公式放在下一页的顶部。如果它是 “[t]”，则需要加载 stfloats(甚至更好的 dblfloatfix) 包，并且必须在图中的等式之前执行 \vspace 命令，然后是 \hrulefill 命令。

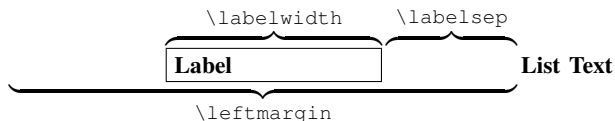
然后可以在正文中引用双栏公式，如：

```
% The previous equation was number five.
% Account for the double column equations here.
\addtocounter{equation}{2}
As can be seen in (\ref{eqn_dbl_x}) and
(\ref{eqn_dbl_y}) at the top of the page ...
```

值得庆幸的是，双栏公式很少见。

XI. 清单

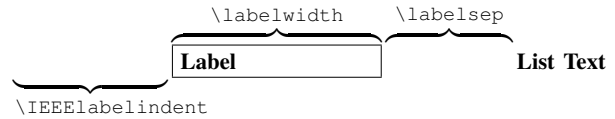
传统的 L^AT_EX itemize、description 和 (IED) list 环境不适合生成 IEEE 出版物中使用的列表样式。主要问题是它们不向用户提供控制结果列表的参数的手段。此外，对基础列表的参数进行全局更改将导致 (通常意外地对用户) 其他依赖于它的命令的不正确行为，例如 \quote。最后，L^AT_EX 的 \list 将列表文本的左边距视为参考点，该参考点确定列表相对于主文本左边距的定位方式：



这与使用标签盒作为列表结构的参考点的 IEEE 列表形成对比。即对于给定的情况，列表标签将缩进一定量，列表文本块将从标签框缩进给定量，并且这些间距将确定列表文本的位置。

出于这些原因，IEEEtran 提供了增强的 IED 列表环境，使得生成 IEEE 样式列表变得更加容易。底层的

\list 与传统的 L^AT_EX 保持一致，以免破坏依赖它的代码。IEEEtran 使用新的长度变量 \IEEElabelindent，以使用户可以直接以 IEEE 方式指定 IED 列表结构：



IEEEtran IED 列表忽略列表长度参数的所有“外部”更改。相反，IED 列表仅通过两个接口控制：

- 1) 通过 \IEEEiedlistdecl 命令进行“全局”控制；
- 2) 通过可以提供给 \itemize，\enumerate 和 \description 的可选参数进行局部控制。

例如，在 IEEEtran 文档中声明

```
\renewcommand{\IEEEiedlistdecl}
{\settowidth{\labelwidth}{Hello}}
```

将所有后续 IED 列表中的标签框的默认宽度设置为等于“Hello”的宽度。注意：由于设置 \labelwidth 是如此常见，因此 IEEEtran 提供了一个命令：\IEEEsetlabelwidth{X}，它是 \settowidth{\labelwidth}{X} 的一种较短的形式。

如果参数仅应用于单个 IED 列表，则使用局部控制：

```
\begin{itemize}[\IEEEsetlabelwidth{\gamma$}]
```

在 IED 列表中，局部控制在全局控制之后执行，因此，局部控制中的命令可以增强和反对全局控制中的命令。请注意，局部和全局控件中的代码的执行方式与普通 L^AT_EX 代码相同。因此，用户应确保控件中不显示不需要的空格。如果控件定义太长而无法放在一行上，请用“%”屏蔽行尾，以防止它们被解释为空白 (IV-B1 中有关于此主题的一些信息)。另请注意，L^AT_EX 解析器要求将括号放在带有可选参数的命令周围，这些参数直接放在其他命令的可选参数中：

```
\begin{itemize}[\mycmd{1}{example}]
```

这种 IEEEtran \IED 实现可以轻松控制 IED 列表，即使它们是深度嵌套的。

IED 列表使用的默认间隔存储在各种长度 (非宏) 命令中。很少需要对这些“主”默认值进行更改，并且只应在文档的开头进行，而不是在 IED 列表控件中进行。现在将简要解释这些常数。

\IEEEilabelindent：此长度是 itemize 标签框从左边距缩进的默认间距。IEEE 似乎使用至少两个不同的值。例如，在 IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY 和 IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATION 中，它们倾向于使用等于 \parindent 的缩进，而对于 IEEE TRANSACTION ON COMMUNICATIONS，它们倾向于缩进逐条列表更多 (1.3\parindent)。较短的长度存储为 \IEEEilabelindentA，更长的用

$$x = 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19 + 21 + 23 + 25 + 27 + 29 + 31 \quad (6)$$

$$y = 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20 + 22 + 24 + 26 + 28 + 30 \quad (7)$$

`\IEEEilabelindentB`，默认设置是使用较短的版本。要使用更长的版本，请在文件的开头执行

```
\setlength{\IEEEilabelindent}{\IEEEilabelindentB}
```

`\IEEEelabelindent`：此长度是 `enumerate` 标签框从左边距缩进的默认值。通常与 `\parindent` 相同。

`\IEEEedlabelindent`：同上 `description` 标签。通常与 `\parindent` 相同。

`\IEEEiednormlabelsep`：此长度是 IED 列表标签框和列表文本之间的正常默认间距。

`\IEEEiedmathlabelsep`：对于命名 `description` (数学符号列表及其解释)，IEEE 通常会增加术语和定义之间的分离。此长度设置为长于正常长度。要调用它，只需发出命令 `\IEEEiedmathlabelsep` 即可。

`\IEEEiedtopsep`：此长度是在每个 IED 列表上方和下方放置的额外垂直间距。IEEE 通常会在每个列表周围添加一些额外的间距。但是，这个额外的间距几乎不可察觉。

`\IEEElabelindentfactori` 与 `\IEEElabelindentfactorvi`：这些包含有效 `\IEEElabelindent` 随着列表嵌套深度增加而减少的因子。当列表嵌套级别增加时，IEEE 通常会减少缩进量，因为没有太多空间来缩进两列文本。IEEEtran 具有“自动缩进削减”功能，可提供此行为。标签框缩进的实际情况是 `\IEEElabelindent` 乘以对应于嵌套深度级别的 `\IEEElabelindentfactorX` (其中“X”是罗马数字的嵌套深度)。这提供了一种方法，用户可以通过该方法有效地更改 `\IEEElabel` 缩进以获得更深层次。可能没有“标准 IEEE”值这样的东西。IEEE 实际上做的可能取决于具体情况。第一个列表级别几乎总是有完整的缩进。第二级通常只有正常缩进地的 75%。第三级和更大的嵌套是非常罕见的，并且可能不使用任何缩进。这些因素不是长度，而是像 `\baselinestretch` 这样的常量宏，因此如果需要更改，则应使用 `\renewcommand`。默认值为

```
\IEEElabelindentfactori 1.0
\IEEElabelindentfactorii 0.75
\IEEElabelindentfactoriii 0.0
\IEEElabelindentfactoriv 0.0
\IEEElabelindentfactorv 0.0
\IEEElabelindentfactorvi 0.0
```

可以通过在列表控件中发出命令 `\IEEEenolabelindentdentortrue` 来暂停在 IED

列表中使用这些因素 (其效果与将所有缩进因子设置为 1.0)。

通常，IEEEtran 会根据 `\IEEElabelindent`，`\labelwidth` 和 `\labelsep` 的当前值自动计算 `\leftmargin`。要停止此自动计算，以便使用手动指定的 `\leftmargin` 值，只需在列表控件中使用 `\IEEEenocalcleftmargintrue`。在正常的 IEEE 相关工作过程中不应该需要此功能。

IEEEtran 提供了一种在 IED 列表标签框中手动指定对齐方式的方法。命令 `\IEEEiedlabeljustifyl`，`\IEEEiedlabeljustifyc` 和 `\IEEEiedlabeljustifyr` 可以在列表控件中使用，以分别将列表标签对齐到左侧，中间和右侧。`itemize` 和 `enumerate` 默认为右对齐，而 `description` 默认为左对齐。在正常的 IEEE 相关工作过程中，不应该需要这些命令。

除了修改 `itemize`，`enumerate` 和 `description` 的行为之外，IEEEtran 还提供了相应的别名 `IEEEitemize`，`IEEEenumerate` 和 `IEEEdescription`，它们为用户提供了一种使用 IEEE 样式列表环境的方法，即使在加载另一个包的情况下也是如此。覆盖 IED 列表环境。对于专门的应用程序，原始 `LaTeXIED` 列表环境保留为 `LaTeXitemize`，`LaTeXenumerate` 和 `LaTeXdescription`。

1) *Itemize*: `itemize` 通常会自动计算当前列表深度使用的任意符号的宽度，以使用户只需调用 `\begin{itemize}...\end{itemize}` 而不做任何特殊操作。此外，即使在列表开始之前重新定义了 `\labelitemX` (其中“X”表示“i, ii, .. iv”)，自动标签宽度功能也能正常工作。但是，如果要通过 `\item[X]` 指定任何项目符号 (这很少见，并且就 IEEE 相关工作而言可能是非标准的)，则可以使用以下形式：

```
\begin{itemize}[\IEEEsetlabelwidth{Z}]
\item[X] blah
\item[Y] blah
\end{itemize}
```

其中“Z”是列表中最长的标签。

2) *enumerate*: 关于 `enumerate` 的重要注意事项是 `\labelwidth` 将默认为正常大小和样式的长度“9”)。因此，如果满足以下任一条件，则必须手动指定最长标签的宽度：

- 1) 顶级列表包含 9 个以上的项目；

- 2) 重新定义了相关 `\labelenumX` 或 `\theenumX` ;
- 3) `\item[X]` 已用于手动指定标签;
- 4) 标签使用的字体不是正常尺寸和样式;
- 5) `enumerate` 是嵌套的 (即, 不在顶层), 因此不使用阿拉伯数字作为标签。

例如:

```
\begin{enumerate}[\IEEEsetlabelwidth{12}]
\item blah
\item blah
.
.
% 12 items total
\end{enumerate}
```

3) *description*: 一般来说, 总是必须为 `dexcription` 指定最长的标签宽度。此外, 作者可能希望在构建数学符号列表时使用 `\IEEEmathlabelsep` 代替 `\labelsep`。例如:

```
\begin{description}[\IEEEsetlabelwidth{\$alpha\omega
\pi\theta\mu\$}\IEEEusemathlabelsep]
\item[{\$gamma\delta\beta\$} Is the index of..
\item[{\$alpha\omega\pi\theta\mu\$} Gives the..
.
.
\end{description}
```

