



快速 Word 公式排版指南

Word 上排版数学公式,用键盘就好!

作者:宋宁

单位:山东理工大学

E-mail: songning@sdut.edu.cn

时间: June 8, 2019



我们必须知道,我们必将知道 —— 大卫·希尔伯特

目 录

1	简介	2
1.1	手不离键盘的操作	2
1.2	Unicode-Math 和 \LaTeX	3
2	编码简单的数学公式	6
2.1	分式和除法	6
2.2	乘法和上下标	7
2.3	希腊字母	8
2.4	开根号	9
2.5	一些常用数学符号	10
2.6	常见数学函数	11
2.7	空心体、花体和手写体英文字母	11
3	带帽子、穿鞋和穿衣服的数学公式	12
3.1	穿衣服的数学公式	12
3.2	向量和圆弧	12
3.3	其他常见的戴帽子的公式	12
3.4	头上戴括号以及脚上穿括号的数学公式	13
3.5	极限、积分以及连续求和、求积、求并和求交	14
4	编码大型的数学公式	15
4.1	矩阵类公式	15
4.2	方程组和分段函数	16

敬告读者

-
1. 本指南的箴言是五个字：**手不离键盘**；
 2. 本指南不是用来阅读的，而是用来操作的；
 3. 如果你觉得本指南很傻，请参考第二条；
 4. 如果你所用 Word 版本低于 2007 或者使用了 WPS，请出门左拐不送谢谢；
 5. 如果你在实践本指南的时候，总是管不住自己总是想去摸鼠标，请默念第一条的五个字箴言，直到不想摸鼠标或者放弃练习；
 6. 如果你学习了本指南之后发现，公式录入速度并没有提高，请练习打字后重新学习本指南；
 7. 目前本指南可能有错，如果有一天你发现本指南的最新版本没有了本条，那就是真没有了本条；
 8. 本人写了本指南，并不等于本人喜欢 Word……

第 1 章 简介

1.1 手不离键盘的操作

常见的计算机排版模式有两种。一种是类似 Word 软件这样的，你在输入的同时就能看到排版效果，这种模式称为“所见即所得”。它的好处在于简单、直观、易学，但是缺点也很明显，其排版结果既不精准也不足够美观，尤其是在文档中有大量输血公式的情况下。另一种模式，称为“所想即所得”，比如科研著作常用的 \LaTeX 、网页文件常用的标记语言 HTML、样式表 CSS 等。它们的特点是，输入文档时，实际输入的是代码，然后通过编译性或解释性的程序，使之呈现出来。它的特点是精准、美观，尤其是 \LaTeX ，他天生就是为数学排版准备的，其排版水准远远高于 Word 之类的“所见即所得”模式。缺点是，看上去一堆各种代码，让很多人觉得望而却步，尤其是 \LaTeX 需要编译，更让很多人觉得难。

其实，两种排版模式之间还有一种不易察觉的对比，那就是：“所见即所得”模式的工作效率极低，而 \LaTeX 这种“所想及所得”的工作效率非常之高。原因在于， \LaTeX 的主要工作就是输入代码，其输入设备是键盘，这种操作俗称“手不离键盘”操作，只要你打字足够准确和快速，其工作效率就会非常高。笔者曾经见过一个熟手，用了几个小时把一份十几页的数学文献排版完成。这对于 Word 这种需要频繁地交替使用键盘鼠标的软件来说几乎是不可能的。

为了平衡效率和易用性这对矛盾，从 2007 版 Word 开始，微软公司抛弃了原来的公式编辑器（即公式编辑器 3.0），事实上也就抛弃 MathType 等一干赚差价的插件，推出了一款新的公式编辑器。它在一定程度上能够通过输入代码来实现一些 latex 的功能，而且用它所编辑的公式在颜值上也大为提高。随后，微软逐步改进了这个自带公式编辑器（同时也保留了老版本的公式编辑器 3.0）。在目前比较新的 Word 2016 中，这个自带的公式编辑器已经几乎完全支持了 Unicode-Math 代码，部分地支持了 \LaTeX 代码，再配合上各种快捷键，已经基本上能实现“手不离键盘”的操作了。

