

tabu 宏包说明文档*

xiaodaxialiang[†]

2011 年 8 月 22 日

说明

tabu 宏包是制作表格的宏包, 相对于我们常见的 array, delarray, longtable, colortbl, multirow, booktabs 有很强的可操作性. 比如单元格的格式设置, 横线竖线的设置, 表格颜色的设置, 表格内公式的使用, 脚注的使用, 抄录环境的使用, …… tabu 几乎都可以完成.

目录

1	tabu 环境	3
1.1	tabu, tabu to 和 tabu spread	3
1.2	longtabu, longtabu to 和 longtabu spread	4
1.3	tabu 的 X 列模式—调整水平距离	4
1.4	tabulinesep 和 extrarowsep—控制垂直距离	8
1.5	tabu 的数学公式模式	9
2	线条和颜色的设置	11
2.1	竖线: 有可选项	11
2.2	多样式的 firsthline 和 lasthline	11
2.3	线条的更多设置	12
2.4	横线格式的重复和行的颜色设置	15
3	修改字体和每行的对齐方式	18
4	保存和使用 tabu 数据	18

*适用于 tabu 2.8 版本.

[†]xiaodaxialiang@163.com

5	其他功能	21
5.1	使用抄录环境	21
5.2	制定新环境	21
5.3	对齐数字	22
6	宏包选项	22
6.1	debugshow 选项	22
6.2	delarry 选项	22
6.3	linegoal 选项	22
7	总结	23

1. tabu 环境

1.1. tabu, tabu to 和 tabu spread

```
\begin{tabu}[pos]{tabular preamble}
\begin{tabu}to<dimen>[pos]{tabular preamble}
\begin{tabu}spread<dimen>[pos]{tabular preamble}
```

tabu 环境类似于 tabular, {tabular preamble} (后文简称为表格导言区) 的设置也是通过 array.sty 宏包来完成的, 并且在输出表格之前需要设置一些变量或尺寸. tabu 环境改进了 tabular 和 array 宏包:

- 可以在 tabu 中设置脚注和索引, 通过 hyperref 设置的脚注链接不会被阻止. 也可以在 tabu 和 langtabu 环境中使用 \footnote[number]{<text>} 语法.
- X 列 的语法改进为含有如下可选参数的模式: 宽度 (可以是负数), 水平对齐方式, 右, 中, 左, 自适应 (r, c, l, j, 还可以是通过 ragged2e 设置的 R, C, L, J), 垂直对齐方式, 上, 中, 下 (p, m, b). 另外 [pos] 可以设置为 t, c, b, 分别表示表格与文本段落顶部, 中部, 底部对齐.

tabu 在 X 列模式下有一个默认目标宽度 (default target width), 这使得表格嵌套时设置很简单.

- 大家习惯在文字模式中使用 tabular, 而在数学公式模式中使用 array, 但是 tabu 既可以在文字模式中也可以在数学公式模式中使用. tabu 的 X 列也可以在数学公式模式中使用.
- tabu 环境可以兼容其他任何形式的表格: tabular, tabular*, tabularx 甚至 tabu 自身, 而且可以放在 tabu 中的任何一个单元格里. 同样地, tabu 也可以在 tabular, tabularx 等环境中使用.
- tabu 提供了设置横线和竖线格式的命令, 而且可以在 X 列 模式中使用抄录环境 (verbatim text).
- tabu 完全兼容 color, delarray, hline, makecell, booktabs, siunitx, dcolumn, warpcol 等宏包. 在 tabu 环境中你可以不需要担心 \arraybackslash 命令, 随意使用 \raggedleft, \raggedright 以及 \centering, 在 X 列 中 “\” 有正常的含义, 即换行.

`\begin{tabu}to<dimen>` 类似于 `tabular*`, 但是其内部列的间距给定了一个 1fl 的伸展距离. 如果想消除此伸展间距需通过 `\extracolsep` 命令设定相应的值. 所以 `tabu to` 可以使表格占满 `<dimen>` 设定的宽度.

`\begin{tabu}spread<dimen>` 输出一个比表格自然宽度大 `<dimen>` 距离的表格. 自然宽度就是我们不对表格宽度设定数值时表格自适应的宽度.

1.2. longtabu, longtabu to 和 longtabu spread

```
\begin{longtabu}[l|c|r]{tabular preamble}
\begin{longtabu}to<dimen>[l|c|r]{tabular preamble}
\begin{longtabu}spread<dimen>[l|c|r]{tabular preamble}
```

`longtabu` 与 `tabu` 一样, 但是允许表格的行换页. `longtabu` 是基于 `longtable` 宏包开发的, 所以 `longtable` 宏包需要加载. `longtable` 环境的所有功能都可以在 `longtabu` 中使用: `\endhead`, `\endfirsthead`, `\endfoot`, `\endlastfoot` 和 `\caption`.