有时可能难以从检查中确定哪个标签最长。对于这种情况, 一些诊断代码可能有助于测量长度, 然后在控制台上显示结果:

```
\newlength{\mydiaglen} % put in preamble
.
.
\settowidth{\mydiaglen}{\$alpha\beta\gamma\$}
\showthe\mydiaglen
```

XII. 定理和证明

定理和相关结构, 如公理, 推论和引理, 以传统的 L^AT_EX 方式处理。用户必须首先通过声明声明结构名称:

```
\newtheorem{struct_type}{struct_title}[in_counter]
```

其中 `struct_type` 是用户为结构选择的标识符的命令, `struct_title` 是该结构的标题, `in_counter` 是计数器的可选名称, 其编号将显示结构编号, 其更新将重置结构计数器。大多数 IEEE 论文在整个工作中使用顺序定理编号, 因此通常不指定 `in_counter`。然而, 那些使用 `in_counter` 的论文通常使用 “section”, 使得节的编号是每个定理数的一部分。在定义结构之后, 可以通过以下方式使用它:

```
\begin{struct_type}[extra_title]
.
.
\end{struct_type}
```

其中 `extra_title` 是与结构一起显示的可选名称。

例如, 最常用的申明定理方法是使用:

```
\newtheorem{theorem}{Theorem}
```

按照环境的需要跟随

```
\begin{theorem}[Einstein-Podolsky-Rosenberg]
```

有时希望结构与另一个结构共享其计数器。这可以通过使用 `\newtheorem` 的替代形式来完成:

```
\newtheorem{struct_type}[num_like]{struct_title}
```

其中 `num_like` 是现有结构的名称。

IEEE 定理编号以它们定义的节编号为前缀 (例如, 2.5)。这给附录带来了困难 (特别是当用罗马数字编号时), 因为定理数字不是唯一的。为了解决这个问题, 在罗马编号的附录中, IEEEtran 将添加 “A” 前缀 (例如 A2.5)。对于 Alpha 数附录, 定理编号更直接 (例如 A.5, B.5 等)。对于单个附录, 使用常量 “A” 前缀 (例如 A.5)。

A. 证明

预定义的 `IEEEproof` 环境可以轻松处理证明:

```
\begin{IEEEproof}
.
.
\end{IEEEproof}
```

QED 符号 ■ 自动放置在每个证明的末尾。如果需要, 可以通过 `\IEEEQED` 命令手动访问该符号。实心 (默认) ■ 和空心 □ 形式分别作为 `\IEEEQEDclosed` 和 `\IEEEQEDopen` 提供。要将默认值从实心更改为空心 (某些期刊和/或作者更喜欢空心地), 只需根据需要重新定义 `\IEEEQED` :

```
\renewcommand{\IEEEQED}{\IEEEQEDopen}
```

`IEEEproof` 还支持可选参数, 该参数允许覆盖默认字符串 “Proof”:

```
\begin{IEEEproof}[Proof of Theorem \ref{thm:my}]
```

XIII. 结束部分

A. 附录

`\appendix` 命令用于启动单个附录。可选参数可用于指定标题:

```
\appendix[Proof of the Zonklar Equations]
```

用了 `\appendix` 后, `\section` 命令将被禁用, 任何使用 `\section` 的尝试都将被忽略, 并将导致生成警告消息。(单个附录标记了枚举部分的结尾, 并且部分计数器固定为零。当只有一个附录时, 不应使用 “参见附录 A”, 而是使用 “参见附录”。) 但是, 所有比 `\subsection` 和 `\section*` 低层的命令表单将正常工作, 因为 `acknowledgments` 等内容可能仍然需要这些命令。

如果有多个附录部分, 则使用 `\appendices`。然后用 `\section` 声明每个附录:

```
\section[Proof of the First Zonklar Equation]
```

如果不需要标题, 则可以将部分的必需参数留空 (`\section{}`)。重要的是要记住在引用节 (或小节

等)的任何其他小节或标签之前声明`section`。与`\appendix`一样,`\section*`命令和比`\subsection`等低的命令仍将照常工作。

IEEE 使用了两种附录编号约定。大写字母(例如“附录 B”)和罗马数字(例如“附录 II”)。前者似乎更受欢迎,是 IEEEtran 的默认方法。使用 IEEEtran 选项 `romanappendices` 获取罗马编号的附录。

一些作者倾向于将附录编号作为附录中出现的等式的等式数的一部分。这可以通过在第一个附录公式之前重新定义公式数来实现

```
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

对单个附录应使用常量“A”代替`\thectionction`。

B. 致谢

使用`\section*`命令创建致谢和其他未编号的部分:

```
\section*{Acknowledgment}
\addcontentsline{toc}{section}{Acknowledgment}
```

需要第二个可选命令来手动将这些部分添加到目录(很少使用,但有些作者可以使用草稿文件)以及文档的 PDF 书签(如果使用 `hyperref.sty`)。

请注意,IEEE 计算机协会的论文通常使用复数形式的“Acknowledgments”。

C. 参考文献

使用 IEEEtran B_BT_EX 包 [32] 最容易(并且正确)生成参考文献,可以通过

```
\bibliographystyle{IEEEtran}
\bibliography{IEEEabrv,mybibfile}
```

快速生成。有关更多信息,请参阅 IEEEtran B_BT_EX 包文档。

在向外部方提交文档源(.tex)文件时,强烈建议将 B_BT_EX 的 .bbl 文件手动复制到文档中(在传统的 L^AT_EX 参考文献环境中),以免依靠外部文件来生成参考文献并防止其中发生变化的可能性。

D. 传记

期刊文章的传记是使用 IEEEbiography 环境创建的,该环境支持包含照片的可选参数:

```
\begin{IEEEbiography}[\includegraphics[width=1in,
height=1.25in,clip,keepaspectratio]
{./shell}]{Michael Shell}
.
.
\end{IEEEbiography}
```

请注意,当在另一个命令的可选参数中使用带有可选参数的命令时,防止 L^AT_EX 解析器混淆所需的额外括号集。

或者,可以定义 L^AT_EX 宏(命令)以便于作者照片的简写表示法。如果未使用可选参数,则将为照片保留空格,并且将显示消息“PLACE PHOTO HERE”以代替照片。

IEEEtran 对于阻止 IEEEbiography 照片区域在页面中被破坏比较谨慎。如果看起来 IEEE 应该能够在页面末尾“挤压”,而是从新页面开始,请尝试插入

```
\vspace*{-2\baselineskip}
```

或在 IEEEbiography 之前试试该命令,看看它是否合适。

IEEE 的传记间距算法可能有点复杂,因为必须考虑美学。IEEEtran 将`\vfll`置于传记之上。这允许用户通过在传记之前或之后放置无限多的可拉伸的`\vfill`来根据需要向下或向上推送传记。

照片区域长 1.25 英寸,宽 1 英寸。IEEE 建议作者照片图像的分辨率应为 220dpi(每英寸点数),灰度为 8bits/sample。

如果没有可用的照片,则可以使用`\IEEEbiographynophoto`环境,该环境不支持照片的可选参数或保留空间。

XIV. 尾页列均衡

IEEE(粗略地)均衡最后一页上的列的长度。在参考文献或 IEEEbiography 条目通常不被破坏的前提下粗略的平衡,因此列长度通常不完全相等。

平衡最后两列对于准备工作尤为重要。建议作者使用手动方法,在适当的位置放置`\newpage`,或者在最后一页的第一列顶部的某处放置`\enlargegethispage{-X.Yin}`,其中“X.Yin”是给定页面的有效缩短文本高度。

有时这样的命令必须位于参考文献条目之间。这可能是一个问题,因为虽然命令可以放在 .bbl 文件中,但是下次运行 B_BT_EX 时它会被覆盖。对于这种情况,IEEEtran 提供了一种通过`\IEEEtriggeratref{}`命令在给定引用号之前调用命令的方法。例如,发出命令在参考文献 10 之前使用

```
\IEEEtriggeratref{10}
```

会在参考文献之前插入分页符。执行的命令默认为`\newpage`。但是,这可以通过`\IEEEtriggercmd`命令进行更改:

```
\IEEEtriggercmd{\enlargegethispage{-5.35in}}
```

请注意,如果文档内容发生更改,则必须重新调整手动设置的断点或页面大小。

有 L^AT_EX 包,例如 `balance.sty` [33] 和 `flushend.sty` [34],用于自动平衡最后一页上的列。`Flushend`不需要在最后一页的第一列中放置任何特殊命令,`balance.sty`可能需要。但是,不推荐使用这些宏包,因为已知它们在操作中不太完全可

靠。balance.sty 的作者并不保证它适用于所有可能类型的页面，尤其是带有图片的页面。在某些情况下，flushend.sty 将导致最后一页第二列中参考文献内两行之间的间距异常 (变得大于参考文献之间的空间)。这个问题似乎是因为 IEEEtran 中的参考文献项之间没有空格的列表，这些列表项在脚注中。对于相同类型的列表，在 article.cls 下也会出现此问题。可以通过 flushend 命令 (例如 “\atColsBreak{\vskip-2pt}”) 在列中断处调整间隔符来手动校正 flushend 异常，但必须这样做部分地违背了使用宏包的目的。如果使用 flushend.sty 或 balance.sty，请务必仔细检查文档是否有任何间距问题，尤其是在最后一页。

附录 A 安装 IEEETran

首先，用户应该知道，根据 IEEEtran 存档包的目标操作系统 (例如，Unix 的 tar.gz 或 MS 的 .zip)，基于纯文本的 IEEEtran 文件 (.bst,.cls,.sty,.tex 等) 可以使用两种不同类型的行尾字符约定。Unix (包括 Mac OS X) 系统使用换行符 <lf>(0x0A)，而 MS Windows 系统使用回车符/换行符对 <cr><lf>(0x0D 0x0A) 来表示行尾。¹⁰ 大多数现代 L^AT_EX 系统都能接受不同的行尾约定，但有些文本编辑器却不能。(此处的症状包括在一条长线上出现的文字，双倍间距等)

L^AT_EX .cls 文件放在 texmf>/tex/latex 目录中时，可以在系统范围内访问，其中 texmf> 是用户 T_EX 安装的根目录。在具有本地 texmf 树 (<texmflocal>) 的系统上，可以将其命名为 “texmf-local” 或 “localtexmf”，建议在 <texmflocal> 中安装宏包，而不是 <texmf>。在重新安装和/或升级 L^AT_EX 系统之后，保留前者而不是前者。

建议用户为所有 IEEE 相关的 L^AT_EX 类和包文件创建子目录 <texmf 或 texmflocal>/tex/latex/IEEE。在某些 L^AT_EX 系统上，在对系统文件进行添加或删除后，需要刷新目录查找表。对于 T_EXLive 系统，这是通过作为 root 用户执行

