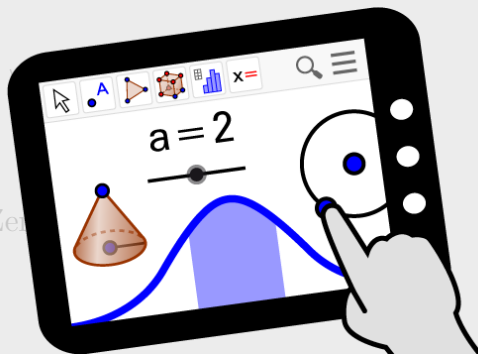


GeoGebra 中的 L^AT_EX 语法

L^AT_EX syntax in GeoGebra



```
\begin{equation*}
```

```
(a_{k1})=\left(
```

```
\begin{matrix}
```

```
0 \ldots 0 & 1 & 0 \ldots 0
```

```
& 0 \\
```

```
\BigZero & \cdots & \BigZero
```

```
& 0 \\
```

```
\end{matrix}
```

```
\right)
```

```
\quad
```

```
f(x)=
```

```
\begin{cases}(a_{k1}) =
```

```
-x^2,& \text{if } x < 0; \\
```

```
\alpha + x,& \text{if } 0 \leqslant x \leqslant 1; \\
```

```
x^2, & \text{otherwise.}
```

```
\end{cases}
```

```
\end{equation*}
```

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leqslant x \leqslant 1; \\ x^2, & \text{otherwise.} \end{cases}$$

GeoGebra 学习交流文档

作者：段明

版本：2.0.0

2020 年 8 月 · 昆明

第二版序

此文档第一次成稿于 2015 年，当时软件的最高版本还是 4.9，五年的时间过去了，学习和使用 GeoGebra 这一软件的人也成几何级数似的增长，着手对此文档进行修改的时候，软件已经进行了 600 多次更新，对于更新如此频繁的软件，可以说任何介绍软件的文档或书籍完稿时就已经落后了，但是一些基础的内容并不会过时，仍然值得有需求的人投入一些时间和精力进行学习。

GeoGebra 5 适合安装在台式电脑上使用，随着触摸设备的发展和普及，GeoGebra 也与时俱进的推出了适用于触摸设备的一个并行版本 GeoGebra 6，版本 5 和版本 6 除了界面和少部分细节有差异外，在使用上基本是相同的，对于一些操作上有差异的地方，文中也分别举例进行了说明。

在 GeoGebra 软件中是使用 JLaTeXMath 这一个优秀的开源 Java 库来处理 \LaTeX 代码的，JLaTeXMath 是显示 \LaTeX 代码的最佳 Java 库，许多重要项目都使用了此库，如 GeoGebra、Scilab、Freeplane、Mathpiper、CaRMetal、Ultrastudio 等。文档中的大部分内容也适用类似于 XMind 这一类使用 MathJax 开源库的软件或网站。

由于作者的水平有限，文中很多地方都写得不尽如人意，错误之处也再所难免，发现错误或不足的地方请与作者联系，邮箱：surfan@126.com，QQ：76482458，谢谢。

此文档为免费文档，任何人可以免费获得并分发此文档的完整版，未经作者授权，不得以任何形式对文档收费或用于商业用途，

段明

2020 年 8 月于昆明

第一版序

发现大家对 GeoGebra^① 这一软件的使用交流中，很多人对此软件所支持的 \LaTeX 这一语言产生了兴趣，而 \LaTeX 本身的功能过于庞大，学习时间较长，很多人需要的仅是其中输入数学公式这一部分，于是便萌生了写这个文档的念头。

此文档不是 GeoGebra 的入门教程，也不是 \LaTeX 的入门教程，其目的并不是让大家学会并掌握 \LaTeX 这一庞然大物，而仅仅只是针对在 GeoGebra 这一软件中数学公式的一些基础的、常用的输入方法进行介绍，希望能对大家的学习和工作起到微弱的作用并达到抛砖引玉的目的，以期待更多的高手能合力完善此文档。

此稿在初稿的基础上做了少量的修改，比如增加了字号设置，介绍了在“标题”选项中使用 \LaTeX 的方法，改进了示例的代码，修改了版面显示效果等内容。

在此感谢唐家军，唐大仕博士等老师提出的意见及指出文中错误。

段明^②

2015 年 12 月于昆明

① 官方网站：<http://www.geogebra.org/>

② 发现错误或者是需要改进的地方，请告诉我，邮箱：surfan@126.com，谢谢！

目 录

第二版序	II
第一版序	III
第一章 简单学点 L ^A T _E X	1
1.1 L ^A T _E X 是什么	1
1.2 L ^A T _E X 的命令	1
1.2.1 命令区分大小写	1
1.2.2 命令的写法	1
1.2.3 特殊符号的输入	2
1.2.4 环境	2
1.3 几个常用的命令	2
1.3.1 换行命令	2
1.3.2 撇号的输入	2
1.3.3 带圈字符的输入	3
1.3.4 给公式加个框	3
1.4 空白间距	4
1.4.1 水平间距	4
1.4.2 长度可控的水平间距	4
1.4.3 垂直间距	5
1.5 字号的设置	5
1.5.1 相对字号命令	5
1.5.2 相对字号修改示例	6

1.5.3	文本缩放命令	6
1.5.4	让文本跟随对象一起缩放	7
1.6	怎样使用 \LaTeX	7
1.6.1	在“标题”中使用 \LaTeX	8
1.6.2	在“文本”工具中使用 \LaTeX	9
1.7	文字颜色和背景色	11
1.7.1	文字颜色	11
1.7.2	背景色	12
1.7.3	设置标签的颜色	12
1.8	数学字体	13
1.9	在其它软件中使用 \LaTeX 公式	14
1.9.1	右键框选公式	14
1.9.2	导出图片	14
1.9.3	在其它软件中使用	15
1.10	行内公式和行间公式	15
1.10.1	行内公式	15
1.10.2	行间公式	16
第二章	数学符号及公式	17
2.1	长度单位	17
2.2	基本元素	18
2.2.1	希腊字母	18
2.2.2	上标和下标	18
2.2.3	分数和根号	20
2.2.4	二项式结构	20
2.2.5	省略号	21
2.3	运算符	21
2.3.1	常用二元运算符	21
2.3.2	常用关系符	22

2.3.3	大尺寸运算符	22
2.3.4	将下标分行	24
2.4	箭头	25
2.5	分隔符	25
2.5.1	大尺寸括号类分隔符	25
2.5.2	集合公式中的分隔符	26
2.6	多行公式	27
2.6.1	长公式	27
2.6.2	公式组	28
2.6.3	分段函数	28
2.6.4	方程组	29
2.7	矩阵	30
2.7.1	无分隔符的矩阵	30
2.7.2	带分隔符的矩阵	30
2.7.3	较复杂的矩阵示例	31
2.8	注音和标注	34
2.8.1	上划线和下划线	34
2.8.2	弧的符号	35
2.8.3	符号上下方的标注	35
2.8.4	向量的输入	36
2.8.5	带上下括号的公式	36
2.9	常用符号	36
2.9.1	几何符号	36
2.9.2	特殊符号	37
2.9.3	特殊字母	38
2.10	标准数学函数	38
2.11	简单表格	38
2.12	虚位与重叠	39

2.12.1	虚位的应用	39
2.12.2	利用虚位生成空格	40
2.12.3	递等式的对齐	40
2.12.4	水平虚位和垂直虚位	41
2.12.5	文字重叠	42
2.12.6	交错括号的排版	42
第三章	简单化学方程式	43
3.1	无机物	43
3.1.1	整数下标、电荷数、氧化数	43
3.1.2	分数下标	44
3.1.3	几种括号	44
3.1.4	上标与下标：同位素、分子、离子	44
3.1.5	圆点：自由基、水合物	44
3.2	有机物	45
3.2.1	化学键的表示	45
3.2.2	斜体： <i>cis</i> & <i>trans</i>	46
3.2.3	chemfig 宏包示例	46
3.3	箭头	47
3.3.1	反应方向	47
3.3.2	添加反应条件	47
3.3.3	沉淀与气体	48
3.4	热化学方程式：能量变化、分数计量数	48
3.5	其他内容	49
3.5.1	在文字下方添加标注	49
3.5.2	多行反应方程式	49
附录 A	L^AT_EX 简介	50
A.1	T _E X 家族	50

A.1.1	\LaTeX 简介	50
A.1.2	引擎: \TeX	50
A.1.3	格式: \LaTeX	51
A.1.4	宏包	51
A.1.5	驱动	51
A.1.6	小结	52
A.2	优点和缺点	52
A.2.1	选择 \TeX 的理由	52
A.2.2	\TeX 的缺陷	53
A.2.3	不选择 \TeX 的理由	53
A.3	软件准备	53
第二版跋		55
第一版跋		56

第一章 简单学点 \LaTeX

1.1 \LaTeX 是什么

\LaTeX 是一种基于 \TeX ^① 的面向数学和其它科技文档的电子排版系统，在生成复杂的数学公式方面表现特别出色。

此文档并不介绍完整的 \LaTeX 使用方法，其目的只是让大家能快速的掌握在 GeoGebra 这个数学软件中使用 \LaTeX 来输入数学公式的方法。

1.2 \LaTeX 的命令

1.2.1 命令区分大小写

\LaTeX 命令是大小写敏感的，在书写时请注意严格区分大小写，这样规定的目的有利于源文件在不同平台之间的移植。

1.2.2 命令的写法

\LaTeX 语法的命令以反斜线 (backslash) `\` 开始，命令名只能由字母组成，命令名后的空格符、数字或任何非字母的字符都标志着该命令结束。

还有一种情况是，反斜线后面跟着一个特殊符号，如换行命令、公式中的空白间距命令、或者是下面将要讲到的一些特殊符号等。

例如：`\sqrt[]{ }` 是一条命令，其中花括号 `{ }` 标识了命令的作用范围，表示这是一个整体，方括号 `[]` 标识的部分是可选参数，`\sqrt[3]{x + 5}` 可以得到 $\sqrt[3]{x + 5}$ 这一根式。

① 更详细的介绍请参看附录，或者网络上搜索。

1.2.3 特殊符号的输入

文档中可以输入的文字符号大致可以分为：普通字符、控制符、特殊符号、预定义字符串、注音符号等。

普通字符可以直接输入，而有些字符（例如 $\$ \wedge \& _ \{ \} \sim$ 等）被用作特殊的控制符，输入这些特殊符号时需要在前面加一个反斜线 \backslash 转义符，而 \backslash 本身则要用 `\backslash` 命令来输入，因为 $\backslash\backslash$ 被用作换行命令。

特殊符号的输入

`\$ \% \^ \& _ \{ \} \sim \backslash`

1.2.4 环境

在 \LaTeX 中，由开始标志 `\begin{环境名}` 到结束标志 `\end{环境名}` 组成的一个代码块称为一个环境，环境名首尾相同，例如：写分段函数的 `cases` 环境、功能强大的 `array` 环境、表格环境 `tabular` 等，我们将在后面具体介绍。

很多复杂的内容都是使用环境来完成的，环境可以嵌套。

1.3 几个常用的命令

1.3.1 换行命令

在 \LaTeX 代码中不会根据回车符（Enter）进行换行，如果要想文字或公式根据需要进行换行则需使用换行命令：双反斜线 $\backslash\backslash$ 手动控制换行。

1.3.2 撇号的输入

公式中字母右上角的撇号用键盘上分号键后面的 `'` 这个撇号输入，两撇则输入两次撇号而不是双引号，以此类推。

命令： `f'` `f''` `f'''`

效果： f' f'' f'''



如果要输入西文状态的单引号，则应该用 `'`（键盘上数字 1 前面的那个符号）输入。

1.3.3 带圈字符的输入

一般情况下，小标题的序号只希望占用一个字符的位置，可以用下面的命令输入，字母也可以换成数字。

命令： `\textcircled{a}` `\textcircled{c}` `\textcircled{e}`

效果： ① ③ ⑤

1.3.4 给公式加个框

有的时候需要给公式或者是文字增加一个边框，达到突出显示的目的，根据边框的样式，可以使用 `\boxed{}`、`\fbox{}`、`\doublebox{}`、`\ovalbox{}`、`\shadowbox{}` 等命令，在 \LaTeX 中 `\boxed{}`、`\fbox{}` 这两个命令的使用环境和显示效果是有区别的，但在 GeoGebra 中显示效果完全相同。

命令： `\boxed{P_n^m = \frac{n!}{(n-m)!}}`

`\fbox{P_n^m = \frac{n!}{(n-m)!}}`

`\doublebox{P_n^m = \frac{n!}{(n-m)!}}`

`\ovalbox{P_n^m = \frac{n!}{(n-m)!}}`

`\shadowbox{P_n^m = \frac{n!}{(n-m)!}}`

效果： 

除了上面讲到的几个边框样式， GeoGebra 还可以输出很多不同效果的边框，例如下面这条命令：`\fcolorbox{blue}{red}{\color{yellow}}` 蓝边，红底，黄字 可以得到这样的效果：

蓝边，红底，黄字

在文档中过多的使用太“炫”的效果可能会适得其反，所以应该合理使用不同的效果，真正做到为内容服务。

1.4 空白间距

在使用 \LaTeX 编辑公式时，插入到公式中的空格在编译时会被自动忽略，如果希望在输出的结果中体现出空格的位置，根据空格的大小和方向，可以使用不同的命令来生成。

1.4.1 水平间距

使用 表 1.1 中的命令可以在公式中生成合适大小的水平方向的空白间距，其中负间距命令 $\backslash!$ 可以用来减小间距。

表 1.1: 水平间距

命令	间距	命令	间距
$\backslash,$	$3/18\text{ em}$	\backslashenspace	0.5 em
$\backslash:$	$4/18\text{ em}$	\backslashquad	1 em
$\backslash;$	$5/18\text{ em}$	\backslashqqquad	2 em
$\backslash!$	$-3/18\text{ em}$		

在 GeoGebra 5 文本对话框中的“ \LaTeX 公式”下拉列表中预设了一个空格命令： $\backslash;$ 即 $5/18\text{ em}$ ，我们可以输入其它空格命令来产生不同的间距，这些命令可以连续使用也可以混合使用，从而得到更大或更小的间隔。



em 是一个相对单位，等于当前字体尺寸，约等于当前字体大写字母 M 的宽度，中文环境下相当于一个汉字的宽度。

1.4.2 长度可控的水平间距

如果需要的空格很长，则可以使用 $\backslashhspace{\text{长度}}$ 命令生成一个长度可控的水平空格，还可以配合下划线命令 $\backslashunderline{\quad}$ 一起使用来生成一个用于填空题的长空格，命令中的“ 10em ”是设定的长度，可以根据实际需要进行修改，还可以使用“虚位”命令达到同样的效果，关于虚位命令的介绍请参看第 2.12 节的内容。效果如下：

命令：文字 $\backslashhspace{10\text{em}}$ 中间有空格

文字 $\backslashunderline{\backslashhspace{10\text{em}}}$ 中间有空格

效果：文字 中间有空格

文字 _____ 中间有空格

1.4.3 垂直间距

如果想要生成类似行间距^①那样的垂直方向的空格，则需使用 `\vspace{长度}` 这一条命令，使用方法类似 `\hspace{ }`，在 GeoGebra 中很少用到此命令。

1.5 字号的设置

GeoGebra 软件本身有多处地方可以改变文字的大小（即字号），但它们的作用范围都是整个对象，如果在使用 \LaTeX 编辑公式时，需要改变某一局部文字的大小，特别是修改标签文字的大小，则可以使用表 1.2 中的命令来改变文字的字号。

1.5.1 相对字号命令

需要注意的是，表 1.2 中给出的修改字号的命令是相对命令，基本字号（`\normalsize`）的大小发生改变后，则相对字号命令所得到的结果也会相应的改变，这里给出了三种情况下的参考值，大家也可以从 GeoGebra 的选项菜单中修改软件的字号来观察这些命令在不同字号时的显示效果。

