

# LyteX 中文帮助文档

作者:

Jay [Jaeluh@aol.com](mailto:Jaeluh@aol.com),

ollydbg [ollydbg@ctex.org](mailto:ollydbg@ctex.org)

zoho [zoho@ctex.org](mailto:zoho@ctex.org)

June 19, 2009

Draft Version:1.0

## 目录

<b>1 什么是 Ly<sub>T</sub>E<sub>X</sub></b>	<b>5</b>
<b>2 安装 Ly<sub>T</sub>E<sub>X</sub></b>	<b>5</b>
<b>3 开始使用 Ly<sub>T</sub>E<sub>X</sub></b>	<b>5</b>
3.1 撰写英文文档 . . . . .	5
3.2 撰写简体中文文档 . . . . .	6
3.2.1 创建空白中文文档 . . . . .	7
3.3 导入和导出中文 tex 文档 . . . . .	7
3.3.1 导入中文 tex 文档 . . . . .	7
3.3.2 导出中文 tex 文档 . . . . .	8
<b>4 定制中文模版</b>	<b>9</b>
4.1 中文宏包设置 . . . . .	9
4.2 字体设置 . . . . .	9
4.3 行距和段落设置 . . . . .	10
4.4 浮动图像，表格，目录等的标题中文显示 . . . . .	11
4.5 浮动项的设置 . . . . .	12
4.6 文档类设置 . . . . .	13
4.7 给模版添加环境 (Environment Modules) . . . . .	14
4.8 页面设置 . . . . .	16
4.9 目录和编号 . . . . .	17
4.10 文献 . . . . .	18
4.11 数学宏包的使用 . . . . .	19
4.12 输出 pdf 的设置 . . . . .	20
4.13 列表符号的设置 . . . . .	21
4.14 使用分支 . . . . .	22

<b>5</b>	<b>Lyx 内容排版</b>	<b>22</b>
5.1	快捷键	22
5.2	文本样式排版	23
5.3	文章段落排版	24
5.4	浮动项目排版	25
5.4.1	插入算法	25
5.4.2	插入浮动图片	27
5.4.3	插入浮动表格	28
5.4.3.1	长表格	28
5.4.3.2	长表格的位置调整	28
5.5	Box(盒子)	29
5.5.1	自定义 box	29
5.5.1.1	设置边框的厚度	29
5.5.1.2	设置边框到内容的距离	30
5.5.1.3	设置阴影大小	30
5.5.1.4	设置背景颜色	31
5.5.2	迷你页面 (MiniPage)	31
5.5.3	文本框 (Parbox)	32
5.6	有颜色的 Box	33
5.7	自定义有颜色的 box	34
<b>6</b>	<b>数学公式排版</b>	<b>37</b>
6.1	使用数学工具条	37
6.2	插入行内或者显示公式	37
6.3	公式编号	38
6.3.1	内置公式编号样式	39
6.3.2	按章节或段落对公式进行编号	40
6.3.3	对子公式编号	40
6.3.4	自定义公式编号样式	42
6.4	数学公式的对齐方式	43

6.4.1	AMS align 环境 . . . . .	43
6.4.2	AMS alignat() 环境 . . . . .	44
6.4.3	AMS falign 环境 . . . . .	44
6.4.4	AMS gather 环境 . . . . .	45
6.4.5	ANS Multiline 环境 . . . . .	45
6.4.6	Eqarray 环境 . . . . .	46
6.5	数学长公式输入 . . . . .	46
6.6	长公式中的括号 . . . . .	46
6.7	数学特殊字符的输入 . . . . .	47
6.8	数学脚本的使用 . . . . .	47
7	其他排版技巧	48

# 1 什么是 $LyT_{\text{E}}X$

$LyT_{\text{E}}X$  是一个结合了  $LyX$  而且配置了其他组件的一个移动  $\text{Tex}$  套件, 最新版本的  $LyT_{\text{E}}X$  由以下组件构成:

- $\text{L}^{\text{A}}T_{\text{E}}X$ <sup>1</sup>:  $\text{L}^{\text{A}}T_{\text{E}}X$  ( $\text{L}^{\text{A}}T_{\text{E}}X$ ) 是一种基于  $T_{\text{E}}X$  的排版系统, 由美国计算机学家 Leslie Lamport 在 20 世纪 80 年代初期开发, 利用这种格式, 即使使用者没有排版和程序设计的知识也可以充分发挥由  $T_{\text{E}}X$  所提供的强大功能, 能在几天, 甚至几小时内生成很多具有书籍质量的印刷品。对于生成复杂表格和数学公式, 这一点表现得尤为突出。因此它非常适用于生成高印刷质量的科技和数学类文档。这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档。
- $\text{X}^{\text{e}}T_{\text{E}}X$ <sup>2</sup> (版本 0.999.7):  $\text{X}^{\text{e}}T_{\text{E}}X$  使用 Unicode 和支持像 OpenType 或 Apple Advanced Typography 现代字体技术的  $T_{\text{E}}X$  排版引擎它由 Jonathan Kew 开发和维护, 并以 X11 自由软件许可发布。
- $T_{\text{E}}X$ works (版本 r303): 是一个小巧的语法高亮  $\text{tex}$  文件编辑器。
- $LyX$ <sup>3</sup> (版本 1.63):  $Lyx$  是一个强大的所见即所得的  $\text{tex}$  编辑器, 可以在 windows 以及 linux 平台上运行。 $LyX$  为用户提供了一个类似于 Microsoft word 所见即所得的文件编辑环境, 因此不需要记忆和查找大量的  $\text{tex}$  排版指令了, 特别是对于复杂的数学公式的输入, 用户只需要把精神集中在输入的内容上, 而不是版式上,  $LyX$  不仅可以输出流行的 pdf 文档, 而且也支持 dvi 文件, xml 文件, html 文件, pdf 文件和专业的 ps 文件。