其中， \LaTeX 代码由于其在科研领域的优势地位，所以 Word 2019 给予其一定的支持，但是 \LaTeX 有两个问题：一是不易学，它与很多人手写公式的习惯差距较大；二是代码效率比较低，要输入一个公式往往要很大的代码。而 Unicode-Math 代码在一定程度上能解决这个问题。它是目前最接近人类手写习惯的数学排版代码，而且代码效率要比 \LaTeX 高很多。

所以，我们这本小册子主要介绍两个东西，一是 Word 上与数学公式有关的快捷键，二是 Unicode-Math 代码的使用。快捷键只要记住六个就基本够用了，在英文状态下，试试下面这六个重要的快捷键吧（注：<Alt>，<Ctrl>，<Shift>，<Enter>、<Space> 和 <↓> 分别表示：Alt 键、Ctrl 键、Shift 键、回车键、空格键和向下的光标键，为了与文本相区分，所有按键将以 <> 包裹，例如等号键和斜杠键本指南分别记作 <=>，</>）：

- <Alt> + <=> 插入公式（线性模式）；

- <Alt> + <↓> 打开光标所在公式的下拉菜单;
- <Ctrl> + <i> 开始/结束斜体模式;
- <Ctrl> + 开始/结束粗体模式;
- <Space> 将此公式编辑框中的代码变成公式;
- <Shift> + <Enter> 软回车,即换行不换段,通常在显示型公式后使用。

本指南针对 Word 2016。对于其他版本的 Word,如果是 2007 版以及更高版本,笔者介绍的内容大部分仍然可用。至于你非要用 WPS……请出门左拐不送。

1.2 Unicode-Math 和 L^AT_EX

那么什么是 Unicode-Math 呢?它是 Unicode 的一个子集。Unicode 译为“统一码”是目前计算机上非常流行的一种字符编码方式,它之所以被称为统一码,是因为它已经囊括了世界上几乎所有文字符号。而数学相关符号正是其一部分,称为 Unicode-Math 编码。它比常见的 L^AT_EX 以及 MathML 更加紧凑和易读。我们以后简称其为 Uni-Math 代码。

而 L^AT_EX 是一套令人极其舒适的排版语言,主要用于科研著作的排版。Word 2016 兼顾了这二者,一定程度上实现了“手不离键盘的操作”。实际上,Uni-Math 代码要比 L^AT_EX 简洁,二者有相当多的代码是类似的,如果你会用 L^AT_EX,那么也会很快学会 Uni-Math 代码的。

下面通过一个简单的例子展示一下 Uni-Math 代码的使用。

使用 Word 2016 新建一个文档。我们希望输入如下文本:

牛顿发现的万有引力公式

$$F = \frac{GMm}{r^2}$$

开启了天体物理学的新篇章。

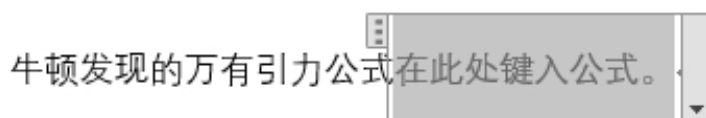


图 1.1

可以按如下步骤进行:

1. 输入文本:“牛顿发现的万有引力公式”。
2. 使用快捷键 <Alt> + <=>:进入数学公式输入状态。
3. 使用快捷键 <Ctrl> + <i>,进入斜体输入状态,如图 1.1所示。

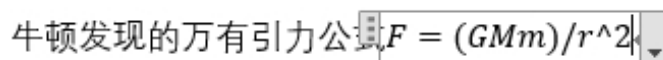


图 1.2

4. 输入代码： $F=(GMm)/r^2$ ，如图 1.2 所示。
5. 使用 <Space>，此时公式中最后的 r^2 变为 r^2 ，如图 1.3 所示，<Space> 的功能其实是将光标前面且紧邻光标的代码变成数学公式。

牛顿发现的万有引力公式 $F = (GMm)/r^2$

图 1.3

6. 再次使用 <Space>，此时公式出现分式，如图 1.4 所示。

牛顿发现的万有引力公式 $F = \frac{GMm}{r^2}$

图 1.4

7. 此时我们已经几乎完成这个公式了，只不过现在这个公式是线性型公式，所谓“线性型公式”就是与前后文本同行显示的公式，是最常见的公式排版形式。我们希望将这个公式单独居中显示（即显示型公式）。所谓“显示型公式”是指公式独占一行居中显示且不换段，显示型公式往往便于展示一些大型数学符号，如求和号、求积号和各种积分号，如果有重要结果需要展示也需要用显示型公式等。回到 word 中，此时光标仍然在公式编辑框中，使用快捷键 <Alt> + <↓>，打开公式编辑框的下拉菜单，如图 1.5 所示。