表 1.2: 相对字号命令

命令	默认值 = 10 pt	默认值 = 11 pt	默认值 = 12 pt
<code>\tiny</code>	5 pt	6 pt	6 pt
<code>\scriptsize</code>	7 pt	8 pt	8 pt
<code>\footnotesize</code>	8 pt	9 pt	10 pt
<code>\small</code>	9 pt	10 pt	10.95 pt
<code>\normalsize</code>	10 pt	11 pt	12 pt
<code>\large</code>	12 pt	12 pt	14.4 pt
<code>\Large</code>	14.4 pt	14.4 pt	17.28 pt
<code>\LARGE</code>	17.28 pt	17.28 pt	20.74 pt
<code>\huge</code>	20.74 pt	20.74 pt	24.88 pt
<code>\Huge</code>	24.88 pt	24.88 pt	24.88 pt

在 GeoGebra 的选项菜单中设置的字号和使用 `\normalsize` 这一条命令设置的字号尺寸是相同的，即基本字号的大小。



中文书籍基本字号为 5 号字，等于 10.5 pt。

^① 行间距有专门的命令：`\lineskip`。

1.5.2 相对字号修改示例

命令与需要修改的内容之间可以用空格分隔，也可以使用一对花括号将需要修改的内容括起来，两种写法都可以。

相对字号的修改

```
{\tiny Aa} {\scriptsize Aa} ..... {\LARGE Aa} {\huge Aa} {\Huge Aa}
```

Aa Aa Aa Aa Aa Aa Aa Aa Aa Aa



如果只改变局部文字的大小，则需要使用花括号把命令及需要改变字号的部分括起来，否则在 GeoGebra 中会把范围作用于命令之后的整行。命令：`ABC{\huge ABC}ABC` 得到的效果是 `ABC`**`ABC`**`ABC`。

1.5.3 文本缩放命令

如果 表 1.2 中的命令不能满足使用需求，我们还可以使用下面这条文本缩放命令，更加灵活的控制文本的大小及方向：

```
\scalebox{水平缩放倍数}[垂直缩放倍数]{要缩放的对象}
```

此命令对其作用的对象进行缩放，使缩放后对象的宽度和高度按照所给参数进行变化，其中“垂直缩放倍数”是可选参数，如果省略，那么将按照给定的水平缩放倍数，保持原始宽高比例进行缩放，如果缩放倍数是负值，则对象进行反射。

文本缩放

```
\scalebox{0.5}[2]{文字}
```

```
\scalebox{3}[2]{文字}
```

```
\scalebox{2}{}{文字}
```

```
\scalebox{2}[-2]{文字}
```

```
\scalebox{-2}[2]{文字}
```

婷

文字

文字

天

文字

1.5.4 让文本跟随对象一起缩放

利用相对字号命令设置的文本不会跟随对象的缩小或放大而进行缩放，如果想让文本跟随对象一起缩放，可以使用上面讲到的文本缩放命令：`\scalebox{ }` 配合“角点 (Corner)”指令来达到目的，方法如下：

文本跟随对象一起缩放

- 1、创建一个滑动条 **a**，设置其最小值为 0.1，最大值为 5
- 2、创建一个缩放倍数参数 **n**， $n = a * 20 / (y(\text{Corner}(3)) - y(\text{Corner}(1)))$
- 3、插入要缩放的文本，输入 图 1.1 所示的命令 `\scalebox{n}{文字}`，注意缩放倍数参数 **n** 必须在“空白公式框”中输入：

上面第二步操作中，如果改变缩放倍数参数 **n** 的分子中的 20 这个数值可以得到不同大小的初始文本。

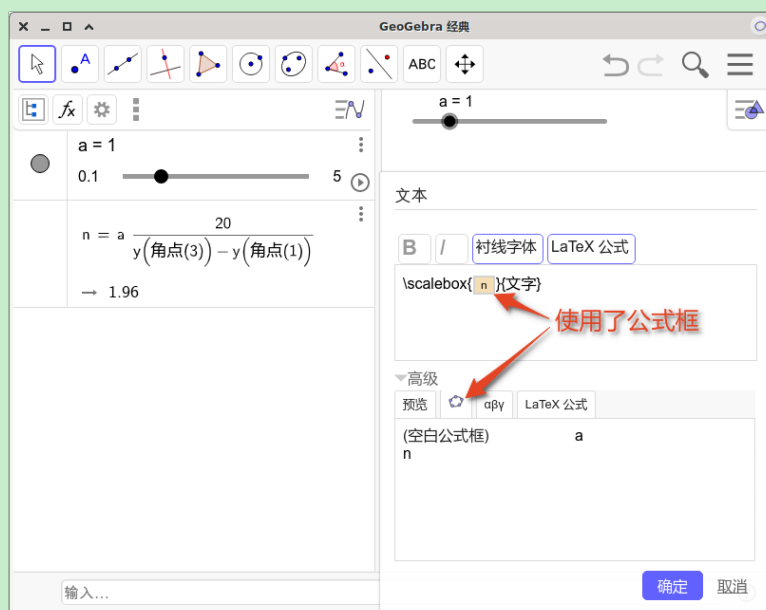


图 1.1: 文本跟随对象一起缩放

这样就得到了一个即可以使用滑动条 **a** 控制大小，也可以跟随对象用鼠标滚轮上下滚动而进行缩放的文本。

1.6 怎样使用 L^AT_EX

在 GeoGebra 中，基本上能输入文本的地方都可以使用到 L^AT_EX 代码，包括“标题”属性、“文本”工具、表格区、指令栏等，不同地方的使用方法上有细微区别，下面主要介绍在

“标题”属性和“文本”工具中使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的方法。

在版本 5 的表格区使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码后，如果没有显示期望的结果，则需要在“属性”菜单中单击“文本”选项卡；而在版本 6 的表格区使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码后，则需要在“属性”菜单中的“文本”选项卡里点选“ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 公式”。

1.6.1 在“标题”中使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

怎样修改标签文字的大小，这是一个经常被问及的问题，利用上文提到的设置字号的命令很容易就可以实现修改标签的大小。

直接修改

右击对象，打开对象的属性^①对话框，在标题这一栏中输入需要的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 字号代码，注意必须在代码的开头和结尾增加 $\$$ 符号进行 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 声明（其它地方不需要），然后把“显示标签”后面的内容改为“标题”，如 图 1.2 所示。

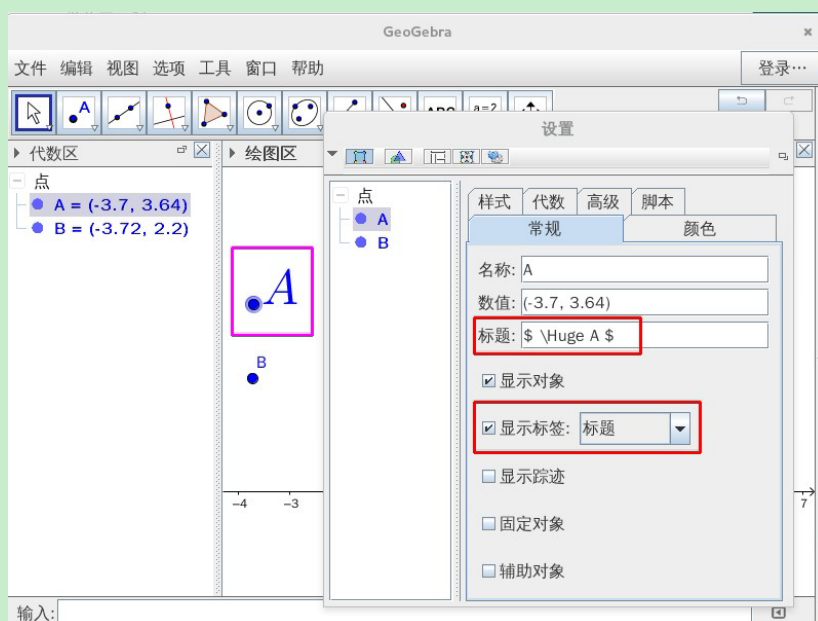


图 1.2: 在标题中使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

如前文所述，这里需要注意 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 命令代码和标签的名称之间有一个空格，如果标签的名称不止一个字符，则需要使用花括号 $\{ \}$ 将标签名称括起来。图 1.2 中点 B 的标签是原始大小，点 A 的标签是改变字号后的效果，在标题栏里支持可以在 GeoGebra 中使用的绝大部分 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码。

^① 版本 5 中叫属性，版本 6 中叫设置。

使用通用参数

在修改标签的时候，除了直接输入标签的名称外，还可以使用两个通用参数 **%n**（表示名称 name）、**%v**（表示数值 value），如 图 1.3 所示，这样的写法有一个好处，如果是要修改多个标签，则可以设置好一个标签后，把设置好的内容拷贝^①到其它对象的属性栏里即可，减少了击打键盘的次数。

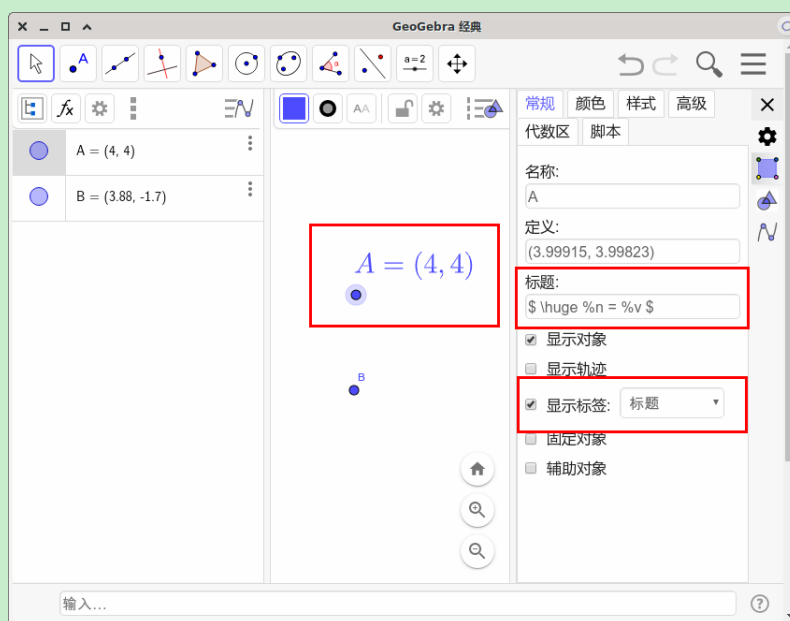


图 1.3: 使用通用参数

使用文本缩放命令

在 表 1.2 中给出的最大字号命令是 `\Huge{ }`，而使用此命令得到的字母大小约等于使用文本缩放命令：`\scalebox{2.5}{ }` 得到的字母大小，如果想要显示更大的标签字母，则可以使用文本缩放命令，将缩放倍数设置得大一点即可，也就是说，在显示效果上：

$$\backslash Huge\{B\} \approx \backslash scalebox\{2.5\}\{B\}$$

在修改标签文本的大小时，名称参数 **%n** 和数值参数 **%v** 同样适用于文本缩放命令中。

1.6.2 在“文本”工具中使用 L^AT_EX

在 GeoGebra 中 L^AT_EX 命令的主要使用环境是文本对话框，首先选择“文本”工具，然后在绘图区单击鼠标就可以打开文本对话框，具体操作步骤如下：

^① 在 GeoGebra 的很多地方都不支持用鼠标复制粘贴，但可以使用键盘 **Ctrl + c** 和 **Ctrl + v** 进行操作。

选择“文本”工具

如图 1.4 是 GeoGebra 5 的工具栏，图 1.5 是 GeoGebra 6 的工具栏，两个版本的工具栏基本没有差别，首先选择“文本”工具，然后在绘图区想要显示文本的地方^①单击鼠标；

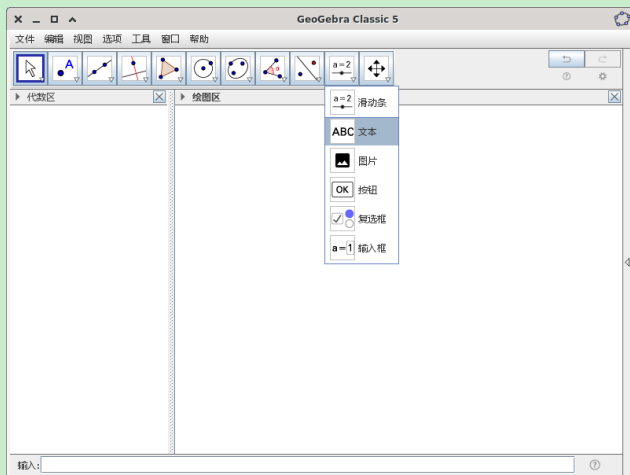


图 1.4: 版本 5 的文本工具

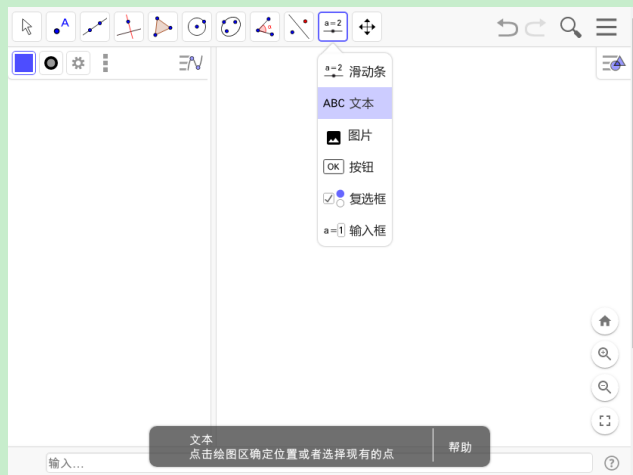


图 1.5: 版本 6 的文本工具

文本对话框的使用

接着在弹出的文本对话框中输入 \LaTeX 代码即可，版本 5 的文本对话框图 1.6 和版本 6 的文本对话框图 1.7 差异比较大，请大家注意区分使用。

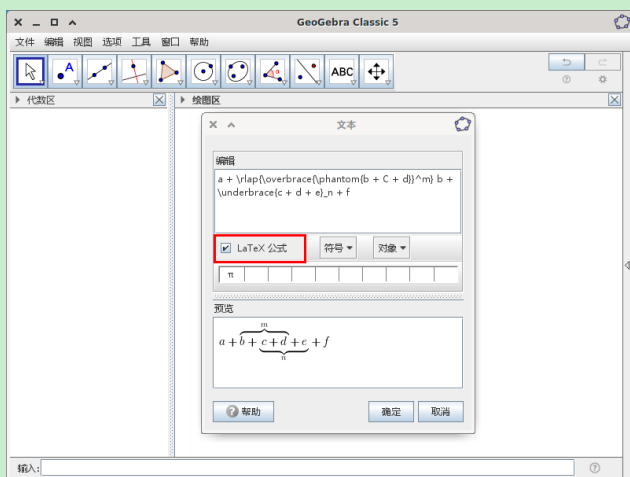


图 1.6: 版本 5 的文本对话框

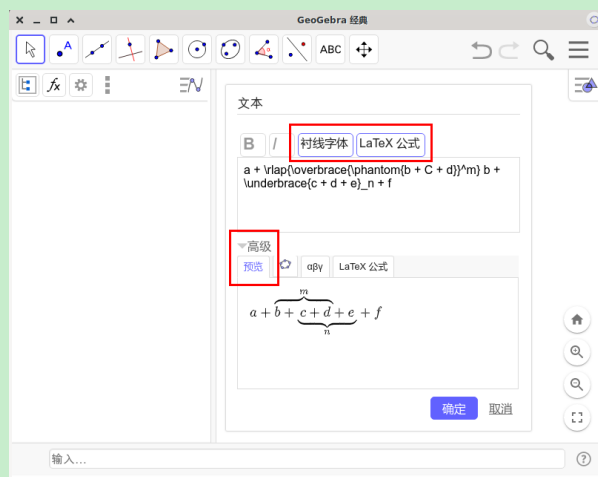


图 1.7: 版本 6 的文本对话框


GeoGebra 5 的文本对话框 在版本 5 中使用 \LaTeX 代码时需要勾选“LaTeX 公式”前的复选框，如图 1.6 所示，接下来就可以在编辑框中输入 \LaTeX 的代码了，在预览框中可以实时的看