```
texhash
```

完成的。MiK_TE_X 用户可以运行

```
initxmf -u
```

来完成同样的事情。

不愿意或无法在系统范围内安装文件的用户可以将它们安装在个人目录中，但是在 L^AT_EX 中引用它们时，除了文件名之外还必须提供路径 (完整或相对)。

¹⁰当然，纯文本存在不同的惯例这一事实本身就是荒谬的。有关历史和详细信息，请参阅 <http://en.wikipedia.org/wiki/Newline> 上的维基百科文章 “Newline”。

附录 B POSTSCRIPT/PDF 输出

某些 L^AT_EX 系统未正确配置以生成高质量的 PostScript 或 PDF 输出。这在历史上一直是 IEEE 相关工作的一个问题，因为已知 IEEE 使用的独特字体组合会引发某些 L^AT_EX 设置的问题。幸运的是，现代 L^AT_EX 系统现在这些类型的问题相对不常见。

为了帮助 IEEE 作者检测和纠正 L^AT_EX PostScript/PDF 生成问题，开发了 “Testflow” 诊断套件 [35]。鼓励作者花时间完成测试流程诊断，并在必须依赖 L^AT_EX 系统进行生产工作之前识别并纠正潜在问题。具有诸如不正确的边距，字体类型，PDF 格式错误或不正确的字体嵌入之类的问题的论文尽可能避免在稿件接受过程中引起延迟。

附录 C 其他有用或有用的外包装

A. acronym.sty 包

Tobias Oetiker 的 acronym.sty [36] 可能对有很多首字母缩略词的论文很有用。但是，请注意首字母缩写词环境和 IEEEtran description 环境之间的兼容性问题 (请参阅附录 F 节)。

B. url.sty 包

包含 URL，电子邮件地址等的论文可能会受益于使用 Donald Arseneau 的 \url.sty L^AT_EX 宏包 [37]，该宏包可在此类结构中提供更智能的换行符。请注意，IEEEtran.cls 会自动将 \url.sty 的 url 字体样式设置为 “相同” (即，URL 将以与它们出现的文本相同的字体呈现)，如 IEEE 期刊所做的那样。要覆盖它，作者必须将 \urlstyle 放在 \begin{document} 之后。

C. IEEEtrantools 包

IEEEtran L^AT_EX 类提供的一些独特命令可能在使用其他类文件 (例如论文、技术报告等) 的非 IEEE 相关工作中使用。IEEEtrantools.sty 包 [38] 提供了几种流行的 IEEEtran 命令，包括 \IEEEPARstart，IEEE 样式 IED 列表环境，IEEEeqnarray 命令系列，IEEEproof 环境和 \IEEEauthorrefmark。在使用 IEEEtran 类文件不需要加载 IEEEtrantools 宏包，也不应该加载它。有关更多详细信息，请参阅 IEEEtrantools 文档。

附录 D 常见的用户错误

IEEEtran 的许多用户错误涉及做太多而不是太少。较旧的类文件可能需要改造一番才能使格式更接近 IEEE 的格式。然而不再需要这些调整。用户应仔细检查所有已加载的宏包，以确保它们在最新版本的 IEEEtran 下仍然有用。不要只是因为“这是它一直以来的方式加载宏包”。对于手动调整的间距，边距，纸张尺寸等也是如此。

以下是一些比较常见的错误要避免。

在 \caption 之前放置 \label：这被认为是历史上 L^AT_EX 中最常出现的错误之一。请记住，\label 必须放在 \caption 之后或之内，以便能够正确引用图表。由于它是实际设置引用的是 caption 的计数器，\label 放在 \caption 之前将 \label 对应到 \section 上，而不是所需的图表。

更改默认字体：作者应允许 IEEEtran 设定字体。除非特别指示，例如在 comsoc 模式下或在提交的特定会议/日志的作者指令中，否则不要尝试使用覆盖默认字体的包，例如 pslatex，mathptm 等。

更改默认间距，节标题样式，边距或列样式：作者不应尝试手动更改边距，纸张大小（IEEEtran 类选项中提供的除外）或使用这样做的包（geometry.sty 等）。不需要在图片，公式等周围添加间距（除了可能对于 X-D 部分中描述的双栏浮标）。

简图使用位图：L^AT_EX 一直倾向于使用 Encapsulated PostScript(EPS) 或 pdfL^AT_EX，可移植文档格式(PDF)，(可以认为是 PostScript 的一种子集)，对于图片(有关更多信息，请参阅 X-A)。EPS/PDF 支持矢量(即，包含数学上描述的线，圆等对象)和位图(即仅包含像素形式的样本)图像。前者应该总是用于绘图，图表等，而后者通常必须与照片一起使用(因为它们的内容通常不能用数学方式容易地描述)。作者使用的绘图和绘图工具应该能够以矢量(EPS 或 PDF)格式直接输出。¹¹矢量 EPS/PDF 图像可以缩放，旋转和放大，而不会出现像素化或变灰或“锯齿状”等劣化。对于照片，IEEE 建议使用 EPS/PDF(很容易直接导入(pdfL^AT_EX))，PNG 或 TIFF。对于作者照片，JPEG(JPG) 通常是可以接受的。使用其他图片格式(如 BMP，EMF，VSD 等)对 IEEE 期刊来说是不可接受的。关于它们接受的图形格式的类型，一些 IEEE 会议可能更自由。使用其他图形格式(如 BMP，EMF，VSD 等)对 IEEE 期刊来说是不可接受的。

¹¹一旦 EPS/PDF 矢量形式的图像转换为位图形式(GIF，PNG，TIFF，JPEG 等)，即使稍后将其转换回 EPS/PDF，它也几乎总是无法挽回地锁定为位图形式。

文档使用点阵字体或嵌入和子集化所有文档字体：作者应使用测试流程诊断 [35] 检查其系统，以确保仅使用矢量(Type 1) 字体，并且所有字体都是嵌入和子集化的。使用点阵字体或未能包含所有(且仅)所需字体字形的文档可能被 IEEE 拒绝。注意产生这些问题的输出的图形绘图应用程序(如果在不包括图片时问题消失)。

使用较旧的插图宏包：作者不应使用 graphics 或 graphicx(首选)包以外的任何插图宏包。诸如 psfig, epsf 等较旧的接口已经过时多年。

未能正确划分长公式：作者有义务确保所有公式都适合其列宽。不可否认，分解公式并不总是容易做到，双栏格式对允许的公式宽度设置了严格的限制。但是，只有作者可以在不改变其含义或影响可读性的情况下划分等式。使用子函数是减少公式宽度的有效方法，但改变数学字体大小则不然。

手动格式化引用：这不仅容易出错，而且还需要大量工作。最好使用 IEEEtran BbL^AT_EX 样式 [32]。

附录 E 已知的问题

acronym.sty：由于修改后的 IEEE 样式 dextrcription 环境，acronym 环境将出现 IEEEtran 的问题。acronym 环境的可选参数不能用于设置最长标签的宽度。解决方法是使用 \IEEEiedlistdecl 来完成同样的事情：

```
\renewcommand{\IEEEiedlistdecl}{\IEEEsetlabelwidth{SONET}}
\begin{acronym}
.
.
\end{acronym}
\renewcommand{\IEEEiedlistdecl}{\relax}% reset back
```

cite.sty：5.0(2009-03-20) 之前的版本引用无法配合 hyperref.sty 生成超链接。

hyperref.sty：6.72u 之前的版本会干扰 \appendix 的可选参数。

小型大写字体变体：免费 L^AT_EX 系统中使用的小型大写字体大小约为普通大小字母的 80%。但是，IEEE 在期刊中使用的小型大写字体略小，比例约为 75%。因此，在免费 L^AT_EX 系统下生成的节标题的宽度将比实际期刊中使用的宽度略宽。在许多商业 L^AT_EX 系统(例如来自 YandY 的系统)中使用的小型大写字体具有大约 65% 的比率。因此，这些系统将生成比 IEEE 出版物更窄的章节标题。这种变化无需担心。

附录 F IEEEEQNARRAY 命令 (可选-适用于高级用户)

实际上, 所有 \LaTeX 对齐命令 (如 $\backslash\text{eqnarray}$, $\backslash\text{array}$ 和 $\backslash\text{tabular}$) 都基于 \TeX 命令 $\backslash\text{halign}$ 。 \LaTeX 的目标是简化 $\backslash\text{halign}$ 的使用, 这是值得推崇的。但是, 在隐藏大部分较低级别的界面时, 会失去相当程度的灵活性。这导致了几个宏包的开发, 例如 amsmath [11], array.sty [28] 和 MDW 工具 [17], 每个宏包都提供了更强大的对齐结构。

IEEEtran 还提供了自己独特的一组对齐工具, 称为 IEEEeqnarray 系列。 IEEEeqnarray 系列的设计理念是提供一个 \LaTeX 对齐接口, 它更接近于底层 $\backslash\text{halign}$, 但要将其与高级列定义管理和自动前导码构建机制相结合 (这些机制很繁琐) 的 \TeX 。因此, IEEEeqnarray 命令系列足够灵活, 几乎可以替代所有其他用于生成多行公式和对齐的框结构的 \LaTeX 命令, 如矩阵和文字表格和数学环境。因为用户使用上更自由, 所以操作规则更加复杂。因此, IEEEeqnarray 命令主要针对更高级的 \LaTeX 用户。

本节中描述的 IEEEeqnarray 系列工具的使用是完全可选的。 IEEEeqnarray 代码是独立的, 不依赖于其他对齐宏包, 它们可以与它一起使用或代替它。 IEEEtrantools.sty 宏包 (参见附录 C-C) 适用于希望在 IEEEtran.cls 之外使用 IEEEeqnarray 系列的用戶。

关于使用 IEEEeqnarray 的推荐信息来源包括 Stefan M.Moser 的 *How to Typeset Equations in \LaTeX* [6] 和 Tobias Oetiker 的 [5] 中的 *The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$* 。

A. IEEEeqnarray

IEEEeqnarray 环境用于占据整列的多行公式。它的使用方式与 $\backslash\text{eqnarray}$ 非常相似, 但有两个额外的参数, 其中一个强制性的, 另一个是可选的:

```
\begin{IEEEeqnarray}[decl]{cols}
.
\end{IEEEeqnarray}
```

可选参数用于在环境中执行在对齐实际开始之前的命令。这就像 IEEEtran $\backslash\text{IED}$ 列表环境的局部控制一样。还有一个全局控件 $\backslash\text{IEEEeqnarraydecl}$, 它在局部控制之前执行。默认情况下, $\backslash\text{IEEEeqnarraydecl}$ 被定义为 $\backslash\text{relax}$ 。如 XI 节所述, 用户应注意不要在这些控件中出现不需要的空格, 因为这些内容将出现在 IEEEeqnarray 结构之前。此外, 请记住, 为防止 \LaTeX 解析器混淆, 如果参数包含带有可选参数的命令, 则可选参数的内容必须括在括号中。

强制参数 cols 包含列和列间分隔符间距 (\TeX 中的 “inter-column tabskip glue”) 类型说明符。列类型由

表 IV

IEEEeqnarray PREDEFINED COLUMN TYPES

I.D.	Description	I.D.	Description
l	left math	v	vertical rule
c	centered math	vv	two vertical rules
r	right math	V	double vertical rule
L	left math with ords	VV	two double vertical rules
C	centered math with ords	h	horizontal rule
R	right math with ords	H	double horizontal rule
s	left text	x	empty
t	centered text	X	empty math
u	right text		

Note: S, T, U, p, and P are likely to be used in future versions.

表 V

IEEEeqnarray PREDEFINED COLUMN SEPARATION (GLUE) TYPES

I.D.	Width [*]	I.D.	Width [*]
!	$-1/6em$.	$0.5\backslash\text{arraycolsep}$
,	$1/6em$	/	$1.0\backslash\text{arraycolsep}$
:	$2/9em$?	$2.0\backslash\text{arraycolsep}$
;	$5/18em$	*	0pt plus 1 fil
'	$1em$	+	1000pt minus 1000pt
"	$2em$	-	0pt

^{*}All em values are referenced to the math font.

$1em = \backslash\text{quad}$, $2em = \backslash\text{qqquad}$

字母标识。有几种预定义的列类型可用, 如表 IV 所示。有两种间距类型。预定义的间距类型由各种标点符号表示, 如表 V 所示。用户定义的间距类型由数字表示。

放置这些说明符的规则如下:

- 1) 没有两个间距符可以彼此相邻-它们不是相加的, 必须至少由一个列说明符彼此分开;
- 2) 假设背靠背列说明符之间的列间距为零;
- 3) 由于规则 (1), 背靠背数字将被视为由所有数字代表的数值指定的单个间距;
- 4) 可以通过将字母括在大括号内来访问多字母列说明符 (否则它将被解释为几个单字母列说明符)。由于规则 (3), 多数字间距说明符周围不需要括号;
- 5) 必须至少有一个列说明符, 但没有固定的上限可以支持多少列;
- 6) IEEEeqnarray , 如果没有指定列间距, 则将在 cols 规范的每一端加上 “+” 的间距。

这导致了一个像 $\backslash\text{eqnarray}$ 这样的中心结构 (每侧的 1000pt 减去 1000pt 的间距 “根据需要从主文本列的每一侧压缩” 到中间的中心)。此外, IEEEeqnarray 会自动为最后指定列的右侧添加公式编号的隐藏列。目前,

左侧不支持公式编号。¹²

B. 定义列类型

新的列类型可以用

```
\IEEEeqnarraydefcol{col_id}{predef}{postdef}
```

命令定义。col_id 参数包含列说明符的名称，该名称应仅包含一个或多个字母。给定的列说明符，甚至是预定义的列说明符，也可以随意重新定义，而不会发出警告或错误。¹³ \predef 参数包含将在列中的每个单元格之前插入的命令。postdef 参数包含将在列中的每个单元格之后插入的命令。例如，

```
\IEEEeqnarraydefcol{g}{\hfil$\clubsuit$}{\diamondsuit$\hfil}
```

将定义“g”文本列，该列将在单元格内容的任一侧放置球杆和钻石套装符号，并将相应结构置于单元格内。例如：

♣Hello◇

使用 \hfil 控制单元格对齐允许用户通过在单元格内容的一侧或两侧放置无限多的可伸展 \hfill 来逐个单元地覆盖列对齐。 \hfill 甚至可以放置在单元格中的项目之间，以强制它们与“单元格墙壁”分开。 \IEEEeqnarray 预定义列旨在允许用户通过 \hfill 尽可能覆盖（即使对于数学模式单元格）。

请注意 TeX 不允许在命令参数中使用不匹配的大括号。如果需要大括号，例如命令的参数，则必须在单元格本身内提供大括号。例如

```
\IEEEeqnarraydefcol{myp}{\parbox[c]{0.5in}}{}
\begin{IEEEeqnarraybox}[myp]c
{first\second}&\alpha\\
&\beta%
\end{IEEEeqnarraybox}
```

定义一个名为“mup”的列类型，它将文本放在 0.5 英寸宽的 parbox 中，该 parbox 以单元格的基线为中心。请注意，因为列类型名称由多个字母组成，所以必须将其括在列规范中的一组额外大括号中，否则它将被解释为三个相邻的列“m”“y”和“p”。此外，单元格的内容必须括在大括号内，以便 (1) \parbox 命令将整个内容视为其参数；(2) parbox 中的换行符不会被解释为对齐行的结尾。请注意，可能会发生列被赋予空单元格，例如在示例的第二行中，或者输入空白分隔符行时。发生这种情况时，将在列中显示 \relax，它将作为命令的参数获取。

作为参考，此处展示预定义列类型的定义：

```
% math
\IEEEeqnarraydefcol{l}{\IEEEeqnarraymathstyle}\hfil}
```

¹²这并不是说现有能力是不可能的，只是丑陋。

¹³因此，允许添加新的预定义列类型，而不会破坏现有代码。

```
\IEEEeqnarraydefcol{c}{\hfil$\IEEEeqnarraymathstyle}\hfil}
\IEEEeqnarraydefcol{r}{\hfil$\IEEEeqnarraymathstyle}\hfil}
\IEEEeqnarraydefcol{L}{\IEEEeqnarraymathstyle}\hfil}
\IEEEeqnarraydefcol{C}{\hfil$\IEEEeqnarraymathstyle}\hfil}
\IEEEeqnarraydefcol{R}{\hfil$\IEEEeqnarraymathstyle}\hfil}
% text
\IEEEeqnarraydefcol{s}{\IEEEeqnarraytextstyle}\hfil}
\IEEEeqnarraydefcol{t}{\hfil\IEEEeqnarraytextstyle}\hfil}
\IEEEeqnarraydefcol{u}{\hfil\IEEEeqnarraytextstyle}\hfil}
% vertical rules
\IEEEeqnarraydefcol{v}{\vrule width\arrayrulewidth}
\IEEEeqnarraydefcol{vv}{\vrule width\arrayrulewidth\hfil}\hfil}
\IEEEeqnarraydefcol{V}{\vrule width\arrayrulewidth\hskip\doublerulesep\vrule width\arrayrulewidth}
\IEEEeqnarraydefcol{VV}{\vrule width\arrayrulewidth\hskip\doublerulesep\vrule width\arrayrulewidth\hfil}%
{\hfil\vrule width\arrayrulewidth\hskip\doublerulesep\vrule width\arrayrulewidth}
% horizontal rules
\IEEEeqnarraydefcol{h}{\leaders\hrule height\arrayrulewidth\hfil}
\IEEEeqnarraydefcol{H}{\leaders\vbbox{\hrule width\arrayrulewidth\vskip\doublerulesep\hrule width\arrayrulewidth}\hfil}
% plain
\IEEEeqnarraydefcol{x}{\relax}
\IEEEeqnarraydefcol{X}{\relax}
```

请注意，分别在数学和文本列中包含命令 \IEEEeqnarraymathstyle 和 \IEEEeqnarraytextstyle。这些命令允许用户控制所有数学和文本列的样式。但是，由于更改适用于所有列，因此如果在同一对齐中需要不同的样式，则用户必须定义新的列类型（或者可以在每个单元格中手动指定不同的样式）。这些命令的默认定义是

```
\newcommand{\IEEEeqnarraymathstyle}{\displaystyle}
\newcommand{\IEEEeqnarraytextstyle}{\relax}
```

它允许文本列处于开始对齐时生效的任何样式，默认数学样式将采用 display 样式，但可以根据需要轻松更改。例如

```
\begin{IEEEeqnarray}[\renewcommand{\IEEEeqnarraymathstyle}
{\scriptstyle}]{rCl}
```

将导致 \scriptstyle 数数列。

与垂直和水平线有关的列将在附录??中讨论，因为它们通常仅在生成表时使用。

“x”和“X”列是基本的空文本和数学模式列，没有任何格式或样式控件。

C. 定义间距类型

新的分隔间距类型用

```
\IEEEeqnarraydefcolsep{colsep_id}{def}
```

定义。参数 colsep_id 参数包含列分隔间距说明符的个数，该编号应仅包含数字。不同的间距类型名称必须具有不同的数值。（“007”与“7”相同。）用户定义的列

间距指定符可以随意重新定义，不会出现警告或错误。`def` 参数包含给定间距类型的宽度。宽度可以指定为绝对值或相对长度命令：

```
\IEEEeqnarraydefcolsep{9}{10pt}
\IEEEeqnarraydefcolsep{11}{2\tabcolsep}
```

间距类型宽度在定义时不会被计算，但每次实际引用为 `IEEEeqnarray` 列说明符时都会进行计算。因此，对于上例中的第二个定义，如果在定义间距类型后修改 `\tabcolsep`，修改后的值将是使用的值。

也允许使用弹性长度。可以利用“+”是已知值的事实来实现一些有趣的效果。例如，

```
\IEEEeqnarraydefcolsep{17}{200pt minus 200pt}
```

将产生一个分隔间距，它始终是从等式边到主文本列末端的距离宽度的 1/5。当然，可以根据需要使用“+”来生成等距离公式组，如 `\verbamsmath|` 的 `\align`：

```
\begin{IEEEeqnarray}{R1+R1+R1}
```

D. 一个简单的使用例子

IX 节中的示例可以使用 `IEEEeqnarray` 实现