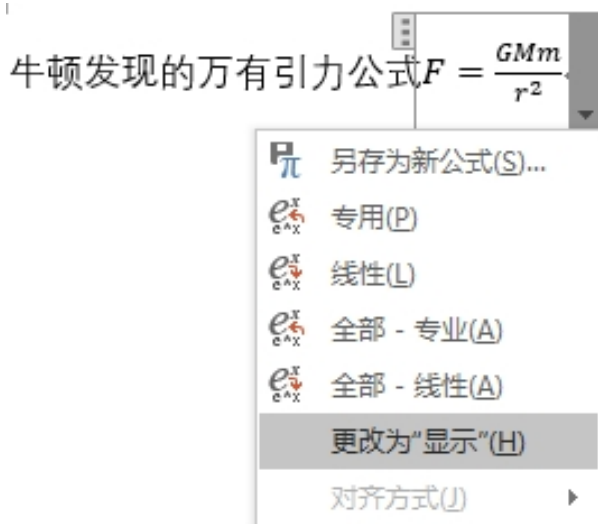


图 1.5

8. 移动光标，或者直接按字母键 <h>，来选择更改为‘显示’，此时该公式变为显示型，如图 1.6 所示。
9. 移动光标，离开公式编辑框，使用快捷键 <Shift> + <Enter>，实现软回车后。注意，如果显示型公式前后文本其实是一个自然段，那么应该使用软回车，它实现了换行不换段，这样前后文本在同一段中，能共用一套段落设置，行文也更加合理。接下来输入下一行文本：“开启了天体物理学的新篇章。”，如图 1.7 所示，任务完成。

牛顿发现的万有引力公式↓

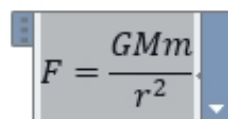


图 1.6

牛顿发现的万有引力公式↓

$$F = \frac{GMm}{r^2} \downarrow$$

开启了天体物理学的新篇章。|↵

图 1.7

这里，我们在公式编辑框中所输入的代码`F=(GMm)/r^2`就是一串 Uni-Math 代码；后来两次使用 `<Space>`，第一次是将代码`r^2` 变成 r^2 ，其中代码`r^n` 表示 r^n ，第二次是改变分式`x/y`，这个代码表示 $\frac{x}{y}$ ，其中代码`GMm` 上的小括号表示这是一个整体，要全部放在分子上，其实这个小括号是可以省略的。



注意：

- 代码中的所有功能性的符号都必须是英文符号。
- 代码输入后必须使用空格键，才能将代码变成数学公式。

后面几章我们详细学习 Uni-Math 代码的使用。

第 2 章 编码简单的数学公式

2.1 分式和除法

分式的代码的格式为(<Space> 表示空格键)

(分子)/(分母)<Space>

例如, $(x+y)/(u+w)$ <Space>, 即

$$\frac{x+y}{u+w}$$

如果分子或分母是连续的字母和数字, 就可以省略相应的括号, 例如, $2xy/5ab$ 得到的是

$$\frac{2xy}{5ab}$$

如果你确实要保留斜杠等号, 那么应该使用

(分子)\/(分母)<Space>

如果你需要使用除号 \div , 那么你需要使用代码\div, 一般格式为:

(被除数)\div<Space>(除数)

例如,

$a\div b$

得到

$$a \div b$$

$(a+c)\div b$

得到

$$(a+c) \div b$$

请注意 <Space> 出现的位置。

如果确实需要保留括号,例如



注意: 1. 如下数学公式

$$\frac{1}{3.1415926}$$

应该输入代码

$$1/(3.1415)$$

而不是

$$1/3.1415$$

请注意括号的使用规则。

2. 符号/之后接某些代码可能不会分式,例如/= 其实是 \neq ,这个我们以后再讨论。

有可能你想造出一个没有分数线的分式,代码如下:

$$(\text{分子})\backslash\text{atop}\langle\text{Space}\rangle(\text{分母})\langle\text{Space}\rangle$$