## 2 安装 $LyT_{\text{E}}X$

下载后的  $LyT_{\text{E}}X$  一般是一个自解压缩包, 双击后选择要解压缩的目录, 比如  $D:\backslash\text{lytex}$ , 然后就可以通过双击根目录下的  $\text{lyx.exe}$  来启动可视化编辑环境。

## 3 开始使用 $LyT_{\text{E}}X$

### 3.1 撰写英文文档

步骤如下:

1. 打开  $Lyx$  可视化编辑环境

---

<sup>1</sup>详细介绍请见:<http://zh.wikipedia.org/wiki/Latex>

<sup>2</sup>详细介绍请见:<http://en.wikipedia.org/wiki/Xetex> (英文)

<sup>3</sup>详细介绍请见:<http://zh.wikipedia.org/wiki/LyX>

2. 单击“文件”菜单，选择“从模版新建文件”，然后选择双击 templates 根目录下的任何一个模版文件或者单击菜单新建空白文档

如果还有其他任何问题，可以点击“帮助”菜单，然后选择任何一个你想查看的帮助文档 (由于 Lyx 自带的帮助文档是英文，中文翻译以后会加进去)，由于  $\text{L}^{\text{Y}}\text{X}$  默认的新建文档是英文，所以这里不再赘述有关创建英文文档，关于英文文档撰写和排版的详细介绍，请点击“帮助”▷“用户手册”。

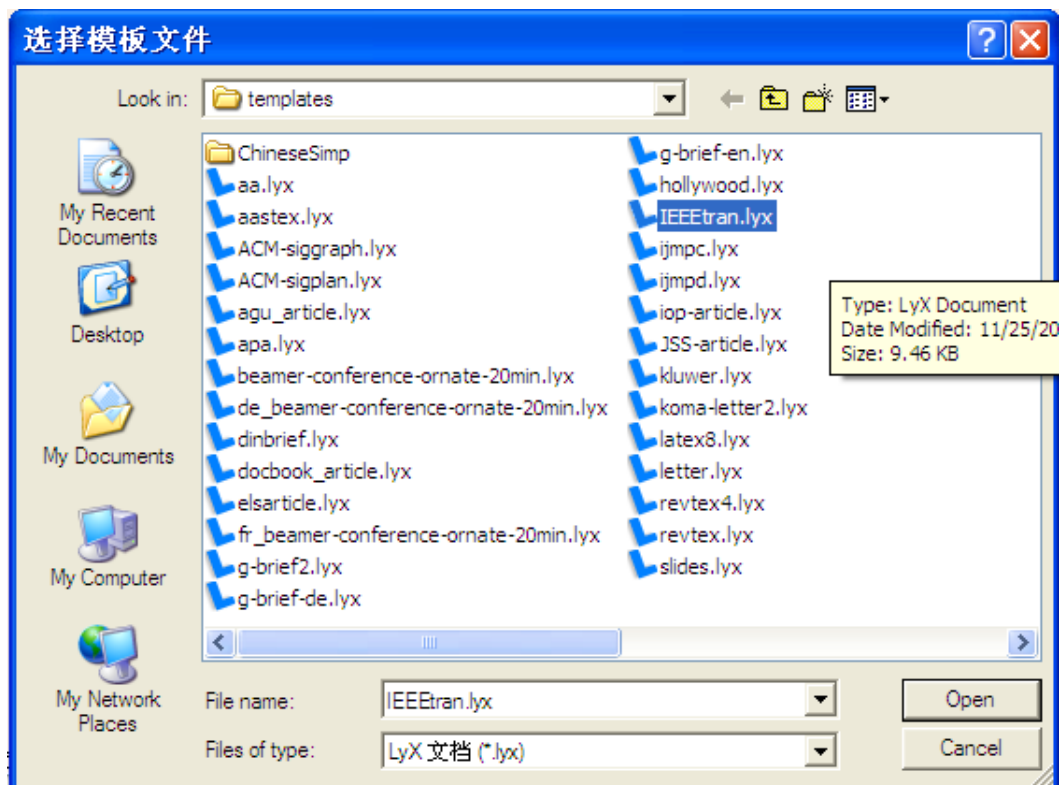


图 1: 选择英文模版

## 3.2 撰写简体中文文档

由于  $\text{L}^{\text{Y}}\text{X}$  对中文支持并不令人满意，所以此部分将花更多的篇幅详细讲解在  $\text{L}^{\text{Y}}\text{X}$  中撰写中文文档，再加上  $\text{L}^{\text{Y}}\text{X}$  是一个类似于 word 的可视化编辑器，因此这里不会详述如何在  $\text{L}^{\text{Y}}\text{X}$  中编写  $\text{T}_{\text{E}}\text{X}$  代码，本部分着重介绍如何在  $\text{L}^{\text{Y}}\text{X}$  中创建文档，定制模版以及一些基本的  $\text{L}^{\text{Y}}\text{X}$  操作技巧，如果你想比较系统的学习  $\text{T}_{\text{E}}\text{X}$ ，可以去阅读一些在线  $\text{T}_{\text{E}}\text{X}$  电子书籍，比如 [LaTeX for Complete Novices](#) (英文)。