^① 其实在任意位置单击鼠标即可，因为输入的文本是可以任意移动的。

到输出结果，文本对话框的大小和位置可以用鼠标拖动进行调整。

在文本对话框中的“LaTeX 公式”和“符号”这两个下拉列表中预设了许多常用的符号和命令，可以用鼠标选择使用，这大大减少了记忆量。“对象”下拉列表中默认有一个“空白公式框”（empty box），当需要显示动态文本的时候，这个工具非常有用，前文图 1.1 中的代码就用到了“空白公式框”这一命令。

GeoGebra 6 的文本对话框 在版本 6 中使用 LaTeX 代码时需要用鼠标点选“衬线字体”和“LaTeX 公式”这两个选项（版本 5 中默认就是衬线字体），要显示预览效果则需点击“高级”按钮打开预览框，如图 1.7 所示，版本 6 中的文本对话框只能移动位置而不能调整大小。

版本 6 的“空白公式框”（empty box）这一命令放到了 GeoGebra 的 logo 图标  选项卡中，常用的符号和公式分别放到了“ $\alpha\beta\gamma$ ”、“LaTeX 公式”两个选项卡中，GeoGebra 6 中预设的公式和符号没有 GeoGebra 5 中的多。

1.7 文字颜色和背景色

有时需要给公式添加一个背景色或者是修改特定字符的颜色，GeoGebra 软件本身的对象属性对话框中提供了颜色设置功能，但其作用范围是整个文本，使用起来缺少灵活性，此时就可以使用 LaTeX 代码来灵活设置颜色。

1.7.1 文字颜色

文字颜色也称为前景色，可以使用 `\color{ }{ }` 和 `\textcolor{ }{ }` 两条命令进行设置，这两条命令都是有两个参数，第一个参数是颜色代码，可以使用颜色的英文名称，也可以使用颜色的 RGB 编码或者是十六进制颜色码，第二个参数是要显示出来的内容。

修改文字颜色

```
\color{blue}{f(x)}=\prod_{i=1}^n\left(i-\frac{1}{2i}\right)
\color{255,0,0}{f(x)}=\prod_{i=1}^n\left(i-\frac{1}{2i}\right)
\textcolor{blue}{f(x)}=\int_1^{\infty}\textcolor{red}{x}\,\mathrm{d}x
\textcolor{0000FF}{f(x)}=\int_1^{\infty}\textcolor{FF0000}{x}\,\mathrm{d}x
```

$$f(x) = \prod_{i=1}^n \left(i - \frac{1}{2i}\right) \quad f(x) = \prod_{i=1}^n \left(i - \frac{1}{2i}\right) \quad f(x) = \int_1^{\infty} x \, dx \quad f(x) = \int_1^{\infty} x \, dx$$



仔细观察上表中命令的写法，可以发现，`\color` 和 `\textcolor` 两条命令在使用上有细微的区别，`\color` 命令需要用花括号 `{ }` 把整条命令括起来，否则命令作用范围将变成整行；而 `\textcolor` 的作用范围只是第二个参数。

1.7.2 背景色

修改背景色可以使用 `\bgcolor{ }{ }` 或者 `\colorbox{ }{ }` 命令来实现，两条命令得到的背景色效果有差异，具体的请看下面图 1.8 和图 1.9 中的例子。

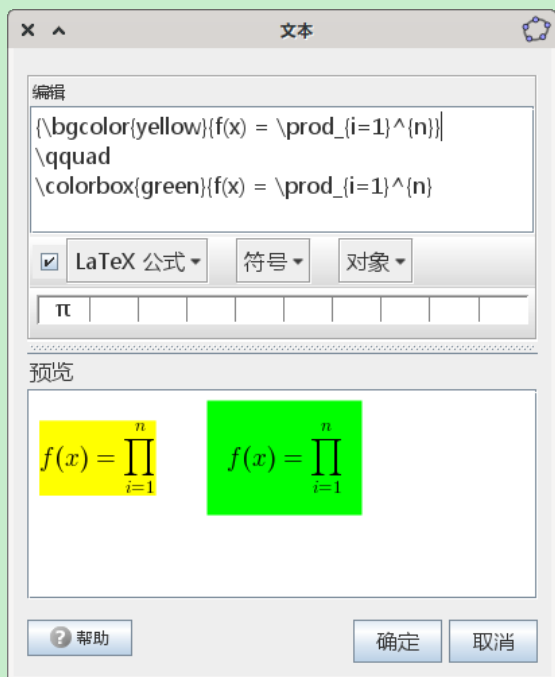


图 1.8: 版本 5 修改背景色

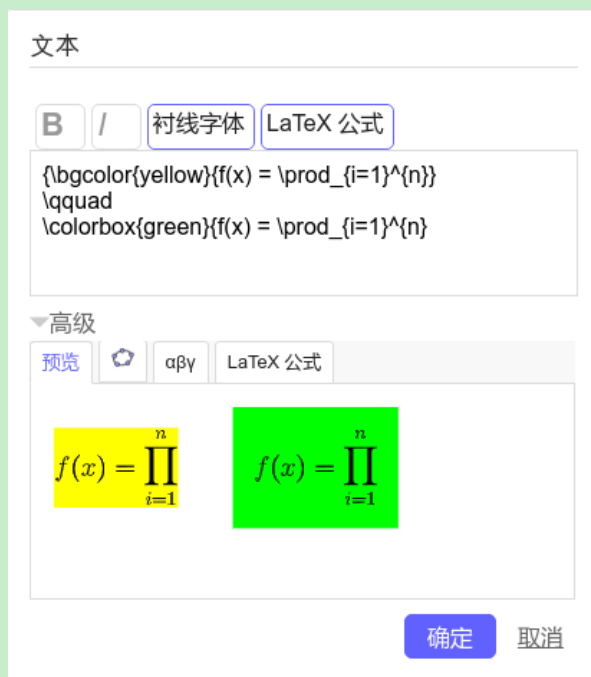


图 1.9: 版本 6 修改背景色

同样的，`\bgcolor{ }{ }` 命令也需要使用花括号 `{ }` 把整条命令括起来。

1.7.3 设置标签的颜色

前景色和背景色的设置同样可以应用到标签设置中，如图 1.10 和图 1.11 所示，在设置颜色的时候，如果省略表示颜色的代码，则以黑色显示。



修改字号、修改字体、修改前景色、修改背景色这些命令可以配合使用，得到不同效果的标签。

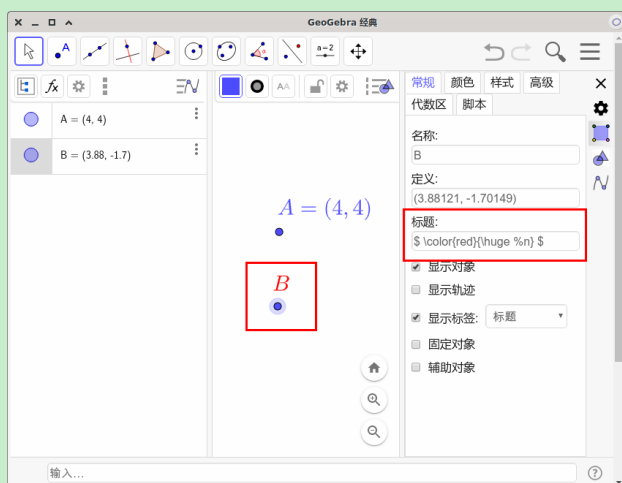


图 1.10: 修改标签前景色

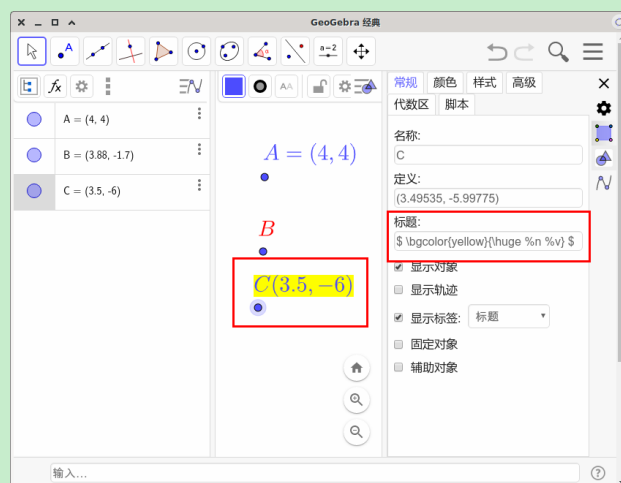


图 1.11: 修改标签背景色

1.8 数学字体

我们在编辑数学公式时，可能会根据实际需求而选用不同的字体样式，表 1.3 中给出了一些改变数学字体的命令及显示效果，需要注意的是，有的命令只针对大写字母有效。

表 1.3: 数学字体

命令	字体名	示例	效果
默认值		abcABC	<i>abcABC</i>
<code>\mathrm</code> 或 <code>\rm</code>	罗马体	<code>\rm{abcABC}</code>	abcABC
<code>\mathbf</code> 或 <code>\bf</code>	黑体	<code>\bf{abcABC}</code>	abcABC
<code>\mathsf</code> 或 <code>\sf</code>	等线体	<code>\sf{abcABC}</code>	abcABC
<code>\mathtt</code> 或 <code>\tt</code>	打字机字体	<code>\tt{abcABC}</code>	abcABC
<code>\mathfrak</code> 或 <code>\frak</code>	哥德体	<code>\frak{abcABC}</code>	abc℔BC
<code>\mathit</code> 或 <code>\mit</code> , <code>\it</code>	斜体	<code>\it{ABCXYZ}</code>	<i>ABCXYZ</i>
<code>\mathcal</code> 或 <code>\cal</code>	书法体	<code>\cal{ABCXYZ}</code>	<i>ABCXYZ</i>
<code>\mathscr</code> 或 <code>\scr</code>	草书体	<code>\scr{ABCXYZ}</code>	<i>ABCXYZ</i>
<code>\mathbb</code> 或 <code>\Bbb</code>	黑板体	<code>\Bbb{ABCXYZ}</code>	ABCXYZ
<code>\mathds</code>		<code>\mathds{ABCXYZ}</code>	ABCXYZ
<code>\pmb</code>		<code>\pmb{abcABC}</code>	<i>abcABC</i>

更多的数学符号有专门的命令，而不是简单的改变字体，关于各种符号的介绍请参看下一章。



在 GeoGebra 中，大多数修改字体的命令可以省略 `math` 这几个字母，只写后面的部分，之所以会出现两个命令，是因为在 \LaTeX 中有文本环境和数学环境之分。

1.9 在其它软件中使用 \LaTeX 公式

如果希望把用 \LaTeX 编辑生成的这些漂亮的公式用于别的不支持 \LaTeX 语言的软件中，可以把生成的公式导出成图片^①，然后再把图片导入到别的软件中使用，具体操作如下：

1.9.1 右键框选公式

公式编辑好后，如果直接导出则会发现图片的空白部分很多，解决的办法就是，按住鼠标右键拖动选择需要导出的部分，然后再导出，如图 1.12 所示。

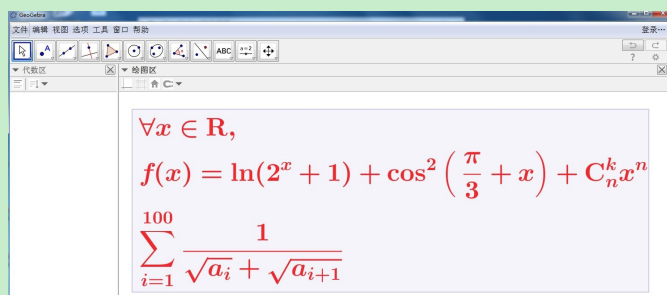


图 1.12: 右键框选公式

1.9.2 导出图片

利用右键框选好需要的部分后，单击“文件”菜单，选择“导出”--“图片”，导出图片时，选择 png 格式，并勾选“透明”这一选项（默认已经选中），如图 1.13 所示，这样就可以得到占用空间小且背景透明的图片。

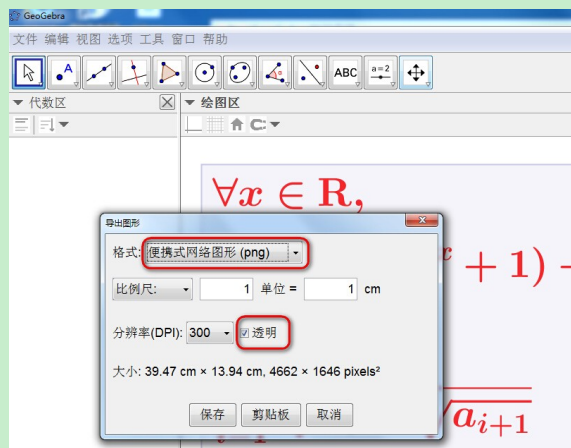


图 1.13: 导出图片

① 在此感谢贵州的邹颖老师首先提出这一方法和所截图片。

版本 5 的导出功能比版本 6 的导出功能要强大一些，版本 5 能导出多种格式的文件，且导出的图片格式也多，并且能进行一些细节设置，而版本 6 只能直接导出图片，但不能进行细节设置（希望后续的版本能改进），在使用导出功能时，建议使用版本 5 来导出，这样能得到更好效果的图片。

同样的方法导出 png 格式的图片，版本 5 得到的图片背景是透明的，而版本 6 得到的则是有背景颜色的图片。

1.9.3 在其它软件中使用

在 GeoGebra 中导出图片后，通过其它软件的插入图片功能，将所导出的图片插入到软件中，如 图 1.14 所示，使用版本 5 导出的图片，可以看出公式背景是透明的。

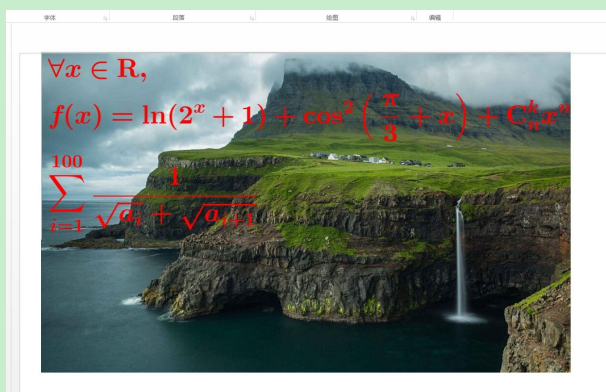


图 1.14: 在其它软件中使用

1.10 行内公式和行间公式

在 \LaTeX 中，数学模式有两种表现形式：行内模式（inline）和行间模式（display）^①，行内模式输入的公式会被压缩，让其和同一行中的文字上下对齐，行间模式输入的公式不会进行压缩处理。

1.10.1 行内公式

行内公式的意思就是文字和公式处于一行中，在 GeoGebra 中，默认情况下输入的公式是行间公式，如果要使用行内公式，需要在公式两边加上定界符： $\backslash($ 和 $\backslash)$ 。

^① 有的书籍也会把行间模式翻译为特显模式、独立模式等，我个人喜欢称之为行内公式和行间公式，这样从字面上就很容易理解公式出现的位置。

行内公式

行内公式: $\backslash(\backslash\text{sum_}\{i=1\}^n i = \backslash\text{frac}\{n(n+1)\}\{2\}\backslash)$.

行内公式: $\sum_{i=1}^n i = \frac{n(n+1)}{2}$.

使用行内公式的形式输出带有上标和下标的公式时，上标和下标的位置是被压缩的，分数（式）的大小同样也被压缩了，使其与同一行的文字高度一样。

1.10.2 行间公式

行间公式的意思就是在文字和公式混合排版时，公式独立占用一行，此时的公式比较漂亮，这也是 GeoGebra 的默认显示方式，GeoGebra 中省略了行间公式的定界符 $\backslash[$ 和 $\backslash]$ ，直接输入公式即可。

行间公式

行间公式: $\backslash\text{sum_}\{i=1\}^n i = \backslash\text{frac}\{n(n+1)\}\{2\}$.