下面所列的命令目前只支持 `tabu` 环境, 并不支持 `longtabu` 环境.

tabu 命令	不可用	未开发	说明
<code>\tabucline</code>		✓	<code>\tabucline</code> 目前还不支持分页, 使用 <code>\hline</code> 代替
<code>\usetabu</code>	✗		但是可以使用 <code>\savetabu</code> 和 <code>\preamble</code>
数学模式	✗		<code>longtable</code> 不支持数学模式
<code>delarray</code> 的快捷命令	✗		符号不能跨页
<code>\tabuphantomline</code>	✗		在 <code>longtabu</code> 内部可用

然而, `X` 列, `\rowfont`, `\extrarowsep`, `\tabulinesep`, `\tabudecimal`, `\tabucline` (分页时有限制), `\taburulecolor`, `\tabulinestyle`, `\taburowcolors`, `\preamble`, `longtabu to`, `longtabu spread` 在 `longtabu` 环境内部均可以使用.

1.3. tabu 的 X 列模式—调整水平距离

`tabu X` 列可以看作是 `tabular X` 列模式的改进, 但二者不会相互影响.

宽度参数 (`width coefficient`) 在 `X` 列模式中是可选项

例如: $X[2.5]X[1]$ 命令同 $X[2.5]X$, $X[5]X[2]$ 效果相同, 参数的设定只与最后的比例有关. 上面三个命令的意思是, 第一列宽度是第二列宽度的 2.5 倍. 也可以理解为第一列宽度是总宽度的 5/7.

$X[2.5]$	X
----------	-----

负宽度参数 (negative width coefficients) 可以在 X 列模式中使用:

例如, $X[-2.5]X[1]$, $X[-2.5]X$, $X[-5]X[2]$.

上面命令的意思是, 第一列宽度至多是第二列宽度的 2.5 倍, 如果第一列的自然宽度比 $2.5 \times$ (第二列宽度) 小, 那么第一列的宽度将会自动缩减到其自然宽度. 下面的表格是使用相同的表格设置命令生成的:

`\begin {tabu} to\linewidth {\lX[2.5c]\lX[c]\}`

$X[-2.5]$	X
此单元格的自然宽度比第二列宽度的 2.5 倍大, 所以其最后的宽度至多是第二列的 2.5 倍.	X

水平对齐 水平对齐方式的设置通过命令 $X[5,r]X[2,c]$ 很容易完成. 而垂直对齐方式可以通过 $X[5,r,m]X[2,p,c]$ 来完成. 注意中间的逗号可省略, 还有数字和对齐方式的顺序可以颠倒, 但是不要让 $\text{T}_{\text{E}}\text{X}$ 产生误会, 例如 $X[1\text{cm}]$, 此时就应该主动加上逗号, $X[1,c,m]$. 另外, 需要稍微注意的是 m, p, b 命令与 $\backslash\text{parbox}$ 的 m, p, b ¹.

命令	格式	默认
l, c, r, j, L, C, R, J	左对齐, 居中, 右对齐, 自适应	j
p, m, b	X 列模式下转换为 p, m, b 模式	P
$\$$	$X[\$]$ 相当于 $>\{\$\}X<\{\$\}$	
$\$\$$	$X[\$\$]$ 相当于 $>\{\$\displaystyle\}X<\{\$\}$	

- $\text{tabu } X$ 列可以通过 $\backslash\text{multicolumn}$ 合并单元格.
- $\text{tabu } X$ 列可以和 tabu spread 一起使用来制作小表格.
- $\text{tabu } X$ 列可以含有 tabular , tabular* , tabularx 或者 tabu , 仍然保持每个宏包的原有命令, 没有语法冲突. tabu 也可以放在 tabular , tabular* , tabularx 环境之中. 对于嵌套的 tabu 环境, 全局默认的对齐方式是 t (即顶部对齐), 而在文本段落中默认的对齐方式是 c (即居中).

¹详细解释可以参考 <http://tex.stackexchange.com/questions/26212/>.

X 列模式和“tabu spread”

tabu X 列模式可以和“tabu spread”一起使用来调节文字比较少的表格中列的宽度. 学习了上面的内容, 可以尝试操作: 制作一个与页面同宽的 6 列表格, 其中第 1, 2, 5 列和第 6 列同宽, 而第 3 和第 4 列是其余表格宽度的一半.

```
\begin{tabu} to\linewidth{|X[2]|X[2]|X|X|X[2]|X[2]|} \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

上面的表格中文字很短, 每个单元格只有一个字符, 这就显得每个单元格太宽. 如果想使每个单元格变得紧凑, 而不知道表格的总宽度, 可以使用如下代码:

```
\begin{tabu} spread 0pt{|X[2]|X[2]|X|X|X[2]|X[2]|} \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

但是上面得到的表格又太窄了, 那么可以再加一点间距:

```
\begin{tabu} spread 2in{|X[2]|X[2]|X|X|X[2]|X[2]|} \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

tabu spread 对宽度很长的表格不起作用. 下面的表格是通过下面的代码设置的:

```
\begin{tabu} spread 3cm{@{}X[9]X[4]|X|}
```

”Like the air we breathe, Sherlock Holmes is everywhere. His pipe-smoking, deer stalkered image peers at us from ads in Yellow Pages, to signs for neighborhood crime-watch; from billboards to the classroom; from film and television to the public library, and now over the Internet. He long ago transcended the boundaries of 19th Century London² to become an international best-seller and has been accepted as part of British folklore. Holmes is alive to millions.”

There the text was too long, and **tabu spread** behaves as if you didn’t give it a target.

The result of this example is the same as if one had written.

Sherlock Holmes

The “official” web site: <http://www.sherlockholmes.com/>

²这里带了一个脚注.

在表格环境中 @{} 表示忽略页边的空白 (margin).

X 列的负宽度参数

```
\tabulinestyle{3pt ForestGreen}
\begin{tabu}{|X[-1m]|X[c m]|}
    \tabucline- \savetabu{FirstNegativTest}
    $\begin{tabu}({X[-1$]X[-1$c]})
        \alpha & \beta \\
        \gamma & \delta + \epsilon + \zeta + \eta + \theta
    \end{tabu}$
&
    This is a tabu with negative width coefficients for \texttt{X}
    columns
    \\ \tabucline-
\end{tabu}
```

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta + \epsilon + \zeta + \eta + \theta \end{pmatrix}$$

This is a tabu with negative width coefficients for X columns

tabu 中的合并单元格

```
\tabuphantomline
```

在 tabu 环境中不能简单地使用 \multicolumn 来合并单元格, 还必须在表格命令结束前 (即 \end{tabu} 之前) 添加命令 \tabuphantomline. 这与 \multicolumn 和 T_EX 的原始命令 \omit 有关系. 如下例:

```
\begin{tabu}{|X|X|X[2]|} \tabucline-
    \multicolumn2{|c|}{Hello} & World \\ \tabucline-
    \tabuphantomline
\end{tabu}
```

Hello	World
-------	-------

你可以尝试将上述代码中的 \tabuphantomline 去掉再重新编译. 虽然在 tabu 中的任意一行都可以使用 \tabuphantomline, 最好还是在表格代码的最后使用. 因为在 tabu 有嵌套时 \tabuphantomline 可能影响垂直对齐的效果.

特别需要注意, `\tabuphantomline` 后面不能存在 `\cr`, `\` 和 `\tabularnewline` 等命令. 宏包制作者正在试图将此代码去掉不用, 正在改进 `\multicolumn` 命令, 应该在 2.9 版本中可以实现.

1.4. `\tabulinesep` 和 `\extrarowsep`—控制垂直距离

```
\tabulinesep=<dimen>
\tabulinesep=^<dimen>
\tabulinesep=_<dimen>
\tabulinesep=^<dimen>_<dimen>
\tabulinesep=_<dimen>^<dimen>
```

`\tabulinesep` 命令用来设置单元格内容和单元格边界之间垂直方向的最小距离. 定义 `\tabulinesep` 命令时使用了 `\global` 作为前缀³.

可以使用 `\tabulinesep` 命令对单元格的“顶部距离”和“底部距离”进行单独设置, 需要使用如下语法:

```
\tabulinesep=^<dimen>      设置 \abovetabulinesep
\tabulinesep=_<dimen>      设置 \belowtabulinesep
\tabulinesep=_<dimen>^<dimen> 设置 \belowtabulinesep 和 \abovetabulinesep.
```

这些参数可以在文本模式中使用, 也可以在数学模式中使用, 以增加行之间的距离, 尤其是在输入数学公式的时候.

如下例中我们可以看到效果. 此例使用了 `interfaces` 宏包中的 `\papergraduate` 来做表格的背景以使效果更加明显, 而且设置了参数 `\tracingtabu=3`.

$\frac{d\Phi}{\theta}$	$\frac{d\theta}{t}$	$\frac{d\Phi}{\theta}$	$\frac{d\theta}{t}$	$\frac{d\Phi}{\theta}$	$\frac{d\theta}{t}$
<code>\tabulinesep = 0mm</code>	<code>\tabulinesep = 1mm</code>	<code>\tabulinesep = 3mm</code>			

`\tabulinesep` 是一个“软参数”(soft parameter), 可能导致行与行之间有不同的行高.