### 3.2.1 创建空白中文文档

从现有的中文模版中创建这种方法，也就是最简单的方法，直接从  $\text{L}_{\text{Y}}\text{X}$  的模版中选择：

1. 与创建英文类似，首先要打开  $\text{L}_{\text{Y}}\text{X}$  可视化编辑环境
2. 单击“文件”菜单，选择“从模版新建文件”，然后再打开子文件夹“ChineseSimp”，选择你所需要的简体中文模版<sup>4</sup>

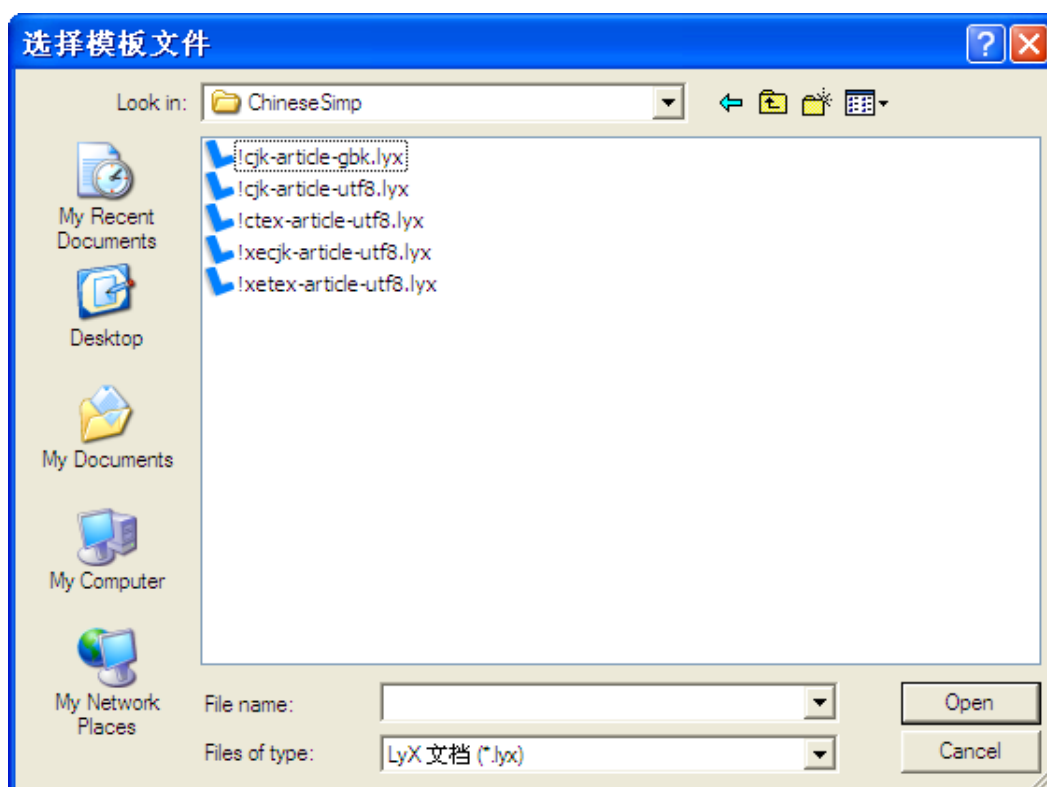


图 2: 从中文模版新建文档

## 3.3 导入和导出中文 $\text{tex}$ 文档

这个小节介绍中文  $\text{tex}$  文档导入和导出时的基本方法。主要目的是避免乱码的出现。

### 3.3.1 导入中文 $\text{tex}$ 文档

如果你要在  $\text{L}_{\text{Y}}\text{X}$  中导入用 CJK 宏包撰写的中文  $\text{tex}$  文档，你可以点击“文件▷导入▷ $\text{\LaTeX}$  (plain)”，然后选择所要导入的文档。

<sup>4</sup>每一个模版打开后都会有一些建议和提示，根据所要求来选择不同的中文模版

如果是 UTF-8 编码的文档，直接导入就可以了。在文档导入之后，你需要删除 *LyX* 文档开头的 `\begin{CJK}{UTF8}{}` 和结尾的 `\end{CJK}` 字符串，然后在“文档▷首选项▷语言”处，将“语言”项选为“Chinese (simplified)”，将“Encoding”项选为“Unicode (CJK) (utf8)”。

对于 GBK 编码的文档（大部分用 CJK 撰写的文档都使用这个编码），在导入之前，你需要临时在 *tex* 文档的导言区中加入如下一行以确保正确地导入：

```
\usepackage[GBK]{inputenc}
```

在文档导入之后，你需要删除 *LyX* 文档开头的 `\begin{CJK}{GBK}{}` 和结尾的 `\end{CJK}`，然后在“文档▷首选项▷语言”处，将语言选为“Chinese (simplified)”，将“Encoding”项选为“Chinese (simplified) (GBK)”。

### 3.3.2 导出中文 *tex* 文档

在 *LyX* 中完成了中文文档的撰写，你经常需要导出 *lyx* 文档为 *tex* 文档。此时可能碰到的问题就是文档导出后用文本编辑器打开是乱码。这个问题是由于不同的编辑器默认的文本编码不同造成，所以在这里我们提供一种设置方法，使得中文 *lyx* 文档导出后在 WinEdt 编辑器和 *T<sub>E</sub>X*works 编辑器中打开均能正确显示。