例如, $(n+m)\backslash\text{atop}\langle\text{Space}\rangle n\langle\text{Space}\rangle$, 其对应的数学公式如下:

$$\frac{n+m}{n}$$

可以借助这个代码写出二项式系数的代码,例如 $((n+m)\backslash\text{atop}\langle\text{Space}\rangle n)\langle\text{Space}\rangle$, 它对应的数学公式是

$$\binom{n+m}{n}$$

2.2 乘法和上下标

如果你的乘法省略了乘号,那么直接输入就可以了。否则就看你使用什么样的乘号下面是常见的乘号的代码使用范例:

- $a \times b$ `a\times\langle\text{Space}\rangle b`
- $a \cdot b$ `a\cdot\langle\text{Space}\rangle b`
- $a \circ b$ `a\circ\langle\text{Space}\rangle b`
- $a \otimes b$ `a\otimes\langle\text{Space}\rangle b`
- $a \odot b$ `a\odot\langle\text{Space}\rangle b`

上下标分别使用符号 \wedge 和 $_$,即 `<Shift> + <6>` 和 `<Shift> + <->`。范例如下:

- x^2 `x^2`
- y_i `y_i`

- a_{11} `a_11`
- b_{i+j} `b_(j+j)`

请注意括号的使用。



注意: 代码

$$a^{i_j}$$

对应的公式为

$$a_j^i$$

但是代码

$$a^{(i_j)}$$

对应的公式为

$$a^{i_j}$$

上面说的是上下标在右侧的情况,这是比较常见的。也有可能在数学公式中出现上下标在左侧的情况。例如 ${}_j^i F$, 这应该使用如下代码

$$(_j^i)F<Space>$$

注意这种情况的空格键 `<Space>` 放在最后。

还有一种上下标是放在上方和下方,例如极限符号,我们将在后面讨论。

2.3 希腊字母

希腊字母按其英文转写形式给出,如果能准确读出希腊字母,几乎就能正确写出其代码。下面给出小写希腊字母的代码:

- α `\alpha`
- β `\beta`
- γ `\gamma`
- δ `\delta`
- ϵ `\epsilon`; 变体 ε `\varepsilon`
- ζ `\zeta`
- η `\eta`
- θ `\theta`
- ι `\iota`
- κ `\kappa`
- λ `\lambda`
- μ `\mu`
- ν `\nu`
- ξ `\xi`

- 第 15 个希腊字母读作 omicron, “欧米克荣”, 字形上与英文字母 o 完全一样, 一般直接用英文字母 o
- π \pi 变体 ϖ \varpi
- ρ \rho 变体 ϱ \varrho
- σ \sigma
- τ \tau
- υ \upsilon
- ϕ \phi; 变体 φ \varphi
- χ \chi
- ψ \psi
- ω \omega

关于大写希腊字母, 其中有很多与英文字母相同, 所以直接使用英文字母, 这些大写希腊字母如下。

小写	α	β	ε	ζ	η	ι	κ	μ	ν	o	ρ	τ	υ	χ
大写	A	B	E	Z	H	I	K	M	N	O	P	T	Y	X

其他大写希腊字母的代码是直接将小写字母的代码的第一个字母大写, 其代码如下

小写	γ	δ	θ	λ	ξ
大写	Γ	Δ	Θ	Λ	Ξ
代码	\Gamma	\Delta	\Theta	\Lambda	\Xi

小写	π	σ	φ	ψ	ω
大写	Π	Σ	Φ	Ψ	Ω
代码	\Pi	\Sigma	\Phi	\Psi	\Omega



注意: 1. 在 Word 公式中, Γ 的代码 \Gamma 可以简写作 \G。

2. 空集不是 ϕ , 也不是 Φ 。空集实际上是一个打斜杠的零, 是 \emptyset , 其代码为 \emptyset。

2.4 开根号

开平方的符号为 $\sqrt{\quad}$, 代码为 \sqrt。开平方的一般格式为:

\sqrt(被开方数)<Space>

开 n 次方的一般格式为:




`\sqrt{n\&(被开方数))<Space>`

当然内层括号可以视情况而省略。
另外,开 n 次方还有一种写法:

`\root<Space>n\of(被开方数))<Space>`

2.5 一些常用数学符号

符号	代码	符号	代码
\pm	<code>+-</code>	\mp	<code>-+</code>
\geq	<code>>=</code>	\leq	<code><=</code>
\cong	<code>~=</code>	\equiv	<code>\equiv</code>
\approx	<code>\approx</code>		
\cap	<code>\cap</code>	\cup	<code>\cup</code>
\in	<code>\in</code>	\ni	<code>\ni</code>
\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>
\because	<code>\because</code>	\therefore	<code>\therefore</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>
∞	<code>\infty</code>		
\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Leftarrow	<code>\Leftarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>		
\parallel	<code>\parallel</code>	\bot	<code>\bot</code>
∂	<code>\partial</code>	\square	<code>\square</code>
$^\circ$	<code>\degree</code>	$^\circ C$	<code>\degC</code>

-  **注意:** 1. 上述大部分符号的代码同时也是 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码。但是请注意,上述有六个不带反斜杠的代码,它们不是 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码,它们要更简洁。`\box`、`\degree` 和 `\degC` 也不是。
2. 属于是 \in , 代码为 `\in`, 不是 ϵ , 这是一个希腊字母。
3. 关系类符号的否定形式, 例如 \neq, \notin 等, 是在对应代码前加反斜杠 `/`。例如 \neq 的代码是 `\neq`, \notin 的代码是 `\notin`, \nless 的代码是 `\nless`



2.6 常见数学函数

以正弦函数为例,其代码的一般格式为

```
sin\of<Space>(自变量)<Space>
```

其中自变量外的括号可以根据情况省略。例如,数学公式

$$y = \sin A(\phi t + \omega)$$

的代码为

```
y=sin\of<Space>A(\phi<Space>t+\omega<Space>)<Space>
```

类似地,上式中的 `sin` 也可以换成 `cos`, `tan`, `log`, 你可以根据需要来。

2.7 空心体、花体和手写体英文字母

空心体、花体和手写体字母的代码其实就是在相应字母前加上前缀。

- 空心体前缀 `double`(因为相当于写了两遍),如 \mathbb{R} ,代码为`\doubleR`
- 花体前缀 `franktu`(可能是德语“法兰克福”,德国城市,也许因为该字体来自于此),如 \mathfrak{M} ,代码为`\franktuM`
- 手写体前缀 `script`(字面意思),如 \mathcal{E} ,代码为`\scriptE`

第3章 带帽子、穿鞋和穿衣服的数学公式

有些数学公式是戴帽子的,例如

$$\widehat{ABC}, \overrightarrow{AB}, \dot{y}$$

类似地,还有的数学公式是穿鞋的,例如

$$\underbrace{1+1+\cdots+1}_{5\text{个}}$$

当然,还有穿衣服的,例如

$$\boxed{E=mc^2}$$

这一章我们讨论这种公式。

3.1 穿衣服的数学公式

给公式 $E=mc^2$ 穿上衣服就是 $\boxed{E=mc^2}$,其代码如下

```
\rect(E=mc^2)<Space>
```

 **注意:** 在单倍行距下,上述公式的外框会有部分看不见。建议段落行距设为 1.1 以上。

3.2 向量和圆弧

一个字母表示的向量,如 \vec{a} ,使用如下代码

```
a\vec<Space><Space>
```

两个或者两个以上字母表示的向量,如 \overrightarrow{AB} ,使用如下代码

```
(AB)\vec<Space><Space>
```

二者的区别在于括号的使用。

3.3 其他常见的戴帽子的公式

格式与向量类似,所用代码稍有区别,见下表

公式	代码	公式	代码
\hat{a}	<code>a\hat{<Space><Space>}</code>	\tilde{a}	<code>a\tilde{<Space><Space>}</code>
\dot{a}	<code>a\dot{<Space><Space>}</code>	\ddot{a}	<code>a\ddot{<Space><Space>}</code>
\ddot{a}	<code>a\dddot{<Space><Space>}</code>	\bar{a}	<code>a\bar{<Space><Space>}</code>