行间公式:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}.$$



注意比较行内公式和行间公式上下标的位置以及公式的大小。

第二章 数学符号及公式

在 GeoGebra 5 文本对话框的“LaTeX 公式”和“符号”这两个下拉列表（GeoGebra 6 是在文本对话框的“ $\alpha\beta\gamma$ ”选项卡和“LaTeX 公式”选项卡）中提供了许多符号和命令，到目前为止（Ver: 5.0.592.0-d）版本 5 中预设的符号和命令要比版本 6 丰富一些。

2.1 长度单位

在上一章的第 1.4 节中介绍了一些空白间距（空格）的命令，GeoGebra 中可以使用不同的长度单位，同一个命令使用不同的长度单位得到的空白间距也是有差异的。

表 2.1: 希腊字母

单位	名称	说明	单位	名称	说明
mm	毫米	1mm = 2.845pt	cm	厘米	1cm = 28.453pt
pt	点 (point)	1pt = 0.351mm	cc	西塞罗	1cc = 4.513mm=12dd=12.84pt
bp	大点	1bp = 0.353mm \approx 1pt	in	英寸 (inch)	1in = 25.4mm = 72.27pt
dd	迪多 (didot)	1dd = 0.376mm = 1.07pt	ex	ex	1ex = 当前字体中 x 的高度
pc	派卡 (pica)	1pc = 12pt	em	em	1em = 当前字体尺寸 \approx 当前字体中 M 的宽度
sp	定标点	65536sp = 1pt			

这些长度单位可以分为两大类：绝对长度单位，它有固定不变的数值，例如 mm、cm、pt 等；相对长度单位，其数值的大小与当前所用文字的字号相关，例如 ex、em 等，当使用中文五号字时，1 em = 10.5 pt。

不同长度单位示例

命令：文字 `\underline{\hspace{10cm}}` 中间有空格

文字 `\underline{\hspace{10cc}}` 中间有空格

文字 `\underline{\hspace{10em}}` 中间有空格

效果：文字 _____ 中间有空格

文字 _____ 中间有空格

文字 _____ 中间有空格

2.2 基本元素

2.2.1 希腊字母

在 GeoGebra 中提供了全部的希腊字母，但为了便于大家查询，所以这里还是把希腊字母的命令用表 2.2 列举出来。

当命令的首字母大写时，即为输出大写希腊字母，如：

大小写希腊字母举例

命令： `\pi` `\phi` `\delta` `\sigma`

`\Pi` `\Phi` `\Delta` `\Sigma`

字母： π ϕ δ σ

Π Φ Δ Σ



需要注意的是，有一些大写希腊字母的显示效果和大写英文字母完全相同，所以在 \LaTeX 中没有专门的命令，但在 GeoGebra 中所有大写希腊字母都有对应的命令。

2.2.2 上标和下标

指数或上标使用上尖号 `^` 表示，下标使用 `_` 表示，上下标命令默认只作用于命令之后的一个字符，如果多于一个字符，需要使用一对花括号 `{ }` 将多个上下标字符括起来。

表 2.2: 希腊字母

命令	字母	命令	字母	命令	字母	命令	字母
<code>\alpha</code>	α	<code>\beta</code>	β	<code>\gamma</code>	γ	<code>\delta</code>	δ
<code>\epsilon</code>	ϵ	<code>\varepsilon</code>	ε	<code>\zeta</code>	ζ	<code>\eta</code>	η
<code>\theta</code>	θ	<code>\vartheta</code>	ϑ	<code>\iota</code>	ι	<code>\kappa</code>	κ
<code>\lambda</code>	λ	<code>\mu</code>	μ	<code>\nu</code>	ν	<code>\xi</code>	ξ
<code>\omicron</code>	\omicron	<code>\pi</code>	π	<code>\varpi</code>	ϖ	<code>\rho</code>	ρ
<code>\varrho</code>	ϱ	<code>\sigma</code>	σ	<code>\varsigma</code>	ς	<code>\tau</code>	τ
<code>\upsilon</code>	υ	<code>\phi</code>	ϕ	<code>\varphi</code>	φ	<code>\chi</code>	χ
<code>\psi</code>	ψ	<code>\omega</code>	ω				
<code>\Alpha</code>	\AA	<code>\Beta</code>	\AA	<code>\Gamma</code>	Γ	<code>\Delta</code>	Δ
<code>\Epsilon</code>	\AA	<code>\Zeta</code>	\AA	<code>\Eta</code>	\AA	<code>\Theta</code>	Θ
<code>\Iota</code>	\AA	<code>\Kappa</code>	\AA	<code>\Lambda</code>	Λ	<code>\Mu</code>	\AA
<code>\Nu</code>	\AA	<code>\Xi</code>	\AA	<code>\Omicron</code>	\AA	<code>\Pi</code>	\AA
<code>\Rho</code>	\AA	<code>\Sigma</code>	\AA	<code>\Tau</code>	\AA	<code>\Upsilon</code>	\AA
<code>\Phi</code>	\AA	<code>\Chi</code>	\AA	<code>\Psi</code>	\AA	<code>\Omega</code>	\AA

上标和下标

命令: `a_{i}` `x_{ij}^{\{3\}}` `x^{1/2}` `e^{\{x^2\}} \neq \{e^x\}^2` `\{a\}_{b} M_{\{c\}}^{\{d\}}`

效果: a_i x_{ij}^3 $x^{1/2}$ $e^{x^2} \neq e^{x^2}$ ${}_b^a M_c^d$

一个良好的书写习惯应该是即使上标或下标只有一个字符，也应该用花括号将其括起来，这样即减少了出错的机率，也增加了代码的可读性。

上标和下标综合应用

上标和下标应用举例

$$e^{i \pi} + 1 = 0$$

$$z = r \cdot e^{2 \pi i}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$e^{i\pi} + 1 = 0 \qquad z = r \cdot e^{2\pi i} \qquad \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

2.2.3 分数和根号

分数（式）用 `\frac{分子}{分母}` 命令表示，根号（式）用 `\sqrt[指数]{被开方数}` 命令来表示，根指数为 2 时可以省略，两条命令都可以嵌套使用。花括号 `{ }` 表示必选参数不能省略，方括号 `[]` 表示可选参数可以省略。

分式和根式

命令： `\frac{1}{2}`

`\frac{1}{x+2}`

`\sqrt{x}`

`\sqrt[5]{x^2+y}`

`\frac{2^{\frac{1}{2}}}{\sqrt{5}-3}`

`\sqrt[3]{5+\sqrt{x+1}}`

效果： $\frac{1}{2}$ $\frac{1}{x+2}$ \sqrt{x} $\sqrt[5]{x^2+y}$ $\frac{2^{\frac{1}{2}}}{\sqrt{5}-3}$ $\sqrt[3]{5+\sqrt{x+1}}$

繁分数

书写繁分数的时候，如果使用 `\frce{ }{ }` 命令，分数会被压缩，数字显得比较小，为了得到更好的排版效果，书写繁分数时应该使用 `\cfrac{分子}{分母}` 命令，注意对比下面分别用 `\frac` 和 `\cfrac` 命令得到的不同效果。

繁分数： `frac` 和 `cfrac` 比较

命令： `\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \cdots}}`

`\cfrac{1}{\sqrt{2} + \cfrac{1}{\sqrt{2} + \cdots}}`

效果： $\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \cdots}}$ $\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \cdots}}$

2.2.4 二项式结构

类似于二项式结构的书写，可以使用 `\mathrm{C}` 或 `\rm{C}`^① 以及使用 `\choose`、`\binom` 和 `\atopwithdelims()` 等几条命令来实现，不同的命令在格式上有一些区别，具体请看下面的几个例子：

^① 实际上 `\mathrm` 是改变字体的命令，可以简写为 `\rm`。

二项式结构

命令: `\rm{C}_n^k = \frac{n!}{k!(n-k)!}`

`{n \choose k} = \frac{n!}{k!(n-k)!}`

`\binom{n}{k} = \frac{n!}{k!(n-k)!}`

`{n \atopwithdelims() k} = {n! \above 1pt k!(n-k)!}`

效果: $C_n^k = \frac{n!}{k!(n-k)!}$ $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ $\binom{n}{k} = \frac{n!}{k!(n-k)!}$



利用 `\atopwithdelims()` 命令还可以得到其他一些复杂的结构, 命令中的圆括号和花括号都不能省略。`\above` 命令是画分数线, 1pt (数字 1) 是线条的粗细, 可以自行修改这一数值得到不同的效果。

2.2.5 省略号

在数学环境中, 有不同位置、不同方向的省略号, 分别使用不同的命令来实现, 具体看下面的例子。

省略号

命令: `\dots` `\cdots` `\ldots` `\dotsb` `\vdots` `\ddots` `\iddots`

效果: ⋮ ⋱ ⋰



在 \LaTeX 中使用 `\dots` 命令得到的省略号会根据前后文的情况自动调整水平对齐的位置, 但在 GeoGebra 中 `\dots` 和 `\ldots` 命令的效果似乎没有区别。

2.3 运算符

像 `+` `-` `=` `<` `>` 这些键盘上有的运算符可以直接使用键盘输入, 更多的符号则需要使用专门的命令来输入, 这里列举了一些常用符号, 与 GeoGebra 的文本对话框中预设的字母和符号有重复的。

2.3.1 常用二元运算符

表 2.3 列出了常用的二元运算符。

表 2.3: 常用二元运算符

命令	符号	命令	符号	命令	符号	命令	符号
<code>\times</code>	\times	<code>\div</code>	\div	<code>\cdot</code>	\cdot	<code>\setminus</code>	\setminus
<code>\pm</code>	\pm	<code>\mp</code>	\mp	<code>\star</code>	\star	<code>\ast</code>	\ast
<code>\cup</code>	\cup	<code>\cap</code>	\cap	<code>\bigcup</code>	\bigcup	<code>\bigcap</code>	\bigcap
<code>\vee</code>	\vee	<code>\wedge</code>	\wedge	<code>\bigvee</code>	\bigvee	<code>\bigwedge</code>	\bigwedge
<code>\Cup</code>	\Cup	<code>\Cap</code>	\Cap	<code>\uplus</code>	\uplus	<code>\biguplus</code>	\biguplus
<code>\sqcup</code>	\sqcup	<code>\sqcap</code>	\sqcap	<code>\bullet</code>	\bullet	<code>\circ</code>	\circ
<code>\oplus</code>	\oplus	<code>\ominus</code>	\ominus	<code>\otimes</code>	\otimes	<code>\oslash</code>	\oslash
<code>\bigoplus</code>	\bigoplus	<code>\bigotimes</code>	\bigotimes	<code>\bigodot</code>	\bigodot	<code>\bigcirc</code>	\bigcirc
<code>\odot</code>	\odot	<code>\wr</code>	\wr	<code>\dagger</code>	\dagger	<code>\ddagger</code>	\ddagger
<code>\lhd</code>	\lhd	<code>\rhd</code>	\rhd	<code>\unlhd</code>	\unlhd	<code>\unrhd</code>	\unrhd
<code>\triangleleft</code>	\triangleleft	<code>\triangleright</code>	\triangleright	<code>\diamond</code>	\diamond	<code>\amalg</code>	\amalg

2.3.2 常用关系符

表 2.4 列出了常用的二元关系符，更多符号请查阅相关文档。

表 2.4: 常用二元关系符

命令	符号	命令	符号	命令	符号	命令	符号
<code>\leq</code> 或 <code>\le</code>	\leq	<code>\geq</code> 或 <code>\ge</code>	\geq	<code>\nless</code>	\nless	<code>\ngtr</code>	\ngtr
<code>\ll</code>	\ll	<code>\gg</code>	\gg	<code>\lll</code>	\lll	<code>\ggg</code>	\ggg
<code>\leqslant</code>	\leqslant	<code>\geqslant</code>	\geqslant	<code>\mid</code>	\mid	<code>\nmid</code>	\nmid
<code>\neq</code>	\neq	<code>\equiv</code>	\equiv	<code>\not\equiv</code>	$\not\equiv$	<code>\approx</code>	\approx
<code>\doteq</code>	\doteq	<code>\doteqdot</code>	\doteqdot	<code>\sim</code>	\sim	<code>\backsimeq</code>	\backsimeq
<code>\nsim</code>	\nsim	<code>\simeq</code>	\simeq	<code>\backsimeq</code>	\backsimeq	<code>\cong</code>	\cong
<code>\in</code>	\in	<code>\notin</code>	\notin	<code>\ni</code>	\ni	<code>\not\ni</code>	$\not\ni$
<code>\subset</code>	\subset	<code>\supset</code>	\supset	<code>\subseteq</code>	\subseteq	<code>\supseteq</code>	\supseteq
<code>\subsetneq</code>	\subsetneq	<code>\supsetneq</code>	\supsetneq	<code>\subsetneqq</code>	\subsetneqq	<code>\supsetneqq</code>	\supsetneqq
<code>\asymp</code>	\asymp	<code>\sqsubset</code>	\sqsubset	<code>\sqsupset</code>	\sqsupset	<code>\Join</code>	\Join
<code>\lessgtr</code>	\lessgtr	<code>\gtrless</code>	\gtrless	<code>\lesseqgtr</code>	\lesseqgtr	<code>\gtreqless</code>	\gtreqless
<code>\lesseqqgtr</code>	\lesseqqgtr	<code>\gtreqqless</code>	\gtreqqless	<code>\smile</code>	\smile	<code>\frown</code>	\frown

2.3.3 大尺寸运算符

像累加、累乘、极限、积分、曲线积分等等这些大尺寸运算符的表示方法如表 2.5，如果要让定积分的上标和下标位于积分号的上下方，可以使用 `\limits` 命令进行标识。

表 2.5: 大尺寸运算符

命令	符号	命令	符号	命令	符号
<code>\sum</code>	Σ	<code>\prod</code>	Π	<code>\coprod</code>	\amalg
<code>\lim</code>	\lim	<code>\int</code>	\int	<code>\oint</code>	\oint

在 \LaTeX 中推荐用命令 `\prod` 来表示累乘，而不是用 π 的大写 `\Pi`，在 GeoGebra 中两个符号的大小有明显的差别。

大尺寸运算符举例

命令: `\sum_{i=1}^n i` `\prod_{i=1}^n` `\lim_{x \to 0} x^2`
`\int_a^b x^2 dx` `\int\limits_a^b x^2 dx` `\oint`

效果: $\sum_{i=1}^n i$ $\prod_{i=1}^n$ $\lim_{x \rightarrow 0} x^2$ $\int_a^b x^2 dx$ $\int_a^b x^2 dx$ \oint

修改积分变量

追求完美的老师可能会觉得积分公式末尾的积分变量 dx 改成直体的 $\mathrm{d}x$ 比较好看，另外积分函数和积分变量之间需要拉开一点距离，那么我们可以在公式中增加空白间距和改变数学字体来达到这一目的。

修改积分变量

命令: `\int_a^b x^2 dx` `\int_a^b x^2 \;\mathrm{d}x`

效果: $\int_a^b x^2 dx$ $\int_a^b x^2 \mathrm{d}x$

关于美观方面的设置，其实应该是一个仁者见仁的问题。

多重积分符号

多重积分如果用多个 `\int` 命令来输入的话，积分号之间的距离会显得过宽，正确的方法是使用 `\iint`、`\iiint`、`\iiiiint` 等命令来输入，最高支持四重，如果高于四重的则需使用 `\idotsint` 命令来输入，可以从下表中比较一下两种方法的差异。

多重积分

命令: `\int` `\int\int` `\int\int\int` `\int\int\int\int` `\int\cdots\int`
`\iint` `\iiint` `\iiiiint` `\idotsint`

效果: \int \iint \iiint \iiiiint $\int \cdots \int$

第一行的命令使用方法是错误的，正确的应该是第二行的方法。

曲面积分符号

在 GeoGebra 中曲线积分 \oint 可以直接使用命令 `\oint` 得到，但曲面积分 \oiint 的符号并不支持 \LaTeX 中命令 `\oiint` 的用法，不过可以通过使用 Unicode 代码的方式调用曲面积分的符号，有十进制和十六进制两种调用方式，如果使用十六进制则需要在代码中用 x 进行标识。