对于 GBK 编码的 *lyx* 文档，可以在“文档▷首选项▷*L<sup>A</sup>T<sub>E</sub>X* 序”中，加入如下一行以让 *T<sub>E</sub>X*works 使用 GBK 编码显示文档（这种方法应该只对简体中文的 Windows 系统有效。）：

```
% !TEX encoding = System
```

对于 UTF-8 编码的 *lyx* 文档，可以在“文档▷首选项▷*L<sup>A</sup>T<sub>E</sub>X* 序”中，加入如下一行以让 WinEdt 使用 UTF-8 编码显示文档（这种方法对 WinEdt 最新的 5.5 版本才有效，*C<sub>T</sub><sub>E</sub>X* 套装 2.7 包含 WinEdt 5.5。）：

```
% !Mode:: "TeX:UTF-8"
```

从 *LyT<sub>E</sub>X* 1.6gamma 版本开始，其中的所有中文模板已经设置好了文本编码。



## 4 定制中文模版

如果你认为  $\text{Ly}\text{T}_{\text{E}}\text{X}$  自带的中文模版过于简单，那么你可以修改已有的中文模版，在其导言中进行各种  $\text{T}_{\text{E}}\text{X}$  命令的编写。由于每个人有不同的文档要求，这里会介绍一些如何定制自己的模版的方法，以提高使用  $\text{L}_{\text{Y}}\text{X}$  的效率。由于  $\text{Xe}\text{T}_{\text{E}}\text{X}$  对中文支持比较完善，特别是不需要指定字体，它可以自动从系统字体库中提取，因此我们这里以  $\text{XeCJK-UTF8}$  编码的文档为例来展示如何定制自己的模版。

### 4.1 中文宏包设置

撰写中文文档的首要任务就是要引入中文宏包，我们可以在  $\text{LateX}$  序中（点击“文档”▷“首选项”▷“ $\text{LateX}$  序”）制定：

```
\usepackage[CJKtextspaces,CJKnumber]{xeCJK}
% $\text{xetex}$  CJK 宏包设置
```

### 4.2 字体设置

对于很多人来讲，从模版新建的中文文档的方法似乎过于简单，比如对中文字体种类也许不够，那么我们可以把如下代码复制到文档的导言中，例如：

```
\setCJKfamilyfont{song}{SimSun}      % 设置宋体
\setCJKfamilyfont{hei}{SimHei}       % 设置黑体
.....% 你可以设置更多的中文字体
```

注意：如果你不知道中文字体的英文名称，可以通过使用 `fc-list` 命令得到系统所支持的所有字体信息，比如在  $\text{Microsoft windows}$  下，运行 `cmd`<sup>5</sup>，输入命令 `fc-list ->c:\fonts.txt`（“->”是  $\text{windows}$  管道输出命令），这样你可以把系统所有的字体信息输出并保存在一个文本中，这里即  $\text{C}$  盘目录下的 `fonts.txt` 文本，然后就可以查找你所需要的字体信息了，如果没有的话，可以去网上搜索并下载。

紧接着就可以在文档主体中使用这两种字体，如下代码所示：

<sup>5</sup> $\text{windows}$  下命令行窗口，点击开始 -> 运行，键入 `cmd`，然后回车

```

\documentclass{article}

\usepackage[CJKtextspaces,CJKnumber]{xeCJK}

\begin{document}

.....

\CJKfamilyfont{song} 我是宋体      % 将显示宋体
\CJKfamilyfont{hei} 我是黑体      % 将显示黑体

.....

\end{document}

```

当然对于字体大小的设置，可以不需要在 LaTeX 序中添加代码，直接点击“文档”▷“首选项”▷“字体”，设置字体大小。由于 lyx1.6 版本对中文支持并不完善，所以对一般的 LyX 文档，“字体”中 CJK 选项不可用，除非是特定 LyxCJK 文档，才能够编辑。