```
\begin{IEEEeqnarray}{rCl}
Z&=&x_1 + x_2 + x_3 + x_4 + x_5 + x_6\IEEEnonumber\\
&&+&:\ a + b%
\end{IEEEeqnarray}
```

如表 I 所示，“C”列类型是一个居中的数学模式列，两侧都有空的占位符 (“{}”)。因此，没有必要在等号周围放置空的占位符。与 `\eqnarray` 一样，& 分离列单元格，并且是列间距将出现的位置 (非零时)。

请注意第二行末尾存在 %。TeX 不会忽略在命令或列分隔符 & 之前出现的空格，但会忽略之后出现的空格。大多数 TeX 对齐实现通过 `& \` 和 `\end` 去除之前的所有间距来保护用户免受此行为的影响。`IEEEeqnarray` 系列不会这样做！因此，除非需要，否则在这些命令之前防止空格 (包括行尾的隐含空格)。如果列中存在无法解释的偏移，则可能是此问题造成的。在给定的示例中，不需要的间距不是问题，因为无论如何都在数学模式中忽略末尾间距。但是，如果列使用文本模式，则会出现问题。

或者，可以使用双列形式：

```
\begin{IEEEeqnarray}{rL}
Z=&x_1 + x_2 + x_3 + x_4 + x_5 + x_6\IEEEnonumber\\
&+&:\ a + b%
\end{IEEEeqnarray}
```

E. 公式编号

像 `\eqnarray`，`IEEEeqnarray`，有一个“星形模式”，`\IEEEeqnarray*`，默认情况下不会在每行的末尾放置公式编号。通过在最后一列中根据需要放置命令 `\IEEEyesnumber` 或 `\IEEEnonumber`，可以覆盖各行的

默认行为。`IEEEeqnarray` 还提供 `\IEEEyessubnumber` 和 `\IEEEnosubnumber`，可用于启用或禁用给定行的子公式编号。为了支持此功能，`IEEEtran` 定义了自己的 `IEEEsubequation` 计数器 (通过更改公式重置) 和 `\theIEEEsubequation` 命令。¹⁴

从 `IEEEtran` 的 1.8 版开始，开始提供 `\IEEEyesnumber*`，`\IEEEnonumber*`，`\IEEEyessubnumber*` 和 `\IEEEnosubnumber*`，它们在行之间持续存在，直到另一个星形命令出现。根据需要，可以使用非星形表格有选择地覆盖以后各行的行为。

尽管有四个编号命令，但记住只有三种 `IEEEeqnarray` 编号模式：

- 1) 什么都不显示，不要改变计数器的值 (`\IEEEnonumber`);
- 2) 增加公式计数器并显示非子公式部分的公式编号 (`\IEEEyesnumber`);
- 3) 增加子公式计数器并显示带有子公式编号 (`\IEEEyessubnumber`) 的公式编号。

`\IEEEnosubnumber` 并不是真正需要的，并且表现得像 `\IEEEyesnumber`，除非前者不启用等式编号 (如果它尚未打开) (并且如果公式编号关闭则不会改变当前行的编号属性)。

一般来说，每行只应使用一个编号命令。特别是在单行上混合显示行号和不显示行号命令可能会导致意外操作。然而，一个值得注意的例外是非常有用的，`\IEEEyesnumber\IEEEyessubnumber` 组合，它开始一个新的子公式序列。例如：

```
\begin{IEEEeqnarray}{c}
x1\IEEEyesnumber\IEEEyessubnumber * \\
x2\\
x3\IEEEyesnumber\IEEEyessubnumber\label{eqn:expl}\\
x4\\
x5\IEEEyesnumber * \\
x6
\end{IEEEeqnarray}
```

产生的结果是：

$$\begin{array}{r}
 x1 \tag{8a} \\
 x2 \tag{8b} \\
 x3 \tag{9a} \\
 x4 \tag{9b} \\
 x5 \tag{10} \\
 x6 \tag{11}
 \end{array}$$

`\IEEEyesnumber` 命令递增公式计数器，否则它将是一个子公式行，重置子公式计数器并关闭子公式编号。然

¹⁴实际显示的是 `\theIEEEsubequationdis` 命令。

后, 以下 `\IEEEyessubnumber` 将子公式计数器递增 1 并恢复子公式编号。¹⁵

请注意, (子) 公式的任何标签必须放在任何编号控制命令之后, 因为在该点之前, 标签将引用在没有任何编号控制命令时将使用的公式编号。

请注意 `\IEEEeqnarray`, 如 `\eqnarray` 一样, 如果等式足够长, 将在没有警告的情况下覆盖等式编号!¹⁶ 对于发生这种结果的情况, 用户可以在行尾(如果使用了 `\IEEEyessubnumber`, 就在这之后) 插入 `\IEEEeqnarraynumspace` 命令, 该命令将插入宽度等于显示的等式编号的空格:

```
... + x_z \IEEEyessubnumber\IEEEeqnarraynumspace\
```

结果是整个多行公式将略微向左移动。当遇到这种情况时, IEEE 经常在其期刊上做同样的事情。如果产生 `overfull hbox` 的结果, 则必须进一步分解该公式。

F. 额外的垂直间距与分页

与 `\eqnarray` 一样, `\IEEEeqnarray` 命令支持一个禁止在给定行上分页的星形模式, 以及一个可选的额外垂直间距参数:

```
&+ \:a+b\*[5pt]
```

从 IX 节提醒用户, `amsmath` 将配置 `TeX` 以禁止多行公式中的分页符, 包括由 `\IEEEeqnarray` 创建的分页符, 因为 `\interdisplaylinepenalty` 的值对它也有效。

和 `\eqnarray` 一样, `\IEEEeqnarray` 通常在行之间放置一些额外的间距(由长度命令 `\jot` 指定) 来拉开公式间距, 以及防止大符号靠近它们上一行的基线。

G. `\IEEEeqnarraybox`

`\IEEEeqnarray` 不适合生成矩阵和表格等结构, 因为它必须具有对主文本列的独占访问权限, 并且不能嵌套在其他结构中。对于这些应用程序, 提供了 `\IEEEeqnarray` 框命令。`\IEEEeqnarraybox` 与 `\IEEEeqnarray` 的区别在于以下方面:

- 1) 整个内容被包装在一个盒子里, 因此可以嵌套在其他显示或对齐结构中(例如 `\equation`, `\IEEEeqnarray` 或甚至是另一个 `\IEEEeqnarraybox`)。请注意, 与所有框结构一样, `\IEEEeqnarraybox` 的行之间不允许分页;

¹⁵仅在正常的等式数字行之后调用 `\IEEEyessubnumber` 将产生类似 14.14a 的序列。IEEE 通常不使用正规方程数, 后面跟着带有相同基本公式的子公式, 但是如果你需要它, 那么能力就在那里。v1.8 之前的 `IEEEtran` 版本在这里有所不同, 因为它们会自动将“第一次”调用中的等式编号提升到 `\IEEEyessubnumber`, 因此没有这种程度的灵活性。

¹⁶如果公式要保持居中而不考虑公式编号的宽度, 这种行为是非常难以避免的, 甚至没有考虑到任何给定情况下接近距离的主观问题!

- 2) 第一列和最后一列外端的默认间距是 `0pt`(“-”), 而不是“+”如 `\IEEEeqnarray` 一样;
- 3) 没有提供自动(隐藏)公式编号;
- 4) 星形“`\IEEEeqnarraybox*`”关闭每行后的额外 `\jot` 垂直间距;
- 5) `\IEEEeqnarrayboxdecl` 是全局控制命令。

有两中次模式可用: 一个是 `\IEEEeqnarrayboxm`, 用于数学模式, 类似于 `\array`; 另一个是 `\IEEEeqnarrayboxt`, 它在文本模式下使用, 类似于 `\tabular`。如果通过 `\IEEEeqnarraybox` 调用, 将自动检测当前的数学/文本模式, 并自动选择正确的模式。因此 `\IEEEeqnarraybox` 可以替换 `\array` 以及 `\tableular`。

`\IEEEeqnarraybox` 的语法类似于 `\IEEEeqnarray`, 但有两个额外的可选参数:

```
\begin{IEEEeqnarraybox}[decl][pos][width]{cols}
.
\end{IEEEeqnarraybox}
```

`pos` 参数可以是 `tcb` 之一, 用于控制框相对于当前基线垂直对齐的位置: `t` 在顶行; `c` 在中心;¹⁷ `b` 位于底行。默认值为 `b`。

`width` 参数指定框的宽度。**警告:** 如果指定了宽度, 则列间间距说明符中必须有一个或多个弹性长度(例如“*”或“+”), 以便可以根据需要调整框的大小。如果没有这样的间距, 或者所提供的间距不能根据需要拉伸/收缩, 则不能将盒子的大小设置为指定宽度, 并且将导致不足或过满的 `hbox` 错误。如果没有提供宽度参数, 则框将设置为其自然宽度(并且不需要使用弹性列间距)。

`\IEEEeqnarraybox` 使用与 `\IEEEeqnarray` 相同的列和间距类型说明符/定义。

H. `TeX` 中的行间距

在讨论 `\IEEEeqnarray` 系列中垂直间距控制的一些更高级方面之前, 重要的是要讨论 `TeX` 的行间距算法的细节。通常, 基线由 `\length` 命令 `\baselineskip` 给定的数量分隔。每次更改字体大小时, `\baselineskip` 都会重置为该字体大小的默认值(乘以 `\baselinestretch`)。然后将 `\baselineskip` 的值保存到长度变量 `\normalbaselineskip` 中(这样即使用户将 `\baselineskip` 设置为另一个值, 也可以稍后引用正常值)。但是, 如果一条线的顶部比 `\lineskiplimit` 更

¹⁷中心实际上是沿着“数学轴”完成的(不完全在文本基线上, 但非常接近它)。许多 `LaTeX` 用户并不知道这种微小的区别。

接近它上面的线的底部, 则将暂停使用 `\baselineskip`, 并且将在两行之间放置 `\lineskip` 间距。¹⁸

该系统适用于文本模式。然而, 对于数学, 其符号具有更高的动态范围的高度和深度, 通常更好的是继续并且总是添加额外的固定量的距离 (`\jot`), 如附录 F-F 中所述。

加载 `IEEEeqnarray` 系列时, 会定义一个新的长度命令, `\IEEEnormaljot`, 它存储 `\jot`¹⁹ 的标称值, 因此即使当前正在使用其他值, 也可以始终引用它。

在 `IEEEeqnarray/box` 的开头, 但在局部或全局控件之前, 会发生以下初始化:

```
\lineskip=0pt
\lineskiplimit=0pt
\baselineskip=\normalbaselineskip
\jot=\IEEEnormaljot
```

因此, `\baselineskip` 设置为当前字体的间距正常值, `\jot` 恢复到其标称值, 并且 `\lineskiplimit` 系统被禁用。²⁰

该系统旨在更好地促进嵌套的 `IEEEeqnarraybox` 结构, 并帮助防止用户遇到看似无法控制的间距行为 (例如, “我如何摆脱那些不需要的间距?!”)。

1. `IEEEeqnarray Strut` 系统

在构建表格时, 尤其是具有垂直规则的表时, 通常不希望表格的行之间有额外的垂直距离, 因为这样的距离将暂停列单元定义并“切入”可能存在的任何垂直规则。然而, 必须有一种方法来保持行间隔有足够的距离。为了解决这个问题, `IEEEeqnarray/box` 命令提供了一个集成系统²¹来管理包含在每个 `IEEEeqnarray/box` 结构右端的隐藏列中的结构。

每行中的支柱将设置为默认的支柱高度和深度。通常, 默认的支柱高度和深度初始化为零, 因此实际上不会出现支柱。用户可以通过设置改变默认的支柱值

```
\IEEEeqnarraystrutsizes{height}{depth}[decl]
```

它可以放在局部或全局控件中的命令。可选参数用于在计算高度和深度参数之前执行的命令。从而

```
\IEEEeqnarraystrutsizes{0.5\baselineskip}{0}{\large}
```

将默认的支柱高度设置为 `\latge` 字体大小使用的 `baselineskip` 的一半, 即使当前 `baselineskip` (和字体大小) 不同。在可选参数中执行的命令包含在它们自己的环境中, 以便在 `\IEEEeqnarraystrutsizes` 命

¹⁸在 `IEEEtran.cls` 中, `\lineskiplimit` 和 `\lineskip` 为零 - 如果事情变得过于接近, 则作者有责任在没有 `IEEEtran.cls` 第二次猜测作者意图的情况下纠正问题。

¹⁹在 `IEEEtran.cls` 中, `\jot` 的标称值是 `\normalsize` 字体的 `\baselineskip` 的 25%。

²⁰只要行不能为负高度。

²¹“Struts”是零宽度的垂直规则, 但具有有限的高度。

令之外不会产生任何影响。为了模仿 `\baselineskip` 的动作, 通常建议的支柱的高度和深度分别是 `\normalbaselineskip`²² 的 70% 和 30%。如果 `\IEEEeqnarraystrutsizes` 高度或深度参数保留为空, 则将采用这些值。例如, 在前面的示例中, 对于 `\large` 字体, 支柱深度将设置为 `\normalbaselineskip` 的 30%。

另外

```
\IEEEeqnarraystrutsizesadd{height}{depth}[decl]
```

命令将添加到当前默认的支柱值, 并且可以像 `array.sty` 包的 `\extrarowheight` 参数一样使用。并且空参数为 `0pt`。

`\IEEEeqnarraystrutsizes` 和

`\IEEEeqnarraystrutsizesadd` 也可以在最后一列的末尾用于更改用于特定行的支柱大小 (其他行的默认支柱值不会受到影响)。

另外

```
\IEEEstrut[height][depth][decl]
```

产生了一个支柱。如果需要“手动”指定支柱, 即使在 `\IEEEeqnarray/box` 环境之外, 也可以使用它。如果未提供高度或深度参数 (或为空), 则将采用与 `\IEEEeqnarraystrutsizes` 相同的方式设置这些参数。

出于诊断目的 (为了查看任何行对象是否超过支柱的高度), 可以使用 `\IEEEeqnarray/box` 或 `\IEEEstrut` 控件放置命令 `\IEEEvisiblestrutstrue` 以使支柱的值可见。

当使用 `\IEEEeqnarraybox` 生成包含垂直线的表时, 通常需要关闭 `\baselineskip` 系统并切换到纯支柱间距。放置在局部或全局控件中的以下命令序列将用于此目的:

```
\IEEEeqnarraystrutsizes{0.7\normalbaselineskip}{0.3\normalbaselineskip}{\relax}
\setlength{\baselineskip}{0pt}%
\setlength{\lineskip}{0pt}%
\setlength{\lineskiplimit}{0pt}%
\setlength{\jot}{0pt}%
```

请注意使用 “%” 来防止以大括号结尾的行的末尾被解释为不需要的空格。由于经常需要调用此序列, `IEEEeqnarray` 系列提供了 `\IEEEeqnarraystrutmode` 命令, 它执行相同的操作。

2. 覆盖列类型

在一行中, 可以通过放置命令来覆盖一个或多个列类型

```
\IEEEeqnarraymulticol{num_cols}{col_type}{text}
```

作为单元格中的第一个命令。此命令是 `IEEEeqnarray` 等效的 `\multicolumn`。第一个参数是要覆盖的列数

²²请注意, 这不是 `normalsize baselineskip`, 而是当前字体大小的正常 `baselineskip`。

(根据需要切割任何列间间距)。第二个参数是要使用的列类型说明符。第三个参数包含单元格文本。如果列类型要将其作为参数获取,则第三个参数必须包含在一组额外的括号中,就像前面示例中的“myp” parbox 列类型(附录 F-B)所做的那样。

还有 `\IEEEeqnarrayomit` 命令,当它用作单元格中的第一个命令时,将会暂时无法使用该单元格的正常列类型。这有点像 `\IEEEeqnarraymulticol{1}{x}{}` 的快速版。

提醒用户不要使用专为其他对齐环境设计的命令(例如 `\multicolumn`)。²³

K. 预定义列类型线段

一些预定义的列类型可以生成垂直或水平线。请注意,在 `IEEEeqnarray` 系列中,线段被声明并视为普通列类型,它们不会被隐藏。虽然这种方法可能会增加用户必须跟踪的列数,特别是在创建表时,它通过允许用户覆盖或以其他方式操纵任何列类型(包括生成线段的列类型)确实提供了非常灵活的方法。

所有预定义的列线段类型都使用 `\arrayrulewidth` 长度来确定其厚度,并使用 `\doublerulesep` 来表示双重线段的间距。

“v”列类型产生垂直线段,“vv”产生两个背靠背垂直线段,它将显示为正常厚度的两倍线段。“V”产生双垂直线段,其两行之间具有 `\doublerulesep` 间距。“VV”产生两个背靠背双垂直线段,这似乎是三个垂直线段,其中中间线段是其他两个线段的两倍。通过在其列中放置间隔物可以“分开”“vv”和“VV”类型,因此它们可以用于生成两个单独或双重垂直线段,其间距是可编程的。

“h”和“H”类型分别产生单水平和双水平线段。水平线段类型通常不在列规范中使用,而是使用 `\IEEEeqnarraymulticol` 命令,以便在一个或多个列上绘制水平线段。

请注意,其他对齐环境的行命令可能无法在 `IEEEeqnarray` 系列中正常工作,该系列提供了自己执行这些类型操作的方法。特别是 `\cline` 完全不兼容,用户应该使用 `\IEEEeqnarraymulticol{num_cols}{h}{}` 命令。但是, `\vline` 和 `\hline` 应该是可以正常使用的,除非另一个 L^AT_EX 包以某种不兼容的方式重新定义它们。`IEEEeqnarray` 系列提供了自己的 `\vline` 版本: `\IEEEeqnarrayvrule[rule_thickness]`

²³那些熟悉 T_EX 的人可能会对 T_EX 的 `\omit`, `\span` 和 `\multispan` 应该在 `\IEEEeqnarraybox` 中使用,但不能在 `\IEEEeqnarray` 中使用这一事实感兴趣。因为需要在后者中使用隐藏计数器来跟踪列使用情况。

它可以生成垂直线段从单元格的顶部延伸到底部而不覆盖列类型。可选参数用于指定线段厚度,如果未提供参数,则默认为 `\arrayrulewidth`。

`IEEEeqnarray` 行命令(在下一节中讨论)提供了 `\hline` 的一些替代方法。

L. 行命令

`IEEEeqnarray` 系列有几个命令可用于生成跨越所有列的特殊行。除非另有说明,否则此处描述的命令必须作为给定行中的第一个命令发出。

要生产依赖于支柱系统的间隔排,请使用

```
\IEEEeqnarrayseprow[height][decl]
```

第一个参数指定支柱行的高度,如果保留为空或未填,则将假定默认值为 `0.25\normalbaselineskip`。第二个可选参数用于在计算第一个参数之前执行的命令,如使用 `\IEEEeqnarraystrutsizesize` 完成的那样。`\IEEEeqnarrayseprow` 不会中断列定义,因此不会使垂直线间断。如果需要列定义暂停,请使用将覆盖整行中所有列类型的切割表单:

```
\IEEEeqnarrayseprowcut[height][decl]
```

要生成水平行线段,请使用:

```
\IEEEeqnarrayrulerow[rule_thickness]
```

它将使用生成水平线段的列覆盖所有列定义。如果未指定可选线段厚度,则将使用 `\arrayrulewidth` 的值。

要生成双行线段,请使用:

```
\IEEEeqnarraydblrow[rule_thickness][spacing]
```

这将生成一个行线段,一个(非切割)分隔行,然后是另一个行线段。如果未指定可选线段厚度,则在生成两个行线段中的每一个时将使用 `\arrayrulewidth` 的值。如果未指定可选的间隔距离,则将使用 `\doublerulesep` 的值。还有一种切割形式:

```
\IEEEeqnarraydblrowcut[rule_thickness][spacing]
```

它的工作方式相同,只是分隔行将覆盖所有列定义。(垂直列线段不会出现在此命令生成的双行线段内。)

M. 有用的低层 T_EX 命令

尽管在 L^AT_EX 中通常不赞成使用较低级别的 T_EX 命令,但是其中一些命令非常有用而无法忽略。

`` 生成一个不可见的框,其中大小是其内容的宽度,高度和深度,但内容本身不会出现在输出中。还有 `\hphantom{}` 和 `\vphantom{}` 形式,它们分别仅保留内容的宽度或高度和深度。例如,请仔细查看表 V 底部的脚注。该表是使用 `\IEEEeqnarraybox` 命令生成的。脚注实际上包含在表格的最后两行中。注意脚注的左侧是如何排列的,即使第一个脚注上有一个脚

注符号的上标星号。第二行排列的原因是，在它的左侧，它采用了相同符号的 `\hphantom`：

```
\hphantom{\textsuperscript{*}}
```

垂直的 `phantoms` 可用于均衡行高或间距，例如，即使一个具有“高度”的符号而另一个没有“高度”的符号，也可以获得适合相同大小的括号内的矩阵。

`\hphantom\{}` 的反面是 `\rlap\{}`，它显示其内容，但宽度为零。还有一个 `\llap\{}` 做同样的事情，但是包含的对象将出现在给定点的左侧，而不是像 `\rlap` 一样。例如，仔细查看表 V 中的第一个“width”列标题。无论星号如何，“width”一词都居中。那是因为星号的宽度为零：