```
\extrarowsep=<dimen>
\extrarowsep=^<dimen>
\extrarowsep=_<dimen>
```

³`\tabulinesep` 并不是一个尺寸.

`\extrarowsep=^<dimen>_<dimen>`

`\extrarowsep=_<dimen>^<dimen>`

`\extrarowsep` 是每行在垂直方向上的额外空白. `array.sty` 给 $\text{T}_{\text{E}}\text{X}$ 提供了 `\extrarowheight` 这一尺寸, 而 `tabu` 提供了 `\extrarowdepth`.

这样, 每行就可以得到相同的高度或者深度, 增加的空白是固定的. `\extrarowsep` 甚至可以使用赋予正值的 `\tabulinesep`, 因为 `tabu` 给每行只增加一个支柱.

`\extrarowsep` 也用 `\global` 做了前缀. 通过如下语法可以分别设置 `\extrarowheight` 和 `\extrarowdepth` 的具体数值:

`\extrarowsep=^<dimen>` 设置 `\extrarowheight`

此时 `\extrarowdepth` 没有变

`\extrarowsep=_<dimen>` 设置 `\extrarowdepth`

此时 `\extrarowheight` 没有变

`\extrarowsep=_<dimen>^<dimen>` 设置 `\extrarowdepth` 和 `\extrarowheight`.

`\extrarowheight` 和 `\extrarowdepth` 都能够被 `array.sty` 宏包中的 `\arraystretch` 因子调整. 这些参数同样能够在文本和数学模式中使用.

X[c]	X[-1,c]	X[\$]	X[-1c]X[-1c\$]
One Two	One Two	α β	First $\frac{\partial \Phi}{\partial \theta}$
Three Four	Three Four	$\frac{\Phi}{\theta}$ Γ_x^t	Second $\frac{d\theta}{dt}$
<code>\extrarowsep = 3mm</code>			
X[c]	X[-1,c]	X[\$]	X[-1c]X[-1c\$]
One Two	One Two	α β	First $\frac{\partial \Phi}{\partial \theta}$
Three Four	Three Four	$\frac{\Phi}{\theta}$ Γ_x^t	Second $\frac{d\theta}{dt}$
<code>\extrarowsep = 0mm</code>			

1.5. tabu 的数学公式模式

在下面粉红色框表格的例子中, 使用的 `tabu` 的表格设置为:

`\begin{tabu}to\linewidth{X[m]X[-1$ $m]}.`

在右侧你可以看到著名的倒向随机微分方程 (简称 BSDE), 此方程的非线性形式由中国彭实戈院士和法国 Pardoux 院士共同提出, 并求得了解的存在唯一性.

$$y_t = \xi + \int_t^T g(s, y_s, z_s) ds - \int_t^T z_s dB_s$$

再给一个稍微复杂点的例子, 对比一下效果.

```
\begin{tabu}to \linewidth{*3{X[c]}}
\toprule
\texttt{array} & \texttt{tabu} & \texttt{tabu spread} 1em \\\
\midrule
$\begin{array} [t] |{cc}|$
\alpha & \beta \\\
\gamma & \delta
\end{array}$
&
$\begin{tabu} |{cc}|$
\alpha & \beta \\\
\gamma & \delta
\end{tabu}$
&
$\begin{tabu} spread 1em |{cc}|$
\alpha & \beta \\\
\gamma & \delta
\end{tabu}$\\
\bottomrule
\end{tabu}
```

array	tabu	tabu spread 1em
$\begin{array} [t] \alpha & \beta \\ \gamma & \delta \end{array}$	$\begin{tabu} \alpha & \beta \\ \gamma & \delta \end{tabu}$	$\begin{tabu} spread 1em \alpha & \beta \\ \gamma & \delta \end{tabu}$

在上面的例子中, `*3{X[c]}` 表示三个相同的 `X[c]`, 即 `X[c]X[c]X[c]`; 第一个矩阵是通过 `array` 实现, 而且使用了 `delarray` 宏包实现左右两条竖线; 第二个和第三个矩阵通过 `tabu` 实现, 左右两条竖线就是表格的左右线. 大表格的三线式形式, 是通过 `booktabs` 宏包的 `\toprule`, `\midrule`, `\bottomrule` 实现的.

2. 线条和颜色的设置

2.1. 竖线: | 有可选项

在 `tabu` 环境中, 竖线 `|` 有一个可选项, 可以用来设置表格竖线的宽度. 默认的宽度为 `\arrayrulewidth`. 此可选项中还可以包含颜色, 只能使用颜色的名称, 而非颜色模式. 而且宽度必须在颜色名称之前. 例如 `|[5pt,red]`. 中间的逗号在不让 $\text{T}_{\text{E}}\text{X}$ 产生歧义的情况下可省略也可以换成空格.

	左侧的效果由右侧的代码生成.	<pre>\begin {tabu}{ [[5pt] c c [5pt] } Hello & World \end {tabu}</pre>
--	----------------	--

	左侧的效果由右侧的代码生成.	<pre>\begin {tabu}{ [[5pt red] c c [5pt Indigo] } Hello & World \end {tabu}</pre>
--	----------------	---

没有必要使用花括号保护这些可选项: `[{...}]`. 这与 `tabu` 宏包的设计有关, 类似于 `siunitx` 宏包的 `S` 列模式.

2.2. 多样式的 `firsthline` 和 `lasthline`