表 2.6: 曲面积分符号

符号	十进制代码	十六进制代码
\oiint	<code>\unicode{8751}</code>	<code>\unicode{x222F}</code>
\oiiint	<code>\unicode{8752}</code>	<code>\unicode{x2230}</code>
\ooint	<code>\unicode{8753}</code>	<code>\unicode{x2231}</code>
\oslash	<code>\unicode{8754}</code>	<code>\unicode{x2232}</code>
\oslash	<code>\unicode{8755}</code>	<code>\unicode{x2233}</code>

2.3.4 将下标分行

有时一些大尺寸运算符的参数不止一个，为了公式的美观，使用 `\substack{ }` 命令可以把下标分成多行来写书，此方法同样适用于上标。`\begin{subarray}{ } \dots \end{subarray}` 环境也可以排版分行的上下标，功能更加强大。

将下标分行

`\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i,j) =`

`\sum_{\substack{1 \in I \\ 1 < j < m}} Q(i,j)`

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{1 \in I \\ 1 < j < m}} Q(i, j)$$

2.4 箭头

在表 2.7 中给出了一些箭头的输入方法，`\xleftarrow` 和 `\xrightarrow` 这两条命令生成的箭头可以根据内容自动调整长度。

表 2.7: 箭头

命令	箭头	命令	箭头
<code>\leftarrow</code>	\leftarrow	<code>\rightarrow</code>	\rightarrow
<code>\longleftarrow</code>	\longleftarrow	<code>\longrightarrow</code>	\longrightarrow
<code>\leftrightarrow</code>	\leftrightarrow	<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\leftrightsquigarrow</code>	\leftrightsquigarrow	<code>\rightleftarrows</code>	\rightleftarrows
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Rrightarrow</code>	\Rrightarrow
<code>\nLeftarrow</code>	\nLeftarrow	<code>\nRrightarrow</code>	\nRrightarrow
<code>\Longleftarrow</code>	\Longleftarrow	<code>\Longrightarrow</code>	\Longrightarrow
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Longleftrightarrow</code>	\Longleftrightarrow
<code>\downarrow</code>	\downarrow	<code>\uparrow</code>	\uparrow
<code>\nwarrow</code>	\nwarrow	<code>\nearrow</code>	\nearrow
<code>\swarrow</code>	\swarrow	<code>\searrow</code>	\searrow
<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookleftarrow</code>	\hookleftarrow
<code>\mapsto</code>	\mapsto	<code>\longmapsto</code>	\longmapsto
<code>\leadsto</code>	\leadsto	<code>\Rsh</code>	\Rsh

命令: `\xleftarrow{x + y + z}` `\xrightarrow[x < y]{a * b * c}`

效果: $\xleftarrow{x+y+z}$ $\xrightarrow[x < y]{a*b*c}$

2.5 分隔符

分隔符也叫定界符，键盘上的圆括号 ()、方括号 [] 和花括号 { } 可以直接使用键盘输入，输入花括号时需要加上转义符 `\{ \}`，如果需要大尺寸的分隔符则需要下面介绍的这些命令。

2.5.1 大尺寸括号类分隔符

大尺寸的括号使用 `\langle \rangle \lVert \rVert \lfloor \rfloor \lceil \rceil` 等命令来表示，在 GeoGebra 中的绝对值符号使用键盘上的管道符 | 输入，我们可以在这些分隔符的

前面加上 `\big`、`\Big`、`\bigg`、`\Bigg` 这几个命令来调整它们的大小。

\LaTeX 中原有的方法是在分隔符前面加 `\left` 和 `\right` 命令让程序自动调整大小, GeoGebra 中的“ \LaTeX 公式”下拉列表中的“括号”就是采用的这种方法, 但有的时候效果不佳, 可以根据实际情况来选择不同的方法以便达到最佳的显示效果。

括号类分隔符 1

命令: `\Bigg(\bigg(\Big(\big((x)\big)\Big)\bigg)\Bigg)`

`\Bigg[\bigg[\Big[\big[[x]\big]\Big]\bigg]\Bigg]`

`\Bigg\{\bigg\{\Big\{\big\{\{x\}\big\}\Big\}\bigg\}\Bigg\}`

效果: $\left(\left(\left(\left(x\right)\right)\right)\right)$ $\left[\left[\left[\left[x\right]\right]\right]\right]$ $\left\{\left\{\left\{\left\{x\right\}\right\}\right\}\right\}$

括号类分隔符 2

命令: `\Bigg\langle\Big\langle\langle x \rangle\Big\rangle\Bigg\rangle`

`\Bigg|\bigg|\Big|\big|| x |\big|\Big|\bigg|\Bigg|`

`\Bigg\lVert\Big\lVert\lVert x \rVert\Big\rVert\Bigg\rVert`

`\Bigg\lfloor\Big\lfloor\lfloor x \rfloor\Big\rfloor\Bigg\rfloor`

`\Bigg\lceil\Big\lceil\lceil x \rceil\Big\rceil\Bigg\rceil`

效果: $\langle\langle\langle x\rangle\rangle\rangle$ $\|x\|$ $\lVert x \rVert$ $\lfloor x \rfloor$ $\lceil x \rceil$

很多时候, 得到相同的显示效果是可以有多条命令的, 例如 `\|`、`\Vert`、`\lVert`、`\rVert` 这几条命令都可以得到双竖线, 可以根据实际情况合理选用。



为了让代码显示得简短一些, 这儿并没有把所有分隔符的情况都写出来, 一般情况并不要求分隔符必须成对出现, 甚至还可以不同大小的不同分隔符配合使用, 可以自行验证。

2.5.2 集合公式中的分隔符

集合类公式中的分隔符可以使用管道符 `|` 或者使用 `\mid` 命令生成, 而这两种方法得到的分隔符是有区别的, 管道符生成的分隔符与前后字符离得太近, 但是可以跟 `\big`、`\bigg`、`\Big`、`\Bigg`、`\left`、`\right` 这些命令配合使用得到比较长的分隔符, 与前后字符的间隔可以用空白间距命令来调整。

集合公式中的分隔符

$$\begin{aligned} \backslash\mathrm{Bbb}\{Q\} &= \backslash\{x \mid x = \frac{q}{p}, p \in \backslash\mathrm{Bbb}\{N^+\}, q \in \backslash\mathrm{Bbb}\{Z\} \backslash\} \\ \backslash\mathrm{Bbb}\{Q\} &= \backslash\{x \left| x = \frac{q}{p}, p \in \backslash\mathrm{Bbb}\{N^+\}, q \in \backslash\mathrm{Bbb}\{Z\} \right. \backslash\}\backslash\mathrm{right}. \\ \backslash\mathrm{Bbb}\{Q\} &= \backslash\{x \Big| x = \frac{q}{p}, p \in \backslash\mathrm{Bbb}\{N^+\}, q \in \backslash\mathrm{Bbb}\{Z\} \backslash\} \\ x \mapsto &\backslash\{ \backslash, c \in C \mid c \leq x \backslash, \backslash\} \\ A = &\backslash\{ \backslash, x \in X \mid x \in X_{\backslash i} \backslash \backslash \backslash \mathrm{mbox}\{\backslash\mathrm{quad for some } \backslash i \in I \backslash, \backslash\} \end{aligned}$$

$$\mathbb{Q} = \{x \mid x = \frac{q}{p}, p \in \mathbb{N}^+, q \in \mathbb{Z}\}$$

$$\mathbb{Q} = \{x \left| x = \frac{q}{p}, p \in \mathbb{N}^+, q \in \mathbb{Z}\right. \}$$

$$\mathbb{Q} = \{x \Big| x = \frac{q}{p}, p \in \mathbb{N}^+, q \in \mathbb{Z}\}$$

$$x \mapsto \{c \in C \mid c \leq x\}$$

$$A = \{x \in X \mid x \in X_i \text{ for some } i \in I\}$$

上面第二个例子中，使用了 `\left`、`\right` 这两个命令，虽然得到了一条比较长的分隔符，但用法比较繁琐，不建议使用，在第 2.12.4 节中还会讲解使用虚位命令来达到这种效果。

2.6 多行公式

有的公式特别长一行放不下，需要手动为它们换行，或者是几个公式需要写成一组，还有些类似分段函数，需要给它加上一个左边的花括号，这时我们就要用到一些多行公式环境。

2.6.1 长公式

不需要对齐的长公式可以使用 `\multline` 环境，需要对齐的长公式可以使用 `\aligned` 环境，用 `\\` 命令来分行，并使用 `&` 符号来标识对齐的位置。

表 2.8: 长公式环境

不对齐	对齐
$\begin{aligned} x &= a+b+c+ \\ &d+e+f+g \end{aligned}$	$\begin{aligned} x &= &a+b+c+ \\ &&d+e+f+g \end{aligned}$
$\end{aligned}$	$\end{aligned}$

请大家验证 表 2.8 中的两组代码，并比较显示效果的差异。

2.6.2 公式组

不需要对齐的公式组可以使用 `gather` 环境，需要对齐的公式组可以使用 `align` 环境，并使用 `&` 来标识对齐的位置，比较一下 表 2.9 中两组公式的显示效果。

表 2.9: 公式组环境

不对齐	对齐
$\begin{gathered} x = a+b \\ \sum = i+j+k+l \end{gathered}$	$\begin{aligned} x &= a+b \\ \sum &= i+j+k+l \end{aligned}$

不对齐（使用 `gather` 环境）

不需要对齐的公式组使用 `gather` 环境，用 `\\` 命令来分行。

```
\begin{gather}
```

$$x = a+b$$

$$\sum = i+j+k+l$$

```
\end{gather}
```

$$x = a + b$$

$$\sum = i + j + k + l$$

对齐（使用 `align` 环境）

需要对齐的公式组使用 `align` 环境，用 `\\` 命令来分行，并使用 `&` 符号来标识对齐的位置，下面这个例子是等号对齐。

```
\begin{align}
```

$$x \&= a+b$$

$$\sum \&= i+j+k+l$$

```
\end{align}
```

$$x = a + b$$

$$\sum = i + j + k + l$$

2.6.3 分段函数

分段函数通常使用 `cases` 环境来写分支公式，并使用 `&` 符号来标识对齐的位置。

$$y = \begin{cases} -x, & x \leq 0 \\ x, & x > 0 \end{cases} \quad D(x) = \begin{cases} 1, & \text{if } x \in \mathbb{Q} \\ 0, & \text{if } x \in \mathbb{R} \setminus \mathbb{Q} \end{cases}$$

表 2.10: 分段函数

$y = \begin{cases} -x, & x \leq 0 \\ x, & x > 0 \end{cases}$	$D(x) = \begin{cases} 1, & \text{if } x \in \mathbb{Q} \\ 0, & \text{if } x \in \mathbb{R} \setminus \mathbb{Q} \end{cases}$
--------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------



说明一下表 2.10 中第二个例子的几个命令，`\text{if}` 的作用是把 if 改为文本以正体显示，并且保留 if 前后的空格，`\Bbb` 命令的作用是把数学字体^①改为黑板体，`\setminus` 命令的显示效果和 `\backslash` 的相同，就是画反斜线，但多用于公式中。

2.6.4 方程组

方程组可以用类似分段函数的写法来表示，也可以使用“[表格文本\(TableText\)](#)”这一条 GeoGebra 特有的指令来表示。

用指令书写方程组

命令：表格文本({ “ $3x + 5y = 3$ ” , “ $6x + 2y = 8$ ” }, “ $\{v\}$ ”)

TableText({ “ $3x + 5y = 3$ ” , “ $6x + 2y = 8$ ” }, “ $\{v\}$ ”)

指令可以在“指令栏”中直接使用，也可以在文本框中配合“空白公式框”一起使用，效果分别如图 2.1 和图 2.2 所示。

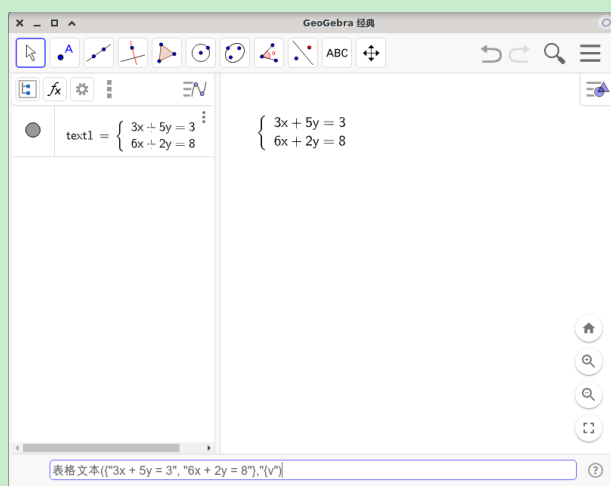


图 2.1: 直接在指令栏中使用

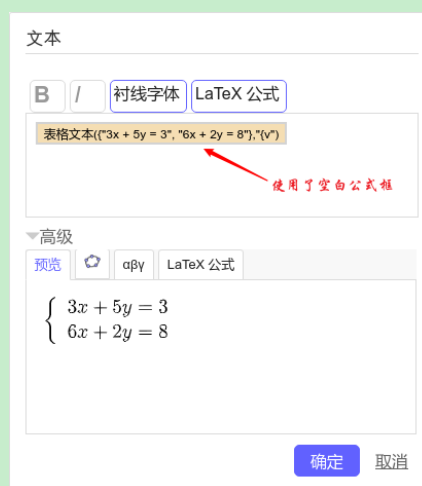


图 2.2: 配合空白公式框使用



指令名称可以使用中文，也可以使用英文，但指令中的符号和标点必须在英文状态

^① 默认情况 L^AT_EX 中的数学字体显示为斜体，关于数学字体的设置，请参看前文的介绍。

下输入，指令的参数放在一对圆括号 () 中，较早的版本是使用方括号，后来两种括号并行了一段时间，现在已经改为圆括号了。

2.7 矩阵

行列式和矩阵的生成方法是相同的，这里统一用矩阵来说明，生成矩阵的环境有多个，分别应用于不同分隔符的情况，这些环境中大部分都不支持自定义，支持自定义的是 `array` 这一环境，也是功能最强大的一个。



这些环境有一个共同的特点：都是用 `&` 符号来分隔一行中的元素，用 `\\` 来换行，这也是 \LaTeX 中的通用方式。

2.7.1 无分隔符的矩阵

可以用 `matrix` 环境生成没有分隔符也没有对齐参数（默认居中）的矩阵。

无分隔符的矩阵

```
\begin{matrix}
x_1 & x_2 & \dots \\
x_3 & x_4 & \dots \\
\vdots & \vdots & \ddots
\end{matrix}
```

```
x_1  x_2  ...
x_3  x_4  ...
⋮    ⋮    ⋱
```

2.7.2 带分隔符的矩阵

`pmatrix`、`bmatrix`、`Bmatrix`、`vmatrix`、`Vmatrix` 等环境可以在矩阵两边加上不同的分隔符，它们同样也没有对齐方式的参数。

带分隔符的矩阵

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

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

```
\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}
```

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

```
\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}
```

效果: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$ $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ $\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$

为了排版时显示得紧凑一些，这里把每一个矩阵写成了一行，大家也可以仿照上面的例子写成多行的样式，得到的输出效果是相同的。

2.7.3 较复杂的矩阵示例

在上面所说的众多环境中，基本上都不支持自定义功能，如果我们需要更加复杂的矩阵，比如要在矩阵内部画横线和竖线，甚至还有其他的一些要求，此时我们就需要使用 `array` 这一环境来达到目的。

较复杂的矩阵示例

```
(A,B) = \left( \begin{array}{cccc|c}
```

```
a & b & c & d & e \\
```

```
3 & 4 & 5 & 6 & -1 \\
```

```
k & f & i & -9 & 2 \\ \hline
```

```
s & x & L & R & E
```

```
\end{array}\right)_{m \times n}
```

$$(A,B) = \left(\begin{array}{cccc|c} a & b & c & d & e \\ 3 & 4 & 5 & 6 & -1 \\ k & f & i & -9 & 2 \\ \hline s & x & L & R & E \end{array} \right)_{m \times n}$$

`array` 环境具有非常强大的自定义功能，它提供了列对齐的参数，有三种方式：居左、居中、居右，分别用字母：`l`、`c`、`r` 表示，不同的列用符号 `&` 符号分隔，用 `\\` 表示换行，它还能在矩阵中根据需要来画横线和竖线^①。