3.4 头上戴括号以及脚上穿括号的数学公式

头上戴小括号的公式很常见，也就是平面几何上的圆弧。圆弧 \widehat{ABC} （你可以试试 ABCDEFG，都没问题）使用如下代码

`\overparen{ABC}<Space>`

另外一类头上或脚上穿戴括号的数学公式，就是如下这种了

$$\underbrace{1+1+\cdots+1}_{5\text{个}}$$

它的代码这样写

`\underbrace{1+1+\dots+1}_5<Space>`

其中，`\dots` 是省略号。省略号有如下几种

省略号	代码
...(脚面上的省略号)	<code>\ldots</code>
...(挂腰带上的省略号)	<code>\cdots</code>
:(竖着的省略号)	<code>\vdots</code>
·.(对角的省略号)	<code>\ddots</code>

而`\dots` 兼顾了`\cdots` 和`\ldots` 两种，就是说适合哪个自动用哪个。

当然括号也可以穿在脚上，比如

$$\overbrace{1+1+\cdots+1}^{5\text{个}}$$

它的代码这样写

`\overbrace{1+1+\dots+1}^5<Space>`

注意，区别有两处。

在实际使用时，有可能出现斜体的情况，快捷键 `<Ctrl> + <i>` 不仅能进入斜体状态还能离开斜体状态，请灵活使用。

3.5 极限、积分以及连续求和、求积、求并和求交

前面讲下标的时候,说到有些公式的下标放在正下方更合理些,比如极限

$$\lim_{n \rightarrow \infty} a_n$$

它的代码是这样的

`\lim_(n->\infty<Space>)<Space>a_n<Space>`

无论它是线性型公式还是显示型公式,这个下标会自动调整到正下方。

而连续求和往往既需要正下方的下标,也需要正上方的上标,例如

$$\sum_{i=1}^n a_i$$

它的代码是这样的

`\sum_(i=1)^n<Space>a_n<Space>`

请注意,如果它是线性型公式,那么上述代码得到的是上下标在右侧的情况。要想将上下标调整为正上和正下,需要将线性型公式调整为显示型公式,即当光标在此公式编辑框中时,使用快捷键 `<Alt> + <↓>`, 打开下拉菜单,再使用字母键 `<h>`。

连续求积与之类似,例如

$$\prod_{i=1}^n a_i$$

它的代码是这样的

`\prod_(i=1)^n<Space>a_n<Space>`

求积分、连续求并和连续求交与上述二者相似只是符号有变,如下表所示

符号	代码	符号	代码
\int	<code>\int</code>	\iint	<code>\iint</code>
\oint	<code>\oint</code>	\oiint	<code>\oiint</code>
\cap	<code>\bigcap</code>	\cup	<code>\bigcup</code>

第4章 编码大型的数学公式

4.1 矩阵类公式

之所以叫“矩阵类”公式，我们是把矩阵、行列式等这一类形式上像表格一样的数学公式给一个统称。在排版时，它们的区别只在于外层使用什么括号。

首先看没有括号的矩阵类公式，例如

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

它的代码是

`\matrix(a&b@c&d)<Space>`

注意&(即 <Shift> +<7>)分离的是同一行的元素，而@ (即 <Shift> + <2>)则用来换行。

其他带括号的矩阵类公式，只需给上述代码带上相应的括号就可以了。使用圆括号给出的矩阵

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

它的代码是

`(\matrix(a&b@c&d))<Space>`

使用方括号的矩阵

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

它的代码是

`[\matrix(a&b@c&d)]<Space>`

行列式

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

它的代码是

`|\matrix(a&b@c&d)|<Space>`

使用花括号的矩阵

$$\left\{ \begin{array}{cc} a & b \\ c & d \end{array} \right\}$$

它的代码是

```
{\matrix(a&b@c&d)}<Space>
```

使用范数给出的行列式

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

它的代码是

```
\norm\matrix(a&b@c&d)\norm<Space><Space>
```

注意, `\norm` 给出范数符号, 但是需要两次空格 `<Space>`。

4.2 方程组和分段函数

如下方程组

$$\begin{cases} 10x + 3y = 2 \\ 3x + 13y = 4 \end{cases}$$

的代码为

```
{\eqarray(10x+3y&=2@3x+13y&=4)\close<Space><Space>
```

其中 `\close` 表示花括号结束。

注意, 如果需要调整方程的对齐情况, 可以在适当位置添加 `&`, 例如
分段函数与之类似, 如

$$f(x) = \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases}$$

其代码为

```
f(x)={\eqarray(x&\enspace<Space>x)>=0@-x&\enspace<Space>x<0)\close<Space><Space>
```

其中 `\enspace` 表示较大的空格。