```
\firsthline [extratabsurround] 制作多线条!
\firstline [extratabsurround]\hline
\lasthline [extratabsurround]
\lastline [extratabsurround]\hline
```

`\firsthline` 和 `\lasthline` 是在 `array.sty` 中定义的, 在使用横线时可以用来保护文本的对齐方式. 此外, 可选项 `[extratabsurround]` 可以用来改变 `\extratabsurround` 的尺寸.

<div>没有 使用 表格 横线 命令</div> <div>对比 使用 表格. 横线 命令</div>	<div>没有 使用 表格 横线 命令</div> <div>对比 使用 表格. 横线 命令</div>
使用了 <code>\firsthline</code> 和 <code>\lasthline</code> 命令	仅使用 <code>\hline</code> 命令 (文本的对齐方式效果不好)

现在我们使用 `\firsthline` 和 `\lasthline` 来制作双线, 三线 (或更多条线). 例如:

Top alignment	<div>没有</div> <div><code>\begin {tabu} [t] {c}</code></div> <div>使用<code>\\ 横线 \\ 命令\\</code></div> <div><code>\end {tabu}</code></div> <div>表格<code>\par</code> 对比</div> <div><code>\begin {tabu} [t] { c }</code></div> <div><code>\firsthline \hline \hline \hline</code></div> <div>使用 <code>\\ 横线 \\ 命令\\</code></div> <div><code>\lasthline \hline \hline \hline</code></div> <div><code>\end {tabu}</code> 表格.</div>	<div>没有 使用 表格</div> <div>横线</div> <div>命令</div> <div>对比</div> <div>使用</div> <div>横线</div> <div>命令</div> <div>表格.</div>
Bottom alignment	<div>没有</div> <div><code>\begin {tabu} [t] {c}</code></div> <div>使用<code>\\ 横线 \\ 命令\\</code></div> <div><code>\end {tabu}</code></div> <div>表格<code>\par</code> 对比</div> <div><code>\begin {tabu} [b] { c }</code></div> <div><code>\firsthline \hline \hline \hline</code></div> <div>使用 <code>\\ 横线 \\ 命令 \\</code></div> <div><code>\lasthline \hline \hline \hline</code></div> <div><code>\end {tabu}</code>表格.</div>	<div>没有 使用 表格</div> <div>横线</div> <div>命令</div> <div>对比</div> <div>使用</div> <div>横线</div> <div>命令</div> <div>表格.</div>

`\firsthline \firsthline \firsthline` 等价于: `\firsthline \hline \hline`
也相当于: `\firstline \hline \hline \hline`

`\lastline` 与其相同.

在这里 你可以看到这个黄色的`\extratabsurround` 支柱, 因为使用了命令`\tracingtabu=3`.

`\firsthline`, `\lasthline`, `\extratabsurround` 都是 `array` 中的命令. 使用 `\firsthline` 与 `\lasthline` 之后, 表格的上部和下部会多出一些空白, 因为使用了多线条, 如果想调节此空白的大小就需要使用 `\extratabsurround`. `array` 中是使用 `setlength` 赋值, 而在 `tabu` 中它作为了 `\firsthline`, `\lasthline` 后的可选项, 可以直接使用 `\firsthline[2pt]`.

2.3. 线条的更多设置

`\taburulecolor{<rule color>}`

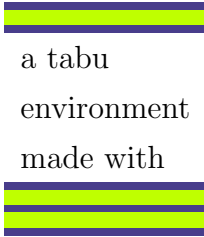
`\taburulecolor|<double rule sep color>|{<rule color>}`

`\taburulecolor` 是用来设置 `\hline`, `\firsthline` 和 `\lastline` 的颜色, 可以局部也可以全局设置. 使用 `\tabulinesstyle` 或 `\tabureset` 设置表格线格式时, `\taburulecolor` 也可以设置竖线的颜色.

上面代码中通过竖线设置的参数 `|<double rule sep color>|` 是设置两条相邻直线之间的

颜色. 如果没有指定此参数, 那么两条直线 (或者三条, 更多条) 之间通过空白隔开, 隔开距离为 `\vskip`.

```
\taburulecolor|lime|{DarkSlateBlue}
\arrayrulewidth=1mm \doublerulesep=2mm
Here is
\begin{tabu}spread 0pt[t]{X[-1]}\firsthline\hline
a tabu \\
environment \\
made with \\ \lasthline[5mm]\hline\hline
\end{tabu} tabu package !\par
And the next paragraph follows...
```



Here is a tabu tabu package!
environment
made with
And the next paragraph follows...

```
\tabulinestyle{<line style specification>}
```

`\tabulinestyle` 设置竖线 (`|`) 和横线 (`\tabuline`) 的格式. 但是 `\hline`, `\firsthline` 等线条的格式不会改变.

使用 `\tabulinestyle` 设置线条格式需要使用如下语法:

```
3pt 线条颜色 on 4pt 虚线颜色 off 5pt 两条虚线之间的颜色
线条颜色 on 4pt 虚线颜色 off 5pt 两条虚线之间的颜色
on 4pt 虚线颜色 off 6pt 两条虚线之间的颜色
3pt 线条颜色
on 4pt 虚线颜色
off 5pt
3pt
```

建立新线条格式名称使用 `\newtabulinestyle`

上面的语法格式表明, 任何参数都是可选的. 线条颜色 与 虚线颜色 相同, 如果两者都存在的话, 前者将会覆盖后者.

颜色的名称可以含有空格, 但是需要注意:

- 设置线条格式时, 如果第一个字符不是一个字母, 那么它将会被作为一个尺寸: 线条的厚度. 否则, 就会使用默认厚度, 即 `\arrayrulewidth`.
- 你所定义的颜色名称不能包含如下类型的的字符:

on?

off?

其中 ? 是第 12 类字符 (见 The \TeX book 第七章), 与 ! 不同.