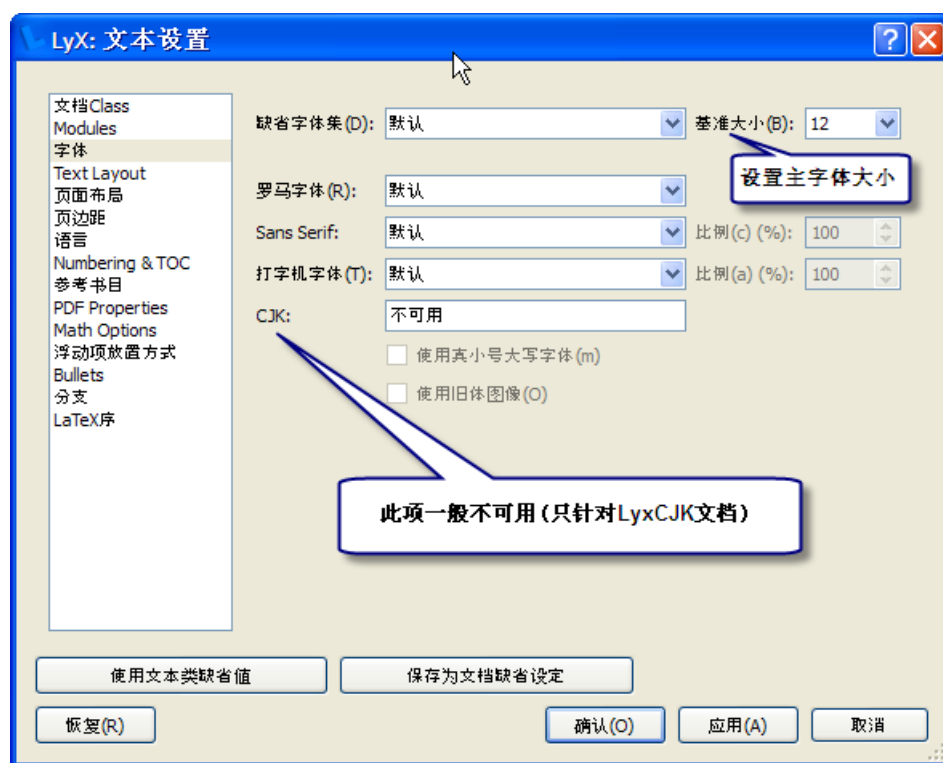


图 3: 指定模版的默认主字体大小为 12 号

### 4.3 行距和段落设置

在文档的导言中可以直接设定中文的基本行距和段落之间的距离：

```
\renewcommand{\baselinestretch}{1.2}
% 设置行距,1.2= 字体大小 (高度)/ 垂直距离上两相邻字之间的间隙长度

\setlength{\parskip}{0.5\baselineskip}
% 设置段落之间距离
```

如果要想精确调整这些参数，建议使用 `geometry` 包；对行或者段落缩进可以使用 `indentfirst` 宏包，只需在 `lateX` 序言中加入它们：

```
\usepackage{geometry}      % 调整行或段落间距
\usepackage{indentfirst}    % 调整行或段落缩进
```

除非你要极其精确地控制文本和段落的样式，一般情况下，你不需要自己在 `LateX` 序言中添加这些 `TEX` 代码来调整这些参数，因为在 `LAX` 中，仅仅通过鼠标就可以调整这些参数了，点击”文档“▷”首选项“设置 `TextLayout`（文本样式）。

## 4.4 浮动图像，表格，目录等的标题中文显示

为了使 `TEX` 中固定的英文术语转成中文，例如要显示“图 1-1”而不是“figure 1-1”，要显示“目录”而不是“content table”等，我们需要在 `LateX` 序言中加入以下代码：

```
\renewcommand\arraystretch{1.2}
%1.2 表示表格中行间距的缩放比例因子 (缺省的标准值为 1), 中文需要更多的间距

\renewcommand{\contentsname}{ 目录}
\renewcommand{\listfigurename}{ 插图目录}
\renewcommand{\listtablename}{ 表格目录}
\renewcommand{\refname}{ 参考文献}
\renewcommand{\abstractname}{ 摘要}
\renewcommand{\indexname}{ 索引}
\renewcommand{\tablename}{ 表}
\renewcommand{\figurename}{ 图}
...
```

这样生成后的 pdf 中，类似 table 1.1 就会自动变成表 1.1。

## 4.5 浮动项的设置

为了使图片能够被正确显示在文档中，经常需要调整浮动项的位置，一般的做法是在 LaTeX 序言中添加  $\text{T}_{\text{E}}\text{X}$  代码，而这里在  $\text{L}_{\text{Y}}\text{X}$  中，只需要使用鼠标就可以轻松完成。比如我们想使插入的浮动图片优先被考虑，那么我们可以采取如下步骤：点击”文档“▷”首选项“▷“浮动项放置方式”，在“高级放置设定”选项里，分别把“页上”，“页底”，以及“浮动项页”和“此处优先”这四项打钩，这样每次被插入的浮动项就不需要再此设置了。

那么这么做会有什么效果呢，根据选项有四种效果：

1. 浮动项页：此图不会被文字环绕，而且如果此项占据当前页面的 50% 以上，就会和其他浮动项一起被放置在单独的一页
2. 页上：如果此浮动项占当前页面少于 70%，就会被放在当前页面的上部
3. 页底：同理，如果此浮动项占当前页面少于 30% 就会被放置在当前页面的底部
4. 如果此图恰好能够被放置在它所在页面中，也就是说它所在的页面能够容纳下，这种效果就是“所见即所得”的效果了。