在 `array` 环境生成的矩阵中，可以在对齐参数中使用管道符 `|` 来生成竖线，在需要生成横线的地方使用 `\hline` 命令。

用 `array` 环境生成分段函数

之前我们是使用 `cases` 环境来生成分段函数的，`array` 环境同样也能用来生成分段函数，使用起来更加灵活。

用 `array` 环境生成分段函数

```
\chi_A(x) = \left\{
\begin{array}{ll}
1, & x \in A \\
0, & x \notin A
\end{array} \right.
\end{array} \right.
```

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$



在使用 `array` 环境时，需要特别注意的是，输出结果的左右两边可以不加分隔符（括号），加分隔符时左右两边的符号可以不相同，但是 `\left` 和 `\right` 这两个命令必须成对出现，如果不想把分隔符显示出来，那么可以使用英文句号 `.` 来代替分隔符，请仔细观察上例中的右括号。

`array` 环境的嵌套

`array` 环境支持嵌套，可以使用嵌套的方法来书写一些比较复杂的矩阵，请看下面的例子，注意 `@{}` 命令的作用是压缩列之间的空白间距，这个例子代码相对长了一点，但用到的知识点并不多，大家多试几次也就能熟练掌握了。

^① 在 `GeoGebra` 中表示对齐方式的参数可以省略，默认为居中，但引用对齐参数的大括号 `{ }` 不能省略。

array 环境的嵌套

```

\left(\begin{array}{c@{ }c@{ }c}

\begin{array}{|cc|}\hline
a_{11} & a_{12} \\
a_{21} & a_{22} \\\hline
\end{array} & \mathbf{0} & \mathbf{0} \\

\mathbf{0} & & \\

\begin{array}{|ccc|}\hline
b_{11} & b_{12} & b_{13} \\
b_{21} & b_{22} & b_{23} \\
b_{31} & b_{32} & b_{33} \\\hline
\end{array} & & \\

\mathbf{0} & & \\

\mathbf{0} & & \mathbf{0} \\

\begin{array}{|cc|}\hline
c_{11} & c_{12} \\
c_{21} & c_{22} \\\hline
\end{array}

\end{array}\right)

```

$$\left(\begin{array}{ccc|ccc|cc} a_{11} & a_{12} & & & & & & \\ a_{21} & a_{22} & & & & & & \\ & & b_{11} & b_{12} & b_{13} & & & \\ & \mathbf{0} & b_{21} & b_{22} & b_{23} & & \mathbf{0} & \\ & & b_{31} & b_{32} & b_{33} & & & \\ & \mathbf{0} & & \mathbf{0} & & c_{11} & c_{12} & \\ & & & & & c_{21} & c_{22} & \end{array} \right)$$

2.8 注音和标注

有的时候，我们会需要在公式的上面或者下面打上一些符号，这时就需要使用注音符号或者是标注符号，表 2.11 列出了一些数学注音符号，表 2.12 列出了一些长的标注符号。

表 2.11: 数学注音符号

命令	符号	命令	符号	命令	符号
<code>\bar{x}</code>	\bar{x}	<code>\acute{x}</code>	\acute{x}	<code>\check{x}</code>	\check{x}
<code>\grave{x}</code>	\grave{x}	<code>\mathring{x}</code>	\mathring{x}	<code>\vec{x}</code>	\vec{x}
<code>\dot{x}</code>	\dot{x}	<code>\ddot{x}</code>	\ddot{x}	<code>\dddot{x}</code>	\dddot{x}
<code>\hat{x}</code>	\hat{x}	<code>\tilde{x}</code>	\tilde{x}	<code>\breve{x}</code>	\breve{x}
<code>\widehat{x}</code>	\widehat{x}	<code>\widetilde{x}</code>	\widetilde{x}		

表 2.12: 长标注符号

命令	符号	命令	符号
<code>\overline{xxx}</code>	\overline{xxx}	<code>\overleftrightarrow{xxx}</code>	\overleftrightarrow{xxx}
<code>\underline{xxx}</code>	\underline{xxx}	<code>\underleftrightarrow{xxx}</code>	$\underleftrightarrow{xxx}$
<code>\overleftarrow{xxx}</code>	\overleftarrow{xxx}	<code>\overbrace{xxx}</code>	\overbrace{xxx}
<code>\underleftarrow{xxx}</code>	\underleftarrow{xxx}	<code>\underbrace{xxx}</code>	\underbrace{xxx}
<code>\overrightarrow{xxx}</code>	\overrightarrow{xxx}	<code>\widehat{xxx}</code>	\widehat{xxx}
<code>\underrightarrow{xxx}</code>	\underrightarrow{xxx}	<code>\widetilde{xxx}</code>	\widetilde{xxx}

2.8.1 上划线和下划线

在上面的表 2.12 中已经给出了上划线和下划线的命令，这里举两个例子让大家有一个直观的认识。

上划线和下划线

命令：`\overline{z_1+z_2} = \overline{z_1}+\overline{z_2}`

`\underline{a^2 + b^2 = c^2}`

效果： $\overline{z_1 + z_2} = \overline{z_1} + \overline{z_2}$

$\underline{a^2 + b^2 = c^2}$

2.8.2 弧的符号

很多时候可能会需要输出类似于 \widehat{AB} 这样表示弧的符号，上弧使用 `\overparen{AB}` 这条命令得到，相对应的下弧则使用 `\underparen{AB}` 这条命令得到，即把弧画到了字母的下方，如图 2.3 所示。



图 2.3: 弧的符号

上弧还可以使用 `\overset{\frown}{AB}` 这条命令得到，显示的效果是： $\overset{\frown}{AB}$ ，两种输入方式得到的结果有细微的差异，可以根据自己的喜好选择使用。

2.8.3 符号上下方的标注

如果要在数字或符号的上下方进行标注，可以使用 `\overset{ }{ }` 和 `\underset{ }{ }` 这两条命令，其作用是将前一括号中的内容置于后一括号的上方或下方。

符号上下方的标注

`u \overset{?}{+} v \underset{1}{\thicksim} w \overset{2}{\thicksim} z`

`f(x) \overset{\text{def}}{=} x^2 - 1`

$$u \overset{?}{+}_1 v \underset{1}{\thicksim} w \overset{2}{\thicksim} z \quad f(x) \overset{\text{def}}{=} x^2 - 1$$

`\thicksim` 命令是画波浪符号 \sim ，此符号不能用键盘上感叹号前面的那个键输入。

2.8.4 向量的输入

向量可以通过在一个变量上方添加小箭头来指定, 用 `\vec{ }` 命令来输入即可, 如果要表示一个从 A 到 B 的向量, 则需要用到 `\overleftarrow{ }` 和 `\overrightarrow{ }` 这两条命令。

向量

命令: `\vec{a}` `\overleftarrow{BA}` `\overrightarrow{AB}`

效果: \vec{a} \overleftarrow{BA} \overrightarrow{AB}

2.8.5 带上下括号的公式

有时我们还会遇到一些公式, 需要在公式的上方或下方用大括号对其进行注释说明, 这类公式虽然遇到的不多, 但是也非常实用, 我们看下面的例子:

带上下括号的公式

命令: `a + {\overbrace{b + \cdots}}^{126} + z`

`\overset{m}{\overbrace{a + \underset{n}{\underbrace{(b + c)}}}}`

`\underbrace{a + \overbrace{b + \cdots + y}^{=t} + z}_{\text{total}}`

效果: $a + \overbrace{b + \cdots}^{126} + z$ $\overset{m}{\overbrace{a + \underset{n}{\underbrace{(b + c)}}}}$ $\underbrace{a + \overbrace{b + \cdots + y}^{=t} + z}_{\text{total}}$

也许有人会问, 类似于图 1.6 中那样上下括号的位置是交错的公式又要怎样输入呢, 这个问题在第 2.12 节中将会给出解答。

2.9 常用符号

除了前面讲到的那些符号以外, 还有一些符号也是比较常用的, 在表 2.13 中列举了一些常见的几何符号、在表 2.14 中列举了一些常用的特殊符号。为了在查找的时候更方便一点, 有个别符号会出现重复的情况。

2.9.1 几何符号

在 GeoGebra 中只能输入很少一部分的几何符号, 像平行四边形、平行且等于 (虽然可以使用命令 `\overset{\text{//}}{=}` 变通得到, 但非常的不美观) 等一些常用的符号都没有提供。

表 2.13: 几何符号

命令	符号	命令	符号	命令	符号
<code>\because</code>	\because	<code>\therefore</code>	\therefore	<code>\perp</code>	\perp
<code>\triangle</code>	\triangle	<code>\triangledown</code>	∇	<code>\vartriangle</code>	\triangleleft
<code>\nabla</code>	∇	<code>\blacktriangledown</code>	\blacktriangledown	<code>\blacktriangle</code>	\blacktriangle
<code>\Delta</code>	Δ	<code>\triangleleft</code>	\triangleleft	<code>\triangleright</code>	\triangleright
<code>\Diamond</code>	\diamond	<code>\lozenge</code>	\lozenge	<code>\blacklozenge</code>	\blacklozenge
<code>\Box</code>	\square	<code>\square</code>	\square	<code>\blacksquare</code>	\blacksquare
<code>\angle</code>	\angle	<code>\measuredangle</code>	\measuredangle	<code>\sphericalangle</code>	\sphericalangle
<code>\parallel</code>	\parallel	<code>\nparallel</code>	\nparallel	<code>\slash</code>	\parallel
<code>\degree</code>	$^\circ$	<code>\odot</code>	\odot	<code>\bigodot</code>	\odot



相同的符号可能会有几种输入方式，不同命令得到的显示效果，有的会有细小差异，有的则完全相同，比如要得到度这个符号，`180\degree` 和 `180^{\circ}` 这两个命令都会得到 180° 这一结果。

2.9.2 特殊符号

前文没有提到的一些有可能会用到的符号统一归类到 表 2.14 的特殊符号中。

表 2.14: 特殊符号

命令	符号	命令	符号	命令	符号
<code>\infty</code>	∞	<code>\propto</code>	\propto	<code>\colon</code>	$:$
<code>\forall</code>	\forall	<code>\exists</code>	\exists	<code>\nexists</code>	\nexists
<code>\wedge</code>	\wedge	<code>\vee</code>	\vee	<code>\lnot</code> 或 <code>\neg</code>	\neg
<code>\spadesuit</code>	\spadesuit	<code>\heartsuit</code>	\heartsuit	<code>\clubsuit</code>	\clubsuit
<code>\diamondsuit</code>	\diamondsuit	<code>\Game</code>	\Game	<code>\P</code>	\P
<code>\flat</code>	\flat	<code>\natural</code>	\natural	<code>\sharp</code>	\sharp
<code>\circledS</code>	\circledS	<code>\copyright</code>	\copyright	<code>\textregistered</code>	\textregistered
<code>\S</code>	\S	<code>\celsius</code>	$^\circ\text{C}$	<code>\textperthousand</code>	\textperthousand
<code>\star</code>	\star	<code>\bigstar</code>	\bigstar	<code>\ast</code>	$*$
<code>\bullet</code>	\bullet	<code>\mid</code>	\mid	<code>\nmid</code>	\nmid
<code>\surd</code>	\surd	<code>\checkmark</code>	\checkmark	<code>\multimap</code>	\multimap
<code>\o</code>	\o	<code>\empty</code> 或 <code>\O</code>	\emptyset	<code>\varnothing</code>	\varnothing
<code>\boxplus</code>	\boxplus	<code>\boxminus</code>	\boxminus	<code>\boxtimes</code>	\boxtimes
<code>\boxdot</code>	\boxdot	<code>\boxbox</code>	\boxbox		

2.9.3 特殊字母

有一些特殊写法的字母并不是通过改变字体得到的，而是有专门的命令。

表 2.15: 特殊字母

命令	符号	命令	符号	命令	符号
<code>\hbar</code>	\hbar	<code>\hslash</code>	\hslash	<code>\imath</code>	\imath
<code>\jmath</code>	\jmath	<code>\ell</code>	ℓ	<code>\ss</code>	\ss
<code>\mho</code>	\mho	<code>\eth</code>	\eth	<code>\partial</code>	∂
<code>\Re</code>	\Re	<code>\Im</code>	\Im	<code>\wp</code>	\wp
<code>\aleph</code>	\aleph	<code>\ae</code>	\ae	<code>\AE</code>	\AE
<code>\Bbbk</code>	\Bbbk	<code>\Finv</code>	\Finv		

2.10 标准数学函数

数学函数一般用直立的 **Roman** 体排版印刷，而普通字母一般用 **Italic** 字体，所以在输入函数的时候需要在函数名的前面加上 `\` 转义符，请参照 表 2.16。

表 2.16: 标准数学函数

命令	函数	命令	函数	命令	函数	命令	函数
<code>\arccos</code>	\arccos	<code>\arcsin</code>	\arcsin	<code>\arctan</code>	\arctan	<code>\arg</code>	\arg
<code>\cos</code>	\cos	<code>\cosh</code>	\cosh	<code>\cot</code>	\cot	<code>\coth</code>	\coth
<code>\csc</code>	\csc	<code>\deg</code>	\deg	<code>\det</code>	\det	<code>\dim</code>	\dim
<code>\exp</code>	\exp	<code>\gcd</code>	\gcd	<code>\hom</code>	\hom	<code>\inf</code>	\inf
<code>\ker</code>	\ker	<code>\lg</code>	\lg	<code>\lim</code>	\lim	<code>\liminf</code>	\liminf
<code>\limsup</code>	\limsup	<code>\ln</code>	\ln	<code>\log</code>	\log	<code>\max</code>	\max
<code>\min</code>	\min	<code>\Pr</code>	\Pr	<code>\sec</code>	\sec	<code>\sin</code>	\sin
<code>\sinh</code>	\sinh	<code>\sup</code>	\sup	<code>\tan</code>	\tan	<code>\tanh</code>	\tanh
<code>\mod</code>	\mod	<code>\pmod</code>	(\mod)				

2.11 简单表格

在 GeoGebra 中绘制表格最简单的办法是利用 GeoGebra 的表格区进行设计，选中需要的部分之后“右击－创建－表格”，这时就会把表格创建在绘图区；或者在指令栏中直接使用指令：**表格文本 (TableText)** 也能制作表格。

当然也可以使用 \LaTeX 代码来绘制一些简单的表格，`tabular` 环境提供了最简单的表格功能，它用 `\hline` 命令来表示横线，`|` 表示竖线，用 `&` 来分列，用 `\\` 来换行；每列可以采用居左、居中、居右等对齐方式，分别用字母 `l`、`c`、`r` 来表示。

简单表格

```
\begin{tabular}{|l|c|r|}\hline
    居左 & 居中 & 居右 \\ \hline
    GGb & GGb & GGb \\ \hline
    GeoGebra & GeoGebra & GeoGebra \\ \hline
    TeX & TeX & TeX \\ \hline
    LaTeX & LaTeX & LaTeX \\ \hline
\end{tabular}
```

居左	居中	居右
GGb	GGb	GGb
GeoGebra	GeoGebra	GeoGebra
TeX	TeX	TeX
\LaTeX	\LaTeX	\LaTeX

在 `GeoGebra` 中用 \LaTeX 代码来绘制表格相对来说比较繁琐，但是软件本身提供了这个功能，所以这里也举了一个简单的例子。

2.12 虚位与重叠

2.12.1 虚位的应用

当使用上标 `^` 和下标 `_` 命令时， \LaTeX 对文本的垂直对齐有时显得太过于“自作多情”，而有的时候我们需要让上下标占据一个虚位 (phantom)^①，使用 `` 命令可以给不显示在输出结果中的字符保留位置，具体的看下面例子。

① 有的书籍也把虚位命令称作幻影命令。

虚位在上下标中的应用

命令: $\hat{12}_6S$ $\hat{12}_{6}S$ Γ_{ij}^k Γ_{ij}^{k} Γ_{k}^{k} 效果: $\hat{12}_6S$ $\hat{12}_6S$ Γ_{ij}^k Γ_{ij}^k Γ_{ij}^k 