```
\newtabulinestyle{<style=line spec., style=line spec., ...>}
```

这个命令定义的格式名称可以用作 `\tabuline` (横线), `|` (竖线), `\tabulinestyle` (全局或局部格式) 的可选项. 格式名称和颜色名称通过 `babel` 保护.

```
\tabucline[style or spec.]{start-end}
```

`\tabucline` 是制作横线的一个多功能的控制命令:

- 通过 `\tabucline` 增加线条厚度之后不会影响竖线格式,
- `\tabucline` 能够很好的处理 `\extrarowheight`,
- 通过使用 `pgf/TikZ` 语法可以制作虚线:

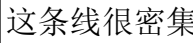
```
\tabucline[<width> on<dash> off<gap>]{<first column>-<last column>}
```

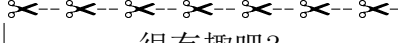
- 可以通过 `\hbox` 给 `\tabucline` 先定义一些命令作为引导符: `<spec.>` 前必须使用 `\hbox`, `\box` 或 `\copy`.
- 最后你还可以在 `\tabucline` 中使用颜色名称, 但是必须在放在最后面.

<code>\tabucline [1pt on 1.5pt off 2pt]{1-4}</code>	将直线设置为虚线, 厚度为 1pt. 虚线长 1.5pt 且虚线间隙为 2pt. 直线从第 1 列到第 4 列. 但此表格只有 2 列, 所以到第 2 列时停止.
<code>\tabucline [1.5pt]{-}</code>	划一条实直线, 厚度为 1.5pt, 从第一列到最后一列.
<code>\tabucline {2-}</code>	划一条厚为 <code>\arrayrulewidth</code> 的实直线 (即默认厚度), 从第 2 列到最后一列.
<code>\tabucline [on 2pt red]{-5}</code>	划一条虚线, 从第一列到第 5 列, 厚度为 <code>\arrayrulewidth</code> . 虚线长 2pt, 间隙长 4pt (默认值).

定义线条格式	使用定义的格式
<code>\newtabulinesstyle {myline=0.4pt on 2.5pt off 1pt red}</code>	<code>\tabucline [myline]{-}</code>
在 <code>\tabucline</code> 里直接使用特殊符号 或者在盒子 <code>\hbox</code> 里面使用引导符	
<code>\tabucline [\hbox {\\$ \scriptstyle \star \\$}]{1-3}</code> , 见顶线与底线效果	

虚线或点线
下面是默认效果

虚线	间距
	
这条线很密集	

虚线	间距...	...引导符
		
很有趣吧?		

2.4. 横线格式的重复和行的颜色设置

横线格式的重复

```
\everyrow{code}
```

如果你的表格从上到下的横线格式是相同的, 那么你就不需要每行都使用 `\tabucline` 或者其他命令设置了, `\everyrow` 可以自动设置多条横线的格式.

```

\begin{tabu}to .5\linewidth{cX[2mc]X} \tabucline[1pt]-
\everyrow{\tabucline[on 2pt]-}
This is      &a small example &of a \texttt{tabu}\\
which        &automatically   &inserts      \\
a horizontal &line after      &each of its row \everyrow{} \\
\tabucline[1pt]-
\end{tabu}

```

This is	a small example	of a tabu
which	automatically	inserts
a horizontal	line after	each of its row

上例中, 使用了 `\everyrow{\tabucline[on 2pt]-}` 之后, 下面的横线都自动的设置为它所设置的格式, 直到重新修改格式或者使用 `\everyrow{}` 清空为止.

`\everyrow{}` 也可以在 `longtabu` 中使用.

表格行颜色设置

```

\taburowcolors[first line]<number>\{first .. last}

```

`\taburowcolors` 用来设置表格每行的交替颜色. 此命令可以在 `tabu` 环境之前或者环境之内使用, 也可以用在最后一行.

可选参数 `[first name]` 是设置背景颜色开始的第一行 (不一定是表格的第一行)—当 `\taburowcolors` 用在表格的最后一行时, 此可选项是不起作用的: 因为只有在此命令开始使用的时候, 行的颜色才可以设置. `<number>` 是设置颜色的交替数. 没有指明的话, 默认就是 2, 也就是每隔一行换一次颜色. 如果设置为 3, 那就是每隔 2 行换一次颜色. `<first>` 和 `<last>` 是设置颜色系中的第一个和最后一个颜色. 具体见下面的例子.

```

\taburowcolors[2] 3{Crimson!30 .. ForestGreen!40}
\taburulecolor |GreenYellow|{OrangeRed}
\arrayrulewidth=1pt \doublerulesep=1.5pt
\everyrow{\hline\hline}
\begin{tabu}to 0.7\linewidth{X[-1]X}
This is      &just a test  \\
and i think &it will    \\
look        &rather bad  \\

```



```

for      &i   have not  \\
chosen   &the colors   \\
with care. &i can not   \\
say      &less...      \\
\taburowcolors 2{Crimson .. ForestGreen}
1        &This is Crimson    \\
2        &This is ForestGreen \\
3        &This is Crimson    \\
4        &This is ForestGreen \\
\end{tabu}

```

这个表格	只是一个简单的测试
------	-----------

而且我想	它看起来应该很难看
------	-----------

而且效果	应该很差
------	------

因为	我没有
----	-----

认真	选择颜色
----	------

额, 好吧.	你自己看着办吧
--------	---------

我就	不多说了...
----	---------

1	这个颜色是 Crimson
---	---------------

2	这个颜色是 ForestGreen
---	-------------------

格式复原