```
Width\rlap{\textsuperscript{*}}
```

`\rlap` 的垂直版是 `\smash\{}`，将其内容的表观高度和深度降低到零。（`TeX` 的 `\raisebox{0pt}[0pt][0pt]\{}` 做同样的事情，并提供可调整的垂直偏移。）`\smash` 可以在已经为对象保留空间时使用，但是那个 `TeX` 不“知道”这个并且会分配不需要的额外垂直空间。对于要被“滑入”隐藏的零高度行的表格对象，或者对于不高于“短”事物（例如水平线段）的行的 `\smash`。

可以在 `IEEEeqnarray` 系列中使用 `TeX` `\noalign\{}` 命令来注入对齐结构之外的文本。例如，

```
\begin{IEEEeqnarray}{rCl}
A_1&=&7\IEEEyesnumber\IEEEyessubnumber\
A_2&=&b+1\IEEEyessubnumber\
\noalign{\noindent and\vspace{\jot}}A_3&=&d+2\IEEEyessubnumber%
\end{IEEEeqnarray}
```

产生的结果是：

$$A_1 = 7 \tag{12a}$$

$$A_2 = b + 1 \tag{12b}$$

and

$$A_3 = d + 2 \tag{12c}$$

使用时，`\noalign` 必须一行中的第一个命令，甚至在任何 `\IEEEeqnarraymulticol`，`\IEEEeqnarrayomit` 以及 `row` 命令之前。

预先提醒正确使用 `\noalign` 可能会很棘手。有三个潜在的问题。

- 1) 记住 `\noalign` 会将其内容放在 `alignment` 之外。因此，`IEEEeqnarray` 命令的行间距控制将不起作用。用户可能必须根据需要手动添加 `\baselineskip` 或 `\jot` 间距（这在前面的示例中已完成）。
- 2) 此外，`\noalign` 不会自动将其内容放在一个框中。但是，当在 `\IEEEeqnarraybox` 命令生成的垂直框

内时，必须将未对齐的部分放在水平框内。因此，在 `\IEEEeqnarraybox` 中使用 `\noalign` 时，请务必将内容包装在 `\hbox\{}` 中：²⁴

```
\noalign{\hbox{and therefore}}
```

- 3) 最后，可能存在一些与 `\noalign` 行周围容易发生分页有关的问题。这只是 `IEEEeqnarray` 的一个问题，因为在 `\IEEEeqnarraybox` 生成的框内不会发生分页。如果需要，可以通过在 `\noaligncontents` 的末尾手动输入 `\pagebreak` 或 `\nopagebreak` 等来更改分页行为。

N. 更实用的使用范例

`IEEEeqnarray` 的使用有点复杂。然而，一旦理解了构建块和核心概念，用户可能会发现更容易使用 `IEEEeqnarray` 系列来处理几乎所有的对齐情况，而不是必须记住许多不同工具的所有接口和独特行为。

现在将演示一些“现实世界”的例子。

1) `IEEEeqnarray` 分段结构：分段结构可以使用 `\IEEEeqnarraybox` 获得：

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \\ -x, & \text{for } x < 0 \end{cases} \tag{13}$$

这是使用以下代码生成的：

```
\begin{equation}
\setlength{\nulldelimiterspace}{0pt}
|x|=\left\{\begin{array}{l}
x,&\text{for } \$x \geq 0\$\\
-x,&\text{for } \$x < 0\$
\end{array}\right.
\end{IEEEeqnarraybox}\right.
\end{equation}
```

注意在条件语句之前使用大的 `\quad (1em)` 间距。`\nulldelimiterspace` 的归零是一个可选步骤，消除了不可见的闭合支撑“`\right`”的宽度。“为了使等式的可见部分完美居中。”²⁵

请注意，两个分支共享一个公共公式编号。如果每个分支都需要一个公式（子）编号，首选的解决方案是使用第 IX-A 节中讨论的 `cases.sty` 包。但是，可以使用 `\IEEEeqnarray` 构建这样的东西，虽然这需要额外的工作和一些技巧。例如，

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \\ -x, & \text{for } x < 0 \end{cases} \tag{14a}$$

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \\ -x, & \text{for } x < 0 \end{cases} \tag{14b}$$

是使用代码生成的：

```
\begin{IEEEeqnarray}{\setlength{\nulldelimiterspace}
{0pt}}{r1's}
&x,&\text{for } \$x \geq 0\$ \IEEEyesnumber \IEEEyessubnumber
```

²⁴`TeX` 的 `\mbox` 无效！

²⁵`null` 分隔符的宽度通常只有 1.2pt，因此通常可以安全地忽略。

```

\\*[-0.625\normalbaselineskip]
\smash{|x|=\left\{\IEEEstrut[3\jot][3\jot]\right.}&&
\nonumber\\*[-0.625\normalbaselineskip]
&-x,&for $x < 0$\IEEEyessubnumber
\end{IEEEeqnarray}

```

隐藏的中间行用于保持相等的左侧。为了防止这一行改变两个分支之间的间距，必须将其高度除去，并且额外的行间距 (由 `\baselineskip` 加上 `\jot` 组成，对于 `IEEEtran.cls` 通常为 `0.25 \baselineskip`。) 必须是从上方移出一半，从下方移开一半，使它看起来好像中间一排从未没有一样。因为大括号不会受到内部分支的影响，所以必须用支柱手动调整大小。新行命令的星形形式用于防止结构内分页的可能性。

2) 矩阵: 可以使用 `\IEEEeqnarraybox` 轻松创建矩阵:

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (15)$$

这个例子的代码很简单:

```

\begin{equation}
I = \left(\begin{IEEEeqnarraybox*}[] [c]{,c/c/c,}
1&0&0\\
0&1&0\\
0&0&1
\end{IEEEeqnarraybox*}\right)
\end{equation}

```

因为示例矩阵具有正常高度的元素，所以可以使用 `\IEEEeqnarraybox` 的星形模式来关闭行间距的额外 `\jot` 分量，以便形成更紧凑的矩阵。如果在矩阵中使用了更大的符号，则非星形模式将是更好的选择。`\arraycolsep` 通常用作元素列分隔符。标准的小数学间距被添加到矩阵的末端，以在它与其括号括起来之间提供一点距离。

展示如何构建“小”矩阵是有益的²⁶,

$$S = \begin{bmatrix} 1/2 & 0 \\ 0 & 3/4 \end{bmatrix} \quad (16)$$

这是通过

```

\newcommand{\mysmallarraydecl}{\renewcommand{
\IEEEeqnarraymathstyle}{\scriptscriptstyle}%
\renewcommand{\IEEEeqnarraytextstyle}{\scriptsize}%
\renewcommand{\baselinestretch}{1.1}%
\settowidth{\normalbaselineskip}{\scriptsize
\hspace{\baselinestretch\baselineskip}}%
\setlength{\baselineskip}{\normalbaselineskip}%
\setlength{\jot}{0.25\normalbaselineskip}%
\setlength{\arraycolsep}{2pt}}
%
\begin{equation}
S=\left[\begin{IEEEeqnarraybox*}[\mysmallarraydecl]
[c]{,c/c,}

```

²⁶IEEE 作者应该注意不建议使用小矩阵，因为 IEEE 通常不会减少方程中的字体大小或改变主文本 `baselineskip` 以适应文本数学。

表 VI
NETWORK DELAY AS A FUNCTION OF LOAD

β	Average Delay	
	λ_{\min}	λ_{\max}
1	0.057	0.172
10	0.124	0.536
100	0.830	0.905*

*limited usability

```

1/2&0\\
0&3/4%
\end{IEEEeqnarraybox*}\right]
\end{equation}

```

生成。使用用户定义的命令 `\mysmallarraydecl` 来包含 `IEEEeqnarray` 设置代码，演示了用户如何通过充分利用 `IEEEeqnarray` 系列的动态可配置性来轻松地重建其最常用的结构。

为了演示一些技术，这个例子比需要的更复杂。将 `\baselineskip` 设置为所需的值会很容易，但是假设矩阵行与 `\scriptsize` 字体的 `\baselineskip` 间隔为多个。更复杂的是，大多数 `TeX` 类文件不允许用户在数学模式下执行设置文本字体大小命令，而矩阵在公式内。因此，`\scriptsize` 不能用于直接设置 `\baselineskip`。

第一步是将数学和文本列设置为所需的样式。那么 `\baselinestretch` 就可以像 `\arraystretch` 一样使用。诀窍是在 `\settowidth` 命令中运行 `\scriptsize`，该命令存储 `\scriptsize` 字体的 `\baselineskip` 乘以 `\normalstlinesch`。在 `\normalbaselineskip` 中，然后用于设置 `\baselineskip`, `\jot` 等。最后，`\arraycolsep` 减少到更适合较小的字体。注意使用“%”来防止在 `\mysmallarraydecl` 中的行尾处的大括号后出现不需要的空格。

3) 表格: 表格，特别是那些有线条的表格往往稍微复杂一些。表 VI 使用以下代码制作:

```

\begin{table}[!t]
\centering
\caption{Network Delay as a Function of Load}
\label{table_delay}
\begin{IEEEeqnarraybox}[\IEEEeqnarraystrutmode
\IEEEeqnarraystrutsadd{2pt}{0pt}]{x/r/Vx/r/v/r/x}
\IEEEeqnarraydbrulerowcut\
&&&\IEEEeqnarraymulticol{3}{t}{Average Delay}&&
&\hfill\raisebox{-3pt}[0pt][0pt]{\beta}&\hfill&&
\IEEEeqnarraymulticol{5}{h}{\hfill}
\IEEEeqnarraystrutsadd{0pt}{0pt}\
&&&\hfill\lambda_{\mbox{min}}&\hfill&&\hfill
\lambda_{\mbox{max}\vphantom{i}}&\hfill&
\IEEEeqnarraystrutsadd{0pt}{2pt}\
&1&& 0.057&& 0.172&\
\end{IEEEeqnarraydbrulerowcut\

```

表 VII
POSSIBLE Ω FUNCTIONS

Range	$\Omega(m)$
$x < 0$	$\Omega(m) = \sum_{i=0}^m K^{-i}$
$x \geq 0$	$\Omega(m) = \sqrt{m}$