注意对比上标和下标的位置及空白间距的大小。

还有一个更为简单的方法: $\{X_{12}\}^6$ $\{X^6\}_{12}$, 得到: X_{12}^6 X_{12}^6 , 但是这种方法只能把上标或下标作为一个整体来处理, 灵活性不如使用虚位命令。

2.12.2 利用虚位生成空格

与 $\backslash hspace{\text{长度}}$ 命令相似, 也可以使用 $\backslash phantom{\text{虚位文字}}$ 命令来生成一段空格, 要注意如果虚位文字中含有空格, 则空格也会计算在内。

用虚位生成空格

命令: I love GGb

 $I \phantom{\text{love}} \text{GGb}$ $I \underline{\phantom{\text{love}}} \text{GGb}$

效果: I love GGb

 $I \phantom{\text{love}} \text{GGb}$ $I \underline{\phantom{\text{love}}} \text{GGb}$

上面例子中第三条命令中 **love** 前后的空格出现在不同位置会影响到下划线的长度。

2.12.3 递等式的对齐

上文提到使用 **align** 环境可以排版需要对齐的公式, 如果式子的每一行都有对齐的符号, 则直接书写就行; 但是像递等式的对齐, 就需要配合虚位 $\backslash phantom$ 命令来进行排版, 下面的例子中第一行相当于虚拟了一个不显示出来的等号 (=), 然后用 **&** 符号来标识对齐的位置。

递等式的对齐

```
\begin{align}
& \phantom{=}(a + b)(a^2 - ab + b^2) \\
& = a^3 - a^2b + ab^2 + a^2b - ab^2 + b^3 \\
& = a^3 + b^3
\end{align}
```

$$\begin{aligned} & (a + b)(a^2 - ab + b^2) \\ & = a^3 - a^2b + ab^2 + a^2b - ab^2 + b^3 \\ & = a^3 + b^3 \end{aligned}$$

2.12.4 水平虚位和垂直虚位

类似虚位命令，还可以使用 `\hphantom{ }` 和 `\vphantom{ }` 这两条命令分别表示水平方向和垂直方向的虚位元素，它们在另一个方向的大小为 0。

水平虚位

虚位命令 `` 和水平虚位命令 `\hphantom{ }` 得到的结果差别非常小，唯一的区别就是使用两条命令后元素的水平位置有细微的差别，可以对比下面两个公式中上标 k 的水平位置（差别很小）。

水平虚位

```
\Gamma_{ij}^{\phantom{ij}k}
\Gamma_{ij}^{\hphantom{ij}k}
```

$$\Gamma_{ij}^k \quad \Gamma_{ij}^k$$

垂直虚位

如果要在垂直方向占用一个虚位，就需要使用 `\vphantom{ }` 这一条命令，例如下面的例子，利用垂直虚位命令可以把集合中的分隔符变长一些，显得更美观。

垂直虚位

$$\left\{x \in \mathbf{R} \mid 0 < |x| < \frac{5}{3}\right\}$$

$$\left\{\vphantom{\frac{5}{3}}x \in \mathbf{R}\right\} \left| 0 < |x| < \frac{5}{3}\right.$$

$$\left\{x \in \mathbf{R} \mid 0 < |x| < \frac{5}{3}\right\} \quad \left\{x \in \mathbf{R} \mid 0 < |x| < \frac{5}{3}\right\}$$

2.12.5 文字重叠

在 \LaTeX 中, 文字都是被装在一个一个的盒子中的, 然后再将盒子按照一定的排版要求排列在一起。根据这一特点, 可以使用命令: $\text{\texttt{\textbackslashllap\{}}}$ 和 $\text{\texttt{\textbackslashrlap\{}}}$ 这两条命令将两部分文字重叠在一起。

文字重叠

正常文字: 文字 $\{\text{\color{red}\texttt{\textbackslashtext}}\}$ 重叠

向左重叠: 文字 $\{\text{\color{red}\texttt{\textbackslashllap\texttt{\textbackslashtext}}}\}$ 重叠

向右重叠: 文字 $\{\text{\color{red}\texttt{\textbackslashrlap\texttt{\textbackslashtext}}}\}$ 重叠

文字 $\text{\color{red}\texttt{\textbackslashtext}}$ 重叠

$\text{\color{red}\texttt{\textbackslashtext}}$ 重叠

文字 $\text{\color{red}\texttt{\textbackslashtext}}$ 重叠

2.12.6 交错括号的排版

利用文字重叠命令、虚位命令和上下括号命令, 可以得到上下括号有交替的公式。

交替括号

$$a + \text{\texttt{\textbackslashrlap\{\overbrace{}^m\}}} b + \text{\texttt{\textbackslashunderbrace\{c + d + e\}_n}} + f$$

$$a + \overbrace{b + c + d}^m + \underbrace{e + f}_n$$

这条命令, 先用 $\text{\texttt{\textbackslashoverbrace\{}^m\}}$ 命令生成一个长度为 $b + c + d$ 的上括号, 然后再用 $\text{\texttt{\textbackslashrlap}}$ 命令将这个上括号与右侧的文字重叠在一起, 最后在 $c + d + e$ 的下方用 $\text{\texttt{\textbackslashunderbrace\{c + d + e\}_n}}$ 命令生成一个下括号。

第三章 简单化学方程式

在 \LaTeX 中实现简单化学方程式的输入主要是依靠 `mhchem` 宏包完成的，而有机化学等一些更复杂的公式主要是使用 `chemfig`、`PPCHTEX` 等宏包来完成，`chemfig` 的使用比较复杂，学习会很痛苦，得到的效果也未必尽如人意，因此遇到太复杂的结构式，先利用化学绘图软件绘制好后保存成图片，然后再把图片插入到文档中可能会是一个更有效的方法。

GeoGebra 中集成了 `mhchem` 宏包^①，可以输入一些简单的化学方程式，只需要在“文本”工具中直接使用 `\ce{ }` 命令即可，使用 `\ce` 命令输入上下标的方法与前面讲的数学公式中的上下标输入方法有一些差异。

3.1 无机物

3.1.1 整数下标、电荷数、氧化数

单个下标、电荷数或者氧化数可以直接写在方程式后面，如果上下标的数量不止一个，则需要使用上尖号 \wedge 标识：

上下标的表示

<code>\ce{CO2}</code>	<code>\ce{H2O}</code>	<code>\ce{H2SO4}</code>
<code>\ce{H+}</code>	<code>\ce{Cl-}</code>	<code>\ce{NH4+}</code>
<code>\ce{Cu^{2+}}</code>	<code>\ce{Fe^{II}}</code>	<code>\ce{Pb^{IV}}</code>
CO ₂	H ₂ O	H ₂ SO ₄
H ⁺	Cl ⁻	NH ₄ ⁺
Cu ²⁺	Fe ^{II}	Pb ^{IV}

^① `mhchem` 宏包的文档可以在 <https://mirrors.nju.edu.cn/CTAN/macros/latex/contrib/mhchem/mhchem.pdf> 这儿得到，主要看 20 页以前的内容。

3.1.2 分数下标

标准分数的表示方法和数学公式中一样，使用 `\frac{ }{ }` 命令，并且需要使用一对花括号 `{ }` 将分数括起来，表示是一个整体；分子式前面的分数可以简写。

`\ce{Fe(CN)_{\frac{6}{2}}}` $\text{Fe(CN)}_{\frac{6}{2}}$

`\ce{1/2 H2O}` $\frac{1}{2} \text{H}_2\text{O}$

`\ce{(1/2) H2O}` $(1/2) \text{H}_2\text{O}$

3.1.3 几种括号

圆括号和方括号直接使用键盘输入即可，花括号需要加入转义符 `\` 才能正常显示，括号支持嵌套使用。

几种括号

圆括号：`\ce{(NH4)2S}`

方括号：`\ce{[AgCl2]-}`

花括号：`\ce{[\{(X2)3\}2]^3+}`

效果： $(\text{NH}_4)_2\text{S}$ $[\text{AgCl}_2]^-$ $[\{(\text{X}_2)_3\}_2]^{3+}$

3.1.4 上标与下标：同位素、分子、离子

上标与下标混合使用

`\ce{^{227}_{90}Th+}` `\ce{O''_{i,x}}`

`\ce{M^{..}_i}` `\ce{V^{4'}_{Ti}}`

`\ce{V_{V,1}C_{C,0.8}V_{C,0.2}}`

效果： $^{227}_{90}\text{Th}^+$ $\text{O}''_{i,x}$ $\text{M}_i^{\bullet\bullet}$ $\text{V}_{\text{Ti}}^{4'}$ $\text{V}_{\text{V},1}\text{C}_{\text{C},0.8}\text{V}_{\text{C},0.2}$

3.1.5 圆点：自由基、水合物

自由基和水合物中的圆点可以用 `*` 表示，要注意的是当圆点在分子式前面的时候，需要在 `*` 号前加一对花括号。

- 自由基

$\text{\ce{Cl*}}$ $\text{Cl}\cdot$ $\text{\ce{{}*CH3}}$ $\cdot\text{CH}_3$

- 水合物

 $\text{\ce{KCr(SO4)2*12H2O}}$ $\text{KCr(SO}_4)_2 \cdot 12 \text{H}_2\text{O}$ $\text{\ce{KCr(SO4)2 * 12H2O}}$ $\text{KCr(SO}_4)_2 \cdot 12 \text{H}_2\text{O}$

3.2 有机物

在 GeoGebra 中只能输入很少的一部分有机化学的符号、标识等，更多的需求请使用专业化学绘图软件。

3.2.1 化学键的表示

单键用减号“-”表示，双键用等号“=”表示，三键用井号“#”表示，配位键用减号和大于小于号“->”“-<”表示，使用 $\text{\bond{}}$ 命令可以得到更多形式的化学键。

单键

 $\text{\ce{CH3-CHO}}$ $\text{CH}_3\text{--CHO}$

双键

 $\text{\ce{CH2=CH2}}$ $\text{CH}_2=\text{CH}_2$

三键

 $\text{\ce{CH\#CH}}$ $\text{CH}\equiv\text{CH}$

配位键

 $\text{\ce{A -> B <- C}}$ $\text{A} \longrightarrow \text{B} \longleftarrow \text{C}$ $\text{\ce{A \bond{->} B \bond{<-} C}}$ $\text{A} \rightarrow \text{B} \leftarrow \text{C}$

更多的化学键

$\text{\ce{X=Y\#Z}}$	$X=Y\equiv Z$
$\text{\ce{A-B=C\#D}}$	$A-B=C\equiv D$
$\text{\ce{A\bond{-}B\bond{=}C\bond{\#}D}}$	$A-B=C\equiv D$
$\text{\ce{A\bond{1}B\bond{2}C\bond{3}D}}$	$A-B=C\equiv D$
$\text{\ce{A\bond{\sim}B\bond{\sim}C}}$	$A\cdots B\cdots C$
$\text{\ce{A\bond{\dots}B\bond{\dots}C}}$	$A\cdots B\cdots C$
$\text{\ce{A\bond{\sim}B\bond{\sim=}C}}$	$A\equiv B\equiv C$
$\text{\ce{A\bond{\sim\sim}B\bond{\sim=}C\bond{\sim\sim}D}}$	$A\equiv B\equiv C\equiv D$
$\text{\ce{A\bond{\sim\sim}B\bond{\sim\sim}C\bond{\sim\sim}D}}$	$A\equiv B\equiv C\equiv D$

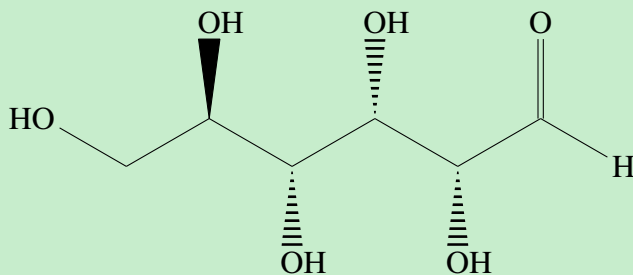
3.2.2 斜体：*cis* & *trans*

有机化学中用来表示顺反异构的“*cis*”和“*trans*”通常都写成斜体，我们知道数学公式中的字母会被写成斜体，所以可以用一对 \$ 符号将要写成斜体的字母进行转义。

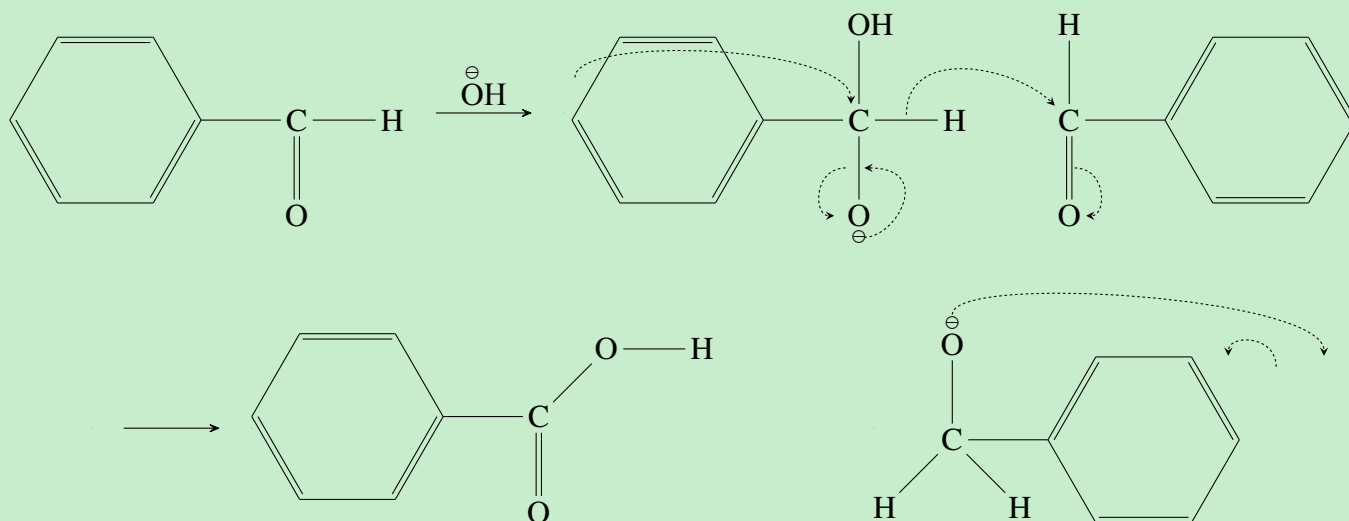
$\text{\ce{\$cis\$-CH3CH=CHCH3}}$	<i>cis</i> -CH ₃ CH = CHCH ₃
-----------------------------------	----------------------------------------------------

3.2.3 chemfig 宏包示例

有机化合物的表示方法除了结构简式，有时还会需要在文章中插入有机物的结构式或者键线式，在 L^AT_EX 中引入 chemfig 宏包后可以得到下面的结构式，而 GeoGebra 中没有引入 chemfig 宏包，所以无法处理。



有时可能还需要表示一下电子的转移，这也需要使用 chemfig 宏包才能达到效果：



3.3 箭头

化学反应中的正向、逆向、可逆等不同类型的箭头有专门的命令：

3.3.1 反应方向

`\ce{A -> B}`

$A \longrightarrow B$

`\ce{A <- B}`

$A \longleftarrow B$

`\ce{A <=> B}`

$A \rightleftharpoons B$

`\ce{A <=>> B}`

$A \rightleftharpoons B$

`\ce{A <<=> B}`

$A \leftrightharpoons B$

3.3.2 添加反应条件

箭头上添加反应条件

在箭头命令后面紧跟一个或两个方括号 “[] ” 来表示反应条件，注意箭头和方括号之间不能有空格，一个括号时，反应条件在箭头上方，两个括号时，反应条件一个在箭头上方，一个在箭头下方。

`\ce{2H2 + O2 ->[\Delta] 2H2O}`

$2\text{H}_2 + \text{O}_2 \xrightarrow{\Delta} 2\text{H}_2\text{O}$

`\ce{2H2 + O2 ->[燃烧] 2H2O}`

$2\text{H}_2 + \text{O}_2 \xrightarrow{\text{燃烧}} 2\text{H}_2\text{O}$

`\ce{A ->[H2O] B}`

$A \xrightarrow{\text{H}_2\text{O}} B$

`\ce{A ->[x][x_i] B}`

$A \xrightarrow[x_i]{x} B$

`\ce{N2+3H2<=>[高温加压][催化剂]2NH3}`

$\text{N}_2 + 3\text{H}_2 \xrightleftharpoons[\text{催化剂}]{\text{高温加压}} 2\text{NH}_3$