```
\tabureset
```

如果想回到 tabu 环境的标准格式, tabu 提供了一个 `\tabureset` 命令, 它可以做到如下格式设置:

<code>\tabulinesep = 0pt</code>	<code>\extrarowsep = 0pt</code>	<code>\extratabsurround = 0pt</code>
<code>\tabulinestyle {}</code>	<code>\everyrow {}</code>	<code>\taburulecolor {}</code>
<code>\taburowcolors {}</code>		

3. 修改字体和每行的对齐方式

`\rowfont[alignment]{font specification}`

在 `tabu` 环境中, 你可以通过 `\rowfont` 修改每行中单元格的字体. 而且 `\rowfont` 优先于对列字体的设置, 就好像在 `colortbl` 宏包中, `\rowcolor` 优先于 `\columncolor` 设置颜色一样.

每一行中单元格的对齐方式可以修改为:

<code>l = left</code>	或者 <code>ragged2e</code> 中的设置:	<code>L</code>
<code>c = center</code>		<code>C</code>
<code>r = right</code>		<code>R</code>
<code>j = justify</code>		<code>J</code>

在选项 `<alignment>` 中的其他设置值将会被忽略. 如果没有加载 `ragged2e` 宏包, 大写的 `L` `R` `C` 和 `J` 将会被视作为小写, 设置效果与小写形式相同.

```
\begin{tabu}{|X|X[-1]|}\tabucline-
\rowfont[c]\bfseries
  This          &Is          &\\ \tabucline[on 2pt,blue]-
  tabu          &&package      &\\ \tabucline[off 2pt blue]-
  \rowfont[r]\itshape
  for           &&\texttt{tabu}  &and
                &&\texttt{longtabu} \\ \tabucline-
\end{tabu}
```

	This	Is
tabu		package
	for	tabu and longtabu

4. 保存和使用 tabu 数据

`\savetabu{<user-name>}`

`\savetabu` 命令可以放在 `tabu` 环境中任意一行的末端, 用来保存 `tabu` 环境中的参数. 这种保存方式是全局的. `\savetabu` 也可以用来制作 `tabbing` 环境, 此环境记住 `\savetabu` 的位置.

如果 `<user-name>` 已经被使用过, 先前的设置将会被改写并且代写信息会被写入 `.log` 文件中. 当 `\tracingtabu>0` 时, `\savetabu` 保存的数据也会被写入 `.log` 文件中.

使用保存好的数据可以通过 `\usetabu` (全部数据将被调用), 或者 `\preamble` (只调用表格导言区的设置, 即 `\begin{tabu}to[pos]{tabular preamble}` 中的 `{tabular preamble}`).

`\usetabu{<user-name>}`

`\usetabu` 与 `\savetabu` 相对应: 它可以单独方式在表格环境开始的地方, 代替我们对列设置的一般方法 (即, 替代 `to\linewidth{X}` 等), 而使用我们之前通过 `\savetabu` 保存好的设置数据. 当然 `<user-name>` 必须是存在的, 否则你会得到一个错误信息.

`\usetabu` 有助于制作格式完全相同的表格. 唯一一个可以修改的参数是整个表格垂直方向与文本段落对齐方式的可选项. 注意, `\usetabu` 不适用于 `longtabu`.

`\usetabu` 可以调用:

- 表格导言区的设置.
- 表格垂直方向与文本段落的对齐方式 `[c]`, `[b]`, `[t]`, 除非这个位置被重新指定.
- `tabu` 的目标宽度: 被保存的目标宽度不包含任何控制序列, 它是按尺寸大小保存的, 单位是 `point`.
- `tabu` `X` 列模式中设置的宽度: 这些宽度将不会被计算第二次—即使是负参数—`X` 列被直接替换为相同宽度的 `p`, `m`, `b` 模式, 这些宽度都是经过 `\savetabu` 直接保存的.
- `\tabcolsep` (或者数学模式中的 `\arraycolsep`), `\arraycolsep`, `\extrarowdepth`, `\arraystretch` 以及 `\extratabsurround`.
- `\arrayrulewidth`, `\doublerulesep` 和 `\everyrow`, `\taburulecolor`, `\tabulinestyle`, `\taburowcolors` 四条命令设置的参数.
- `colortbl` 宏包中的 `\minrowclearance`.

`\abovetabulinesep` 和 `\belowtabulinesep` 并没有被保存, 因为它们与表格内容的关联比表格形状关联更大.

看下面的例子:

```
\tabcolsep=12pt \extrarowsep=1mm
\tabulinestyle{on 1pt ForestGreen}
\begin{tabu}to .7\linewidth{|XXX|X[c]|}
```

```
\savetabu{mytabu} \tabucline-
This & is & tabu & package \\ \tabucline-
\end{tabu}
```

This	is	tabu	package
------	----	------	---------

```
\tabureset
\begin{tabu}{\usetabu{mytabu}} \tabucline-
\multicolumn{3}{c}{This is tabu} & package \\ \tabucline-
\tabuphantomline
\end{tabu}
```

This is tabu			package
--------------	--	--	---------

使用 `\savetabu` 和 `\usetabu` 在同一个文档中可以很轻易地制作相同宽度的表格. 目前这两个命令还不支持修改表格的目标宽度, 新增一列的功能.