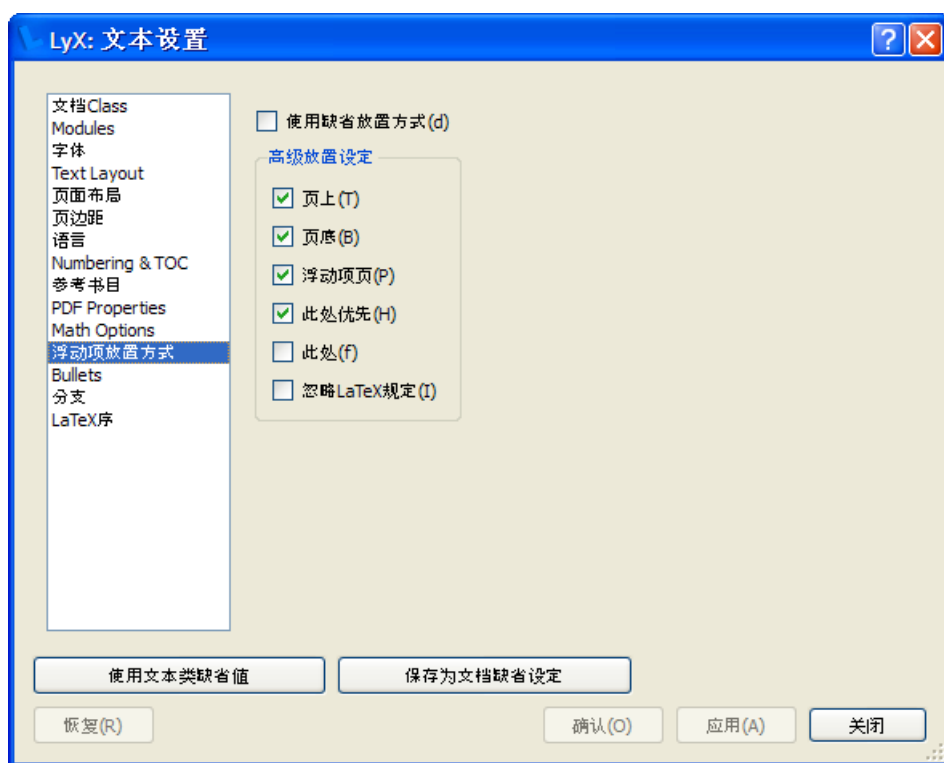
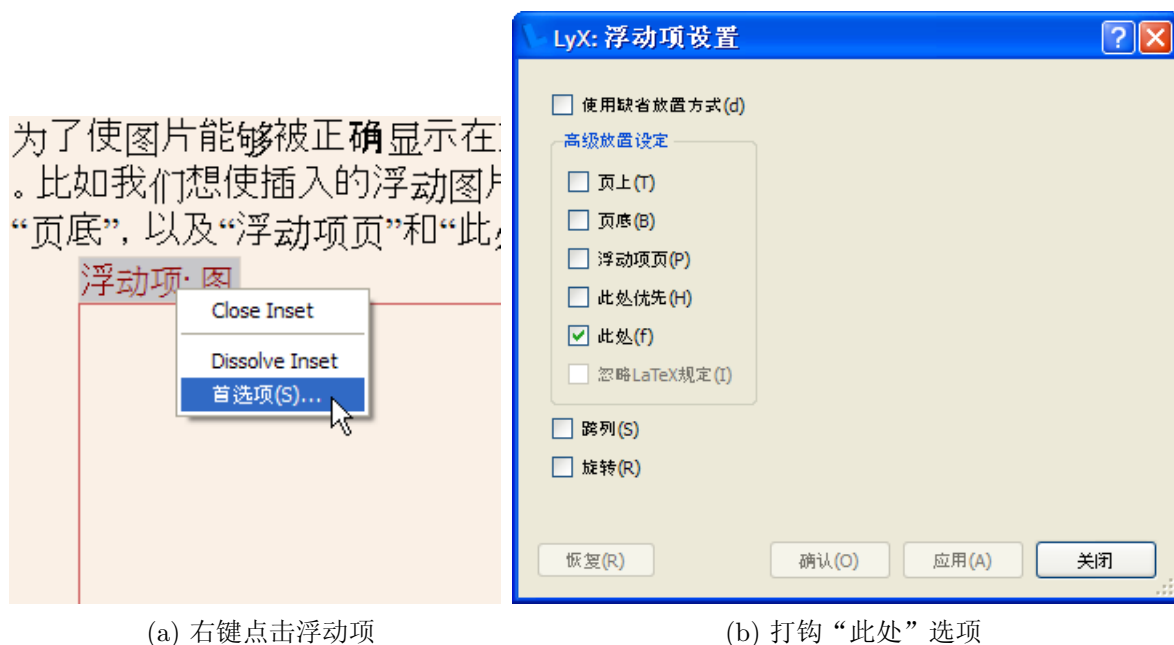


图 4: 浮动项设置: 页上，页底，浮动项页和此处优先

注意：如果要想保证你的一个浮动项在 pdf 中的效果与在  $\text{L}_{\text{Y}}\text{X}$  中编辑的效果一样，有时候你可以点击右键浮动项选择选择“首选项”▷打钩“此处”，这样的话，版面中

所此图的上方所有的项目可能整体会被往上移动或者此图下方的项目整体往下移动来腾出空间 (包括分页), 但是有时候这可能会破坏整个文档的样式。

对于如何自定义浮动项的排版, 我们会在5中讲到。



(a) 右键点击浮动项

(b) 打钩“此处”选项

图 5: 浮动项“所见即所得”设置

## 4.6 文档类设置

一个中文模版不是对所有文章有效的, 因此必须制定此模版的文档类型, 比如 article (文章或者论文), 或者 book(书籍类), 我们可以从“文档”▷“首选项”▷“文档 class”来选择不同的文档类型, 在 LyX 中, 已经有很多文档类型供选择, 如果不特定地设置文档类型, 则默认文档类型一般为 article。