等号上添加反应条件

在化学反应中，mhchem 宏包没有提供一种“等号”让我们连接化学方程式的两端，如果仅仅需要使用最基本的等号，可以直接用键盘上的“=”来表示，要添加反应条件时，则需要使用 \xlongequal 命令。

长等号与反应条件

`\ce{A2 + B2 \xlongequal {1000\celsius} 2AB}`

`\ce{2NaHCO3 \xlongequal {heating} Na2CO3 + H2O + CO2 ^}`

$$A_2 + B_2 \xlongequal{1000^\circ\text{C}} 2AB$$

$$2\text{NaHCO}_3 \xlongequal{\text{heating}} \text{Na}_2\text{CO}_3 + \text{H}_2\text{O} + \text{CO}_2 \uparrow$$



如果要把反应条件写在等号的下方，则把花括号换成方括号即可，如果等号上方和下方都有反应条件，则方括号要写在花括号的前面，有顺序要求。

3.3.3 沉淀与气体

化学反应中会有沉淀与气体的表示，用上尖号 ^ 和 v 字母或 \uparrow 和 \downarrow 命令来表示，上尖号 ^ 和 v 字母与前面的字符之间要有空格。

`\ce{A v B (v) -> B ^ B (^)}`

$$A \downarrow B \downarrow \longrightarrow B \uparrow B \uparrow$$

`\ce{Fe + 2H+ = H2 ^ + Fe^2+}`

$$\text{Fe} + 2\text{H}^+ = \text{H}_2 \uparrow + \text{Fe}^{2+}$$

`\ce{Ca(OH)2 + CO2 = CaCO3 v + H2O}`

$$\text{Ca}(\text{OH})_2 + \text{CO}_2 = \text{CaCO}_3 \downarrow + \text{H}_2\text{O}$$

3.4 热化学方程式：能量变化、分数计量数

热化学方程式需要显示能量变化和分数计量表，这个时候，需要在化学式和能变之间使用空格命令 \quad 分开：

热化学方程式

`\ce{N2(g) + 3H2(g) -> 2NH3(g)} \quad \Delta H_{\rm f}^\circ = \{-92.5\}\text{kJ}`

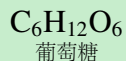
$$\text{N}_2(\text{g}) + 3\text{H}_2(\text{g}) \longrightarrow 2\text{NH}_3(\text{g}) \quad \Delta H_{\text{f}}^\circ = -92.5\text{kJ}$$

3.5 其他内容

3.5.1 在文字下方添加标注

同素异形体和同分异构体的存在使我们有时候需要在化学式的下方添加说明，此时可以使用 `\underset{ }{ }` 命令来实现。

`\underset{葡萄糖}{\ce{C6H12O6}}`



3.5.2 多行反应方程式

连续的多行反应方程式，需要引入 `align` 环境，每一行的开始用 `&` 标识对齐，结尾使用换行命令 `\\`：

`\begin{align*}`

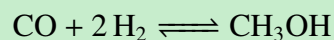
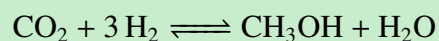
`& \ce{CO2 + 3H2 <=> CH3OH + H2O} \\`

`& \ce{CO2 + H2 <=> CO + H2O} \\`

`& \ce{CO + 2H2 <=> CH3OH} \\`

`& \ce{CH3OH <=> CH3OCH3 + H2O}`

`\end{align*}`



环境名后面加上 `*` 号是表示不用给公式编号。

附录 A $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 简介

A.1 $\text{T}_{\text{E}}\text{X}$ 家族

A.1.1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 简介

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 是一种面向数学和其它科技文档的电子排版系统，我们通常所说的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 是一个总称，包括 $\text{T}_{\text{E}}\text{X}$ 、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 等，它们是免费软件，包括源程序也是免费的。

目前为止， $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 几乎在所有的计算机操作系统平台上得到实现，用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编辑的源文件可在不同的平台之间自由交换，而得到的输出是完全相同的。

在文本模式下， $\text{T}_{\text{E}}\text{X}$ 和 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 应该写作：“ TeX 和 LaTeX ”，在 GeoGebra 中可以使用命令： $\backslash\text{LaTeX}$ 得到 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 。

个人认为，要学好任何一门学科都应该学习其历史，这样才能更好的对所学知识加深理解，并有利于把所学知识融会贯通，所以在这里非常简单的介绍一点点 $\text{T}_{\text{E}}\text{X}$ 方面的知识，如果要了解更多的内容请自行搜索，网上有非常详细的介绍，包括其作者的一些传奇经历，这里所收集整理知识也主要来源于网络。

A.1.2 引擎： $\text{T}_{\text{E}}\text{X}$

1976 年，Donald Ervin Knuth (1938 –)^①为了排版自己的著作《The Art of Computer Programming》(计算机程序设计艺术)而开发的一个全新的排版引擎，这就是 $\text{T}_{\text{E}}\text{X}$ ，从这几个交错起伏的字母中便道出了“排版”二字的几分意味：精确、复杂、注重细节和品位，这里所说的引擎，是指能够实现断行、分页等操作的程序。

$\text{T}_{\text{E}}\text{X}$ 是一个相当稳定而几乎没有 bug 的系统，它的版本号趋近于 π ，每发布一个修正版，版本号就增加一位小数，当前版本是 2014 年 1 月更新的 3.14159265，在 $\text{T}_{\text{E}}\text{X}$ 四十多年的历史

^① 中文名叫：高德纳，1960 年凯斯工学院数学学士，因为长得帅同时获赠硕士，1963 年加州理工数学博士，同年留校任教。1968 年跳槽到斯坦福，1974 年获图灵奖，1992 年退休，1995 年获冯·诺依曼奖。

中，这仅仅是第九次更新，上次更新在 2008 年。

2004 年在 $\text{T}_\text{E}\text{X}$ 的基础上出现了一个新的引擎 $\text{X}_\text{Y}\text{T}_\text{E}\text{X}$ ，该引擎对中日韩等东亚字体的支持非常友好，现在是处理中文文档的首选，注意在使用 $\text{X}_\text{Y}\text{T}_\text{E}\text{X}$ 引擎时，字符编码请选择 UTF-8。

A.1.3 格式： $\text{L}_\text{A}\text{T}_\text{E}\text{X}$

基本的 $\text{T}_\text{E}\text{X}$ 系统只有 300 多个元命令 (primitive)，十分精悍，但很难读懂，只适于非正常人类，所以 Knuth 提供了一种格式 (format，宏命令的集合) 对 $\text{T}_\text{E}\text{X}$ 进行了封装，这就是 Plain $\text{T}_\text{E}\text{X}$ ，包含 600 多个宏命令，然而它还是不够高级。

1980 年代初期，斯坦福研究所 (SRI) 的 Leslie Lamport (1941 –)^①开发了一种新的格式，也就是 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ ，实际上 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 是基于 plain $\text{T}_\text{E}\text{X}$ 定义的格式，它定义了许多新的控制命令并封装成一个可执行文件，这个可执行文件会把新定义的命令解释成为 $\text{T}_\text{E}\text{X}$ 的控制命令，并最终交由 $\text{T}_\text{E}\text{X}$ 引擎进行排版。1994 年在 2.09 版本的基础上发布了 $\text{L}_\text{A}\text{T}_\text{E}\text{X} 2_\epsilon$ ，这也是我们现在使用得最多的版本， $\text{L}_\text{A}\text{T}_\text{E}\text{X} 3$ 的开发也在进行中，虽然正式版看起来遥遥无期，但 $\text{L}_\text{A}\text{T}_\text{E}\text{X} 3$ 已经可以使用了。

A.1.4 宏包

$\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 出现之后，在它的基础上出现了很多宏包 (package)，这些宏包丰富和加强了 $\text{T}_\text{E}\text{X}$ 的功能，比如提供数学功能的 `amsmath` 宏包，制作幻灯片的 `beamer` 宏包，用于绘图的 `tikz` 宏包等等。

使用 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的排版时，许多功能需要在程序的开始部分 (导言区) 引入用到的宏包，比如中文支持需要 `ctex` 宏包，如数学字体 $\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{X}\mathcal{Y}\mathcal{Z}$ 需要 `mathrsfs` 宏包，在 GeoGebra 中已经作过相应处理，不需要引用宏包，可以直接使用相应的命令。

A.1.5 驱动

Knuth 最初设计的 $\text{T}_\text{E}\text{X}$ 只能用于施乐图形打印机，后来有人将其输出改为设备无关的格式，也就是 DVI，发展到现在已经有了更加丰富的输出格式，用户需要用驱动程序把它转换为自己想要的格式，比如 `XeLaTeX` 命令可以将源文件编译成 PDF 格式的文本。

^① 1960 年 MIT 数学学士，布兰迪斯大学数学系 1963 年硕士，1972 年博士。1970 年加入麻省计算机同伙公司，1977 年跳槽到 SRI，1985 年跳到 DEC，2001 年跳到微软。2008 年获冯·诺依曼奖。

A.1.6 小结

数字排版有四个重要环节：标记语言、页面描述语言、光栅图像处理器、输出设备。 \TeX 是最精确、最高级的面向专业排版的标记语言， \TeX 家族可以划分为四个层次：引擎、格式、宏包、驱动。

我个人比较喜欢选择 \XeTeX 引擎和 \LaTeX 格式^①。

A.2 优点和缺点

\TeX 本身的领域是专业排版（即方正书版、InDesign 的领域），但现在 \TeX/\LaTeX 也被广泛用于生成电子文档甚至幻灯片等， \TeX 语言的数学部分偶尔也在其他一些地方使用（如 GeoGebra 中 $\wedge_^$ ），但是要注意 \TeX 并不适用于文书处理（MS Office 的领域），不过很多字处理软件中的“公式编辑”已经支持 \LaTeX 了。

在众多大侠的不断努力下， \LaTeX 现在同样能运用于物理、化学、乐谱、棋谱等领域的排版（这需要用到不同的宏包）。

\TeX 称为所想即所得的系统，它和 Office 这一类的所见即所得的软件各有优缺点。

A.2.1 选择 \TeX 的理由

- 免费、开源软件；
- 高质量、专业的排版效果；
- 数学公式尤其赏心悦目，是事实上的专业数学排版标准；
- 结构化，它的文档结构清晰；
- 跨平台，几乎可以支持于所有电脑硬件和操作系统平台；

.....

^① 新的引擎 LuaTeX 和格式 ConTeXt 旧版可用，新版正在开发中，离真正可用还需一点时间，故不推荐大家现在使用。

A.2.2 \TeX 的缺陷

相应的, \TeX 由于其工作流程, 设计原则, 资源的缺乏, 以及历史局限性等原因也存在一些缺陷:

- 语法不如 HTML 和 XML 严谨、清晰;
- 制作过程繁琐, 不能直接或实时看到结果;
- 宏包鱼龙混杂, 水准参差不齐, 风格不够统一;
- 相对于商业软件, 用户支持不够好, 文档不完善。

实时显示

关于实时显示这一情况, 在很多高手的不断努力下, 目前得到了很大改善, 其中比较有名的是 LyX 这一免费开源软件, 很多同学喜欢用它来记数学笔记, 值得注意的是, 使用 LyX 时并不要求一定要会 \LaTeX , 直接安装这一软件就可以使用。

GeoGebra 中的 \LaTeX 也支持实时显示, 效果很好, 此外还有一些编辑器也在实时显示方面不断的努力, 进步是巨大的。

A.2.3 不选择 \TeX 的理由

- 需要较多的精力学习;
- 图文混合排版能力弱;
- 仅流行于数学、物理、计算机等领域;
- 中文期刊的支持较差;

.....

A.3 软件准备

初学者面对上述那些引擎、格式、宏包、驱动等概念可能手足无措, 看得一头雾水, 所幸有志愿者把这些东西连同一些实用程序打包集成在一起, 形成了一个发行版。

要学习和使用 \TeX 需要事先在电脑上安装一个发行版^①，如果只是要输入数学公式，那么可以不用安装 \TeX 的发行版，用 GeoGebra 这一软件中的 \LaTeX 来输入公式也基本够用了，并且能实时看到生成的公式。

发行版的基本作用是提供 \TeX 后台处理机制和命令行程序，用户还需要一个前台编辑器来编辑源文件，可以使用普通的文本编辑器，也可以使用专用的 \TeX 编辑器，常见的发行版和编辑器见 表 A.1。

表 A.1: 常见发行版和编辑器

操作系统	发行版	编辑器
Windows	MikTeX	TeXstudio
Linux	teTeX	Kile
Mac OS	MacTeX	TeXShop
通用	TeX Live	TeXworks

这里只举例说明了一个编辑器，网上随便一搜索也能查询到几十个编辑器，各有特色，比如现在非常热门的 Visual Studio Code (VScode) 通过配置也能编译 \LaTeX 代码，大家可自行选择使用。

在 Windows 平台下早期还有一个国人制作的发行版 \CTeX 套装，但是已经几年没有更新，以后估计也不会更新了，不建议使用，这里要和我们排版中文时用到的 `ctex` 宏包区分开来，虽然它们的名字是相同的。

^① 如果要安装软件学习和使用 \LaTeX ，建议安装 TeX Live 发行版和 TeXstudio 编辑器，二者都是跨平台、免费开源的。

第二版跋

此文档从初稿的分享已经过去五年多的时间了，从这些年的反馈中得知，文档起到的作用是微乎其微的，首先是因为作者的水平有限，不能把文档写得生动有趣，让读者产生阅读的兴趣，再者也是因为读者的需求更多的是希望能像“快餐”一样拿来就用，直接得到自己问题的答案，而不是还需要花费精力去学习。

今年全球疫情暴发，许许多多的人都在无私的奉献自己的力量，有的人甚至付出了生命的代价，鲁迅说过：“我们从古以来，就有埋头苦干的人，有拼命硬干的人，有为民请命的人，有舍身求法的人……这就是中国的脊梁。”他们是值得敬佩的人，也是值得学习的人。作者在抗疫工作中没有任何的贡献，反而两次被要求居家自行隔离，在此深表歉意。

暑假的时候增加了许多学习 **GeoGebra** 的老师，但总会有初学者问到怎样修改标签文本的大小、怎样显示根号、怎样排版分段函数之类的问题，于是便萌生了修改此文档的念头，所以在这次修改的时候尽可能的增加了一些例子，帮助大家阅读和理解。

由于作者本人并不是教育工作者，平时的工作和生活也用不到 **GeoGebra** 和 **L^AT_EX** 这一类的软件，写这个文档完全是因为兴趣和爱好，按照作者的主观认识来安排一些需要掌握的知识，所以文档的结构和内容有很多不合理的地方，希望大家能自己甄别使用，如果有大能愿意完善此文档，那是不胜感激的。

第一版跋

首先向一路披荆斩棘看到这里的读者表示祝贺，如果您认真阅读，动手完成了文中所有的例子，并琢磨了其中的意义，相信您已经可以用 GeoGebra 中的 \LaTeX 语言排版出非常漂亮的公式了，从此您也可以算得上半个 \LaTeX er 了，只要坚持下去，您会在数学领域中展现出更多的精彩。不管是在什么行业，要想成为高手，只需要勤学苦练就行了，希望您能在不断的练习中进行创新，发现更多的功能和技巧，并请分享出来。

很多读者没有坚持一直学习 \LaTeX 的原因之一可能是因为 \LaTeX 需要记忆的内容太多，所幸的是，在 GeoGebra 的文本对话框中，有两个下拉列表提供了很多常用命令和符号，只须用鼠标选择就能使用，这大大的方便了大家的使用。

费尽九牛二虎之力熬到本文档完成的时候，才发现自己需要学习的地方也非常之多，之前的一些想法很傻很天真。让我们搜练古今，博采沉奥，耐心等待更好的文档出现。