```
\preamble{<user-name>}
```

`\preamble` 也可以调用 `\savetabu` 保存的数据. 这是 `\usetabu` 命令的变异命令, 它可以调用的数据有:

- `tabu` (或 `longtabu`) 的导言区.
- 垂直对齐方式 `[c]`, `[b]`, `[t]` (或 `longtabu` 中的 `[c]`, `[l]`, `[r]`), 除非此可选项被重新指定.
- `tabu/longtabu` 的目标宽度, 除非此宽度被指定.

其他表格参数不会被调用. 将 `\preamble{<user-name>}` 单独放置在 `tabu` (或 `longtabu`) 的导言区以替换一般的列设置命令. `\preamble` 可以在 `longtabu` 中使用. 见下面的例子 (续上):

```
\tabulinestyle{1pt off1pt}
\begin{tabu} to\linewidth{\preamble{mytabu}} \tabucline-
This & is & \texttt{tabu} & package \\ \tabucline-
\end{tabu}
```

This	is	tabu	package
------	----	------	---------

`\tabcolsep`, 直线的颜色等, 并未从 `\savetabu` 调用: 只有 `tabu` 导言区的数据被调用.

5. 其他功能

5.1. 使用抄录环境

在 tabu 中使用抄录, 首先最好使用 fancyvrb 宏包, 然后将 tabu 环境改为 tabu* 即可, longtabu 环境也是如此, 改为 longtabu*. 见下面的例子.

```
\begin{tabu*}{|X|X|} \tabucline-
This is a small \Verb+\Verbatim+\par
insertion
&
\begin{Verbatim}
And this is a complete
Verbatim environment
\end{Verbatim}
\\ \tabucline-
\end{tabu*}
```

This is a small \Verbatim insertion	And this is a complete Verbatim environment
--	--

5.2. 制定新环境

使用 tabu 可以很容易指定一个符合自己习惯的表格环境, 例如:

```
\newenvironment{foo}
{\CTEXnoindent
\begin{tabu}{|X[1.2]| [1pt gray]X|}}
{\end{tabu}}
```

```
\begin{foo}
```

使用这种方法可以使你的代码变得简洁

&

这是~\texttt{tabu}~比其他表格宏包更容易操作的原因

```
\end{foo}
```

使用这种方法可以使你的代码变得简洁	这是 tabu 比其他表格宏包更容易操作的原因
-------------------	-------------------------

5.3. 对齐数字

实际上 `tabu` 也有对齐数字的功能, 例如使一系列数字在小数点处对齐, 等等, 但是需要额外的 `siunitx` 或 `numprint` 宏包等. 其实 `tabu` 就是使用这两个宏包的代码, 设置几乎完全相同. 个人认为还是比较复杂的, 故此处就不做更多讲解, 新手可以自己查看英文文档.

6. 宏包选项

6.1. debugshow 选项

```
\tracingtabu
\tracingtabu=1, 2, 3 或 4
```

`\tracingtabu` 作为 `debugshow` 的选项, 有如下效果:

- 在使用 X 列模式时, `tabu` 会报告计算出来的目标宽度.
- 当使用 `\savetabu` 时保存的 `tabu` 信息会被写入 `.log` 文件.

`\tracingtabu =2` 给出更多测量自然宽度的信息.

`\tracingtabu =3` 在 `tabu` 环境中显示出插入表格的支柱, 给出更多测量每行高度和深度的信息.

`\tracingtabu =4` 显示通过 `\tabucline` 插入的信息.

此说明文档为 `tabu` 宏包在导言区设置参数的即为 `\tracingtabu=2`.

6.2. delarray 选项

`delarray` 选项只会影响加载的 `delarray.sty` 宏包, 可以使 `tabu` 在文本模式和数学模式下都可以使用 `delarray.sty` 宏包定义的符号命令.

6.3. linegoal 选项

使用 `linegoal` 选项时, `tabu` 的 X 列模式的默认目标宽度由 `\linegoal` 取代 `\linewidth`. 当然必须加载 `linegoal` 宏包, 而且必须使用 `pdf \TeX` 编译. 或者你可以自己使用如下命令将此选项改写: `\begin{tabu}to\linewidth`.

7. 总结

虽然 `tabu` 可以制作多种格式的表格, 但是也有一些缺点. `tabu` 并没有单独设置纵向合并单元格的命令, 也没有单独设置画斜线表头的命令. 而且需要更多宏包的支持. 没有哪一个宏包能够做到完美, 这一点我们应该能够认识到.

鉴于笔者时间有限, 目前此文档只能做到这个地步, 有何错误或遗漏还请各位高手指点.