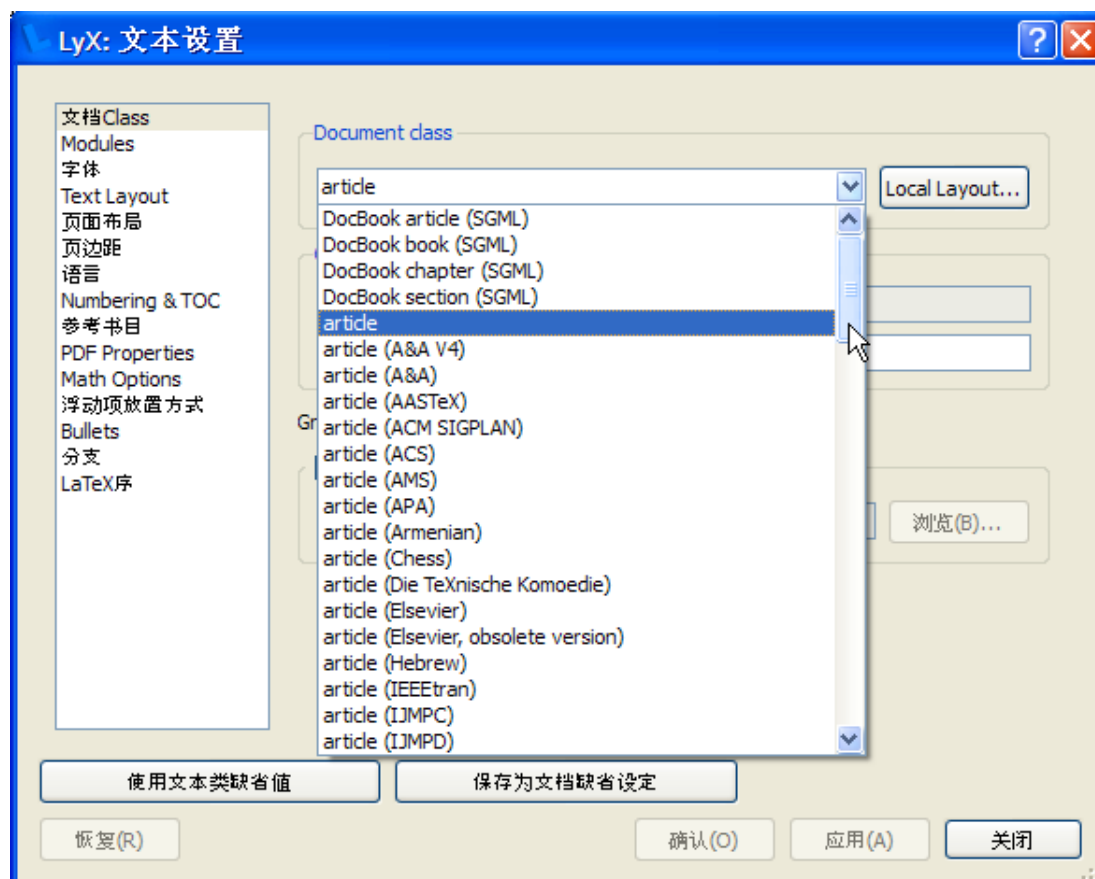


图 6: 文档类型设置

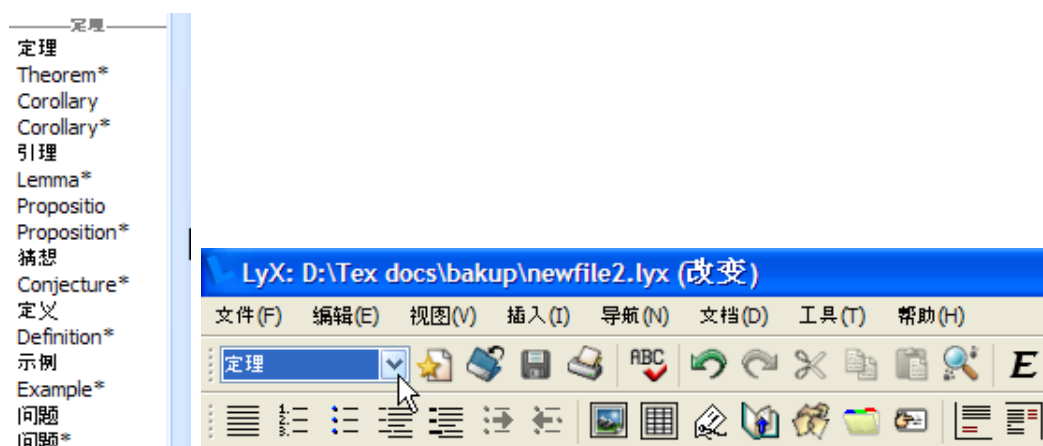
## 4.7 给模版添加环境 (Environment Modules)

在 LyX 中, 还有一个很重要的特性, 就是可以给模版添加很多  $\text{T}_\text{E}\text{X}$  格式环境, 比如想添加对数学各种定义, 猜想, 定理, 证明等环境格式支持, 可以在“文档”▷“首选项”中的 Modules 进行设置, 由于 LyX 中自带了很多标准的环境库, 你可以选择所需要的环境模块。