```
&10&&& 0.124&& 0.536&&\\
&100&&& 0.830&& 0.905&&\rlap{\textsuperscript{*}}&&\\
\IEEEeqnarraydbrulerowcut\\
&\IEEEeqnarraymulticol{7}{s}
{\scriptsize\textsuperscript{*}limited usability}%
\end{IEEEeqnarraybox}
\end{table}
```

因为此表有线段，所以第一步是启用支柱模式行间距。然后将支柱高度增加一些，以在字母上方提供更多的空间。²⁷ 此表使用切割水平线段和开放边，这在 IEEE 出版物中通常的做法。有三个额外的 ‘x’ 列作为占位符。每一端的 ‘x’ 列用作快速方法，使水平线段稍微延伸超过表的内容。中间的 ‘x’ 列用作水平线段的“平均延迟”以下的附着点。没有这个额外的列，该水平线段的左侧将切入中间双垂直线段。²⁸ 请注意 “ β ” 如何作为包含水平线段的行的一部分被吞没。必须消除 β 的高度，以免增加不需要的垂直间距。同样，该行的支柱被禁用。此外，使用 `\raisebox` 而不是 `\smash`，以便 β 可以垂直降低，否则它将出现在其基线上，这对于手头的目的来说太高了。 β 两侧的 `\hfill` 将该单元格的对齐方式改为居中。“min” 和 “max” 下标通常不会处于同一级别，因为 min 中的 “i” 略高于 “max” 中的字母。为了解决这个问题，将 `\vphantom{i}` 添加到 “max”。因为这些下标非常低，所以该线的支柱深度增加了几个点。或者，人们可能只是去除了 “i” 的高度。“0.905” 旁边的星号通过 `\rlap` 减小到零宽度，这样它就不会影响其单元格的宽度或对齐方式。

对于包含高符号的行，支柱间距不能很好地工作，因为这些对象通常会超出支柱的高度。此外，增加支柱高度通常不是一种好选择，因为

- 1) 必须测量或猜测高符号的高度和深度;
- 2) 可能有其他行具有正常的行高。

表 VII 说明了这种情况。它的代码如下所示:

```
\begin{table}[!t]
\centering
\caption{Possible  $\Omega$  Functions}
\label{table_omega}
\begin{IEEEeqnarraybox}
```

²⁷Knuth 称这一额外步骤是质量的标志。

²⁸Some may even think it would be better that way, but we want to show some tricks in these examples.

```
[\IEEEeqnarraystrutmode\IEEEeqnarraystrutsizadd
{2pt}{1pt}]{v/c/v/c/v}
\IEEEeqnarrayrulerow\\
&\mbox{Range}&&\Omega(m)&&\\
\IEEEeqnarraydbrulerow\\
\IEEEeqnarraysepro[3pt]\\
&x < 0&&\Omega(m)=\sum\limits_{i=0}^mK^{-i}
&\IEEEeqnarraystrutsiz{0pt}{0pt}\\
\IEEEeqnarraysepro[3pt]\\
\IEEEeqnarrayrulerow\\
\IEEEeqnarraysepro[3pt]\\
&x \ge 0&&\Omega(m)=\sqrt{m}&&\hfill&
\IEEEeqnarraystrutsiz{0pt}{0pt}\\
\IEEEeqnarraysepro[3pt]\\
\IEEEeqnarrayrulerow
\end{IEEEeqnarraybox}
\end{table}
```

解决方案是使用 `\IEEEeqnarraysepro` 根据需要手动添加固定数量的额外空间。通过这种方式，`\IEEEeqnarraysepro` 可以为有线段的表做 `\jot` 对多行公式做的事。当然，使用此方法，行的基线将不再是等间距。

平方根单元格中的 `\hfill` 是一种简单但有效的方法，可以在不需要额外列的情况下使等号符号排列。

致谢

作者要感谢 Ken Rawson , Kevin Lisankie , Kimberly Sperka , Steve Wareham , Patrick Kellenberger , Laura Hyslop 和 IEEE 的 Cathy Cardon , 感谢他们帮助和支持使这项工作成为可能。Donald Arseneau , Fred Bartlett , David Carlisle , Tony Liu , Frank Mittelbach , Piet van Oostrum , Roland Winkler 和 Mark Wooding 等 TeX 大师的知识和先前工作在开发复杂的 IEEEeqnarray 系列命令方面发挥了重要作用。作者也感谢 Peter Wilson 和 Donald Arseneau 允许包含他们的 `\@ifmtarg` 命令。

最后，如果不是 Gerry Murray , Silvano Balemi , Jon Dixon , PeterNüchter 和 Juergen von Hagen 这些以前的 IEEEtran 开发人员的努力，这项工作可能是不可能的完成。他们的工作某种程度上仍然在 IEEEtran 上存在。

参考文献

- [1] “(2015, jul.)the ieee website.” [Online]. Available: <https://www.ieee.org/>
- [2] “M. shell. (2015, aug.)the ieeetran.cls package.” [Online]. Available: <http://www.michaelshell.org/tex/ieeetran/>
- [3] “— . (2015, jul.) ieeetran homepage.” [Online]. Available: <http://www.michaelshell.org/tex/ieeetran/>
- [4] H. Kopka and P. Daly, “Guide to latex (ed),” 2004.
- [5] “T. oetiker, h. partl, i. hyna, and e. schlegl. (2015, jul.) the not so short introduction to L^AT_EX_{2 ϵ} .” [Online]. Available: <https://www.ctan.org/pkg/lshort>
- [6] S. M. Moser, “How to typeset equations in latex.” [Online]. Available: <http://moser.cm.nctu.edu.tw/manuals.html#eqlatex>

- [7] “R. fairbairns. (2014, jun.) The T_EX FAQ.” [Online]. Available: <https://texfaq.org/>
- [8] “M. sharpe. (2015, jul.) the newtx package.” [Online]. Available: <https://www.ctan.org/pkg/newtx>
- [9] “(2015, jul.) mathtime professional fonts. personal t e x, inc.” [Online]. Available: <https://pctex.com/mtpro2.html>
- [10] “D. carlisle and f. mittelbach. (2015, apr.) the bm package.” [Online]. Available: <https://www.ctan.org/pkg/bm>
- [11] “(2013, jan.) the amsmath package. the american mathematical society.” [Online]. Available: <https://www.ctan.org/pkg/amsmath>
- [12] “S. pakin. (2009, apr.) the ieeeconf.cls package.” [Online]. Available: <https://www.ctan.org/pkg/ieeeconf>
- [13] “J. d. mccauley, j. goldberg, and a. sommerfeldt. (2011, dec.) the endfloat package.” [Online]. Available: <https://www.ctan.org/pkg/endfloat>
- [14] “H. oberdiek. (2012, may) the ifpdf package.” [Online]. Available: <https://www.ctan.org/pkg/ifpdf>
- [15] A. Gefen, “Simulations of foot stability during gait characteristic of ankle dorsiflexor weakness in the elderly,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, no. 4, pp. 333–337, 2001.
- [16] “D. arseneau. (2015, mar.) the cite package.” [Online]. Available: <https://www.ctan.org/pkg/cite>
- [17] “M. d. wooding. (1999, mar.) the mdw tools package.” [Online]. Available: <https://www.ctan.org/pkg/mdwtools>
- [18] “D. arseneau. (2010, feb.) the cases package.” [Online]. Available: <https://www.ctan.org/pkg/cases>
- [19] “S. tolušis and v. statulevičius. (2013, oct.) the stfloats package.” [Online]. Available: <https://www.ctan.org/pkg/stfloats>
- [20] “D. carlisle. (2015, apr.) packages in the ‘graphics’ bundle. grfguide.pdf” [Online]. Available: <https://www.ctan.org/pkg/graphics>
- [21] “K. reckdahl. (2006, jan.) using imported graphics in L^AT_EX 2_ε.” [Online]. Available: <https://www.ctan.org/pkg/epslatex>
- [22] “C. barratt, m. c. grant, and d. carlisle. (1998, may) the psfrag package.” [Online]. Available: <https://www.ctan.org/pkg/psfrag>
- [23] “S. d. cochran. (2005, jul.) the subfigure package.” [Online]. Available: <https://www.ctan.org/pkg/subfigure>
- [24] “S. d. cochran, v. karen-pahlav, z. mehran, and v. khalighi. (2005, jul.) the subfig package.” [Online]. Available: <https://www.ctan.org/pkg/subfig>
- [25] “A. sommerfeldt. (2013, may) the subcaption package.” [Online]. Available: <https://www.ctan.org/pkg/subcaption>
- [26] “P. williams and r. brito. (2009, aug.) the algorithmic package.” [Online]. Available: <https://www.ctan.org/pkg/algorithmic>
- [27] “S. jános. (2005, apr.) the algorithmicx.sty package.” [Online]. Available: <https://www.ctan.org/pkg/algorithmicx>
- [28] “F. mittelbach and d. carlisle. (2015, apr.) the array package.” [Online]. Available: <https://www.ctan.org/pkg/array>
- [29] “D. arseneau. (2010, mar.) the threeparttable package.” [Online]. Available: <https://www.ctan.org/pkg/threeparttable>
- [30] “D. carlisle. (1999, apr.) the fix2col package.” [Online]. Available: <https://www.ctan.org/pkg/fix2col>
- [31] “M. høgholm. (2012, dec.) the dblfloatfix package.” [Online]. Available: <https://www.ctan.org/pkg/dblfloatfix>
- [32] “M. shell. (2015, aug.) the ieetrans bibtex style.” [Online]. Available: <https://www.ctan.org/pkg/ieetrans>
- [33] “P. w. daly. (2013, may) the balance package.” [Online]. Available: <https://www.ctan.org/pkg/balance>
- [34] “S. tolušis and v. statulevičius. (2015, apr.) the flushend package.” [Online]. Available: <https://www.ctan.org/pkg/flushend>
- [35] “M. shell. (2007, jan.) the testflow diagnostic suite.” [Online]. Available: <https://www.ctan.org/pkg/testflow>
- [36] “T. oetiker. (2015, mar.) the acronym package.” [Online]. Available: <https://www.ctan.org/pkg/acronym>
- [37] “D. arseneau. (2013, dec.) the url package.” [Online]. Available: <https://www.ctan.org/pkg/url>
- [38] “M. shell. (2015, aug.) the ieetrantools package.” [Online]. Available: <https://www.ctan.org/pkg/ieetrantools>



Michael Shell (M⁸⁷) received the B.E.E., M.S.E.E. and Ph.D. degrees in electrical engineering all from the Georgia Institute of Technology, Atlanta, in 1991, 1993 and 2004 respectively. He has developed several all-optical packet-switched network subsystems and node demonstrations. His research interests include all-optical packet-switched networks, high speed optoelectronic interface design, discrete simulation and exact Markov models for buffered packet switches.

Dr. Shell is also the author of the most recent versions of the IEEEtran L^AT_EX class and BmL^AT_EX style packages and is the current maintainer of both.