图 7: 添加 AMS 数学环境

这样，从此模版新建的文档就可以选择很多数学环境，比如定理，引理，猜想，问题等，当然在没有模版的情况，这样做也是极其有用的。



(a) 不同的数学环境格式选项

(b) 选择“定理”格式环境

图 8: 添加 AMS 数学环境例子

比较遗憾的是当前 LyX1.6 版本中还不支持在 Modules 中添加自定义环境，估计后续版

本会考虑的，如果要添加自己的环境怎么办呢？老办法，在 LaTeX 序中添加自己定义的环境，比如这个帮助文档就使用了一个自定义的背景为灰色的环境，以便插入 tex 代码，使之突出显示。

```
% 使用此 grayBox 环境的先决条件，需要导入 framed 和 color 宏包
\newlength{\boxwidth}
\definecolor{shadecolor}{gray}{0.85}
% 环境定义如下
\newenvironment{grayBox}{\def\FrameCommand{\fboxsep=\FrameSep
\fbboxrule=\FrameRule\fcbox{white}{shadecolor}}
\MakeFramed {\setlength{\boxwidth}{\textwidth}
\addtolength{\boxwidth}{-2\FrameSep}
\addtolength{\boxwidth}{-2\FrameRule}
\setlength{\hspace}{\boxwidth} \FrameRestore}}
{\endMakeFramed}
```

## 4.8 页面设置

在 LyX 中，页面的设置也是相当的容易，比如要设置纸张的大小为 A4, 可以点击“文档”▷“首选项”在“页面布局”中设置格式为 A4，同时还可以设置纸张的方向；如果要精确控制页边距，可以在“页边距”选项中精确设置，实际上 LyX 会根据这些参数自动调用 geometry 宏包进行设置。(注意：页边距选项中的前七个参数选项分别对应下图所示的七个参数)



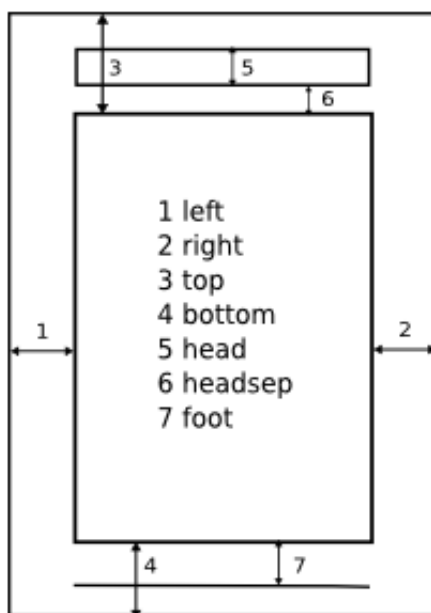


图 9: geometry 宏包的页面设置参数

## 4.9 目录和编号

如果要想控制目录或者编号的级数，可以在文档首选项中“Numbering & TOC”中设定，使用鼠标拖动进度条来控制级别，其中编号的最大级别为 5 级，但是，我们不推荐使用过深的编号，因为这会降低文章的可读性。



图 10: 编号级别和目录显示

## 4.10 文献

文献是论文中很重要的一个部分，在 LyX 中可以选择不同的文献引用格式，点击“文档”▷“首选项”在“参考书目”选项中选择引用的格式，LyX 提供两种主流的文献引用格式：

- Natbib<sup>6</sup>: 是一个标准的 L<sup>A</sup>T<sub>E</sub>X 命令 `\cite` 的实现宏包，它和其他比如 harvard, apalike, chicago, astron, authordate 等宏包也相兼容，它提供两种最基本的命令，`\citet` 和 `\citep` (`\citet*` 和 `\citep*` 用于打印完整引用)

<sup>6</sup>关于更加详细 Natbib 的样式，请查看 <http://merkel.zoneo.net/Latex/natbib.php>

命令	效果预览
<code>\citet{jon90}</code>	Jones et al. (1990)
<code>\citet[chap. 2]{jon90}</code>	Jones et al. (1990, chap. 2)
<code>\citep{jon90}</code>	(Jones et al., 1990)
<code>\citep[chap. 2]{jon90}</code>	(Jones et al., 1990, chap. 2)
<code>\citep[see][ ]{jon90}</code>	(see Jones et al., 1990)
<code>\citep[see][chap. 2]{jon90}</code>	(see Jones et al., 1990, chap. 2)
<code>\citete*{jon90}</code>	<b>Jones, Baker, and Williams (1990)</b>
<code>\citep*{jon90}</code>	<b>(Jones, Baker, and Williams, 1990)</b>

表格 1: Natbib 样式示例

- Jurabib<sup>7</sup>: 此包是德国人 Jens Berger 创建一个主要用于法律和人文科学的文献宏包。特别是对于学习人文科学的人来讲, 此包是相当有用的。

## 4.11 数学宏包的使用

如果要使模版自动对数学公式的支持, 就必须要在文档首选项中的“Math Options”(数学选项) 打钩“使用 AMS 数学包”或者“使用 esint 包”, 或者两者都打勾, 请注意, 如果打钩“自动使用 AMS 数学包”或者“自动使用 esint 包”仅仅在 L<sup>A</sup>T<sub>E</sub>X 下插入数学环境时有效(L<sup>A</sup>T<sub>E</sub>X 会依次自动添加 AMS 和 esint 宏包), 如果在文档中插入 T<sub>E</sub>X 代码并且使用了两者宏包之一的代码, 这时候就必须打钩“使用 AMS 数学包”或“使用 esint 包”, 因为 L<sup>A</sup>T<sub>E</sub>X 并不能智能识别所插入的数学 T<sub>E</sub>X 代码而自动引入 AMS 和 esint 包, 在要编辑数学有关模版(或者文档)的时候, 强烈建议把“使用 AMS 数学包”和“使用 esint 包”都打钩, 否则可能出现编译出错的情况。

<sup>7</sup>关于 Jurabib 的详细用法, 可以下载[Jurabib 的使用手册 \(英文\)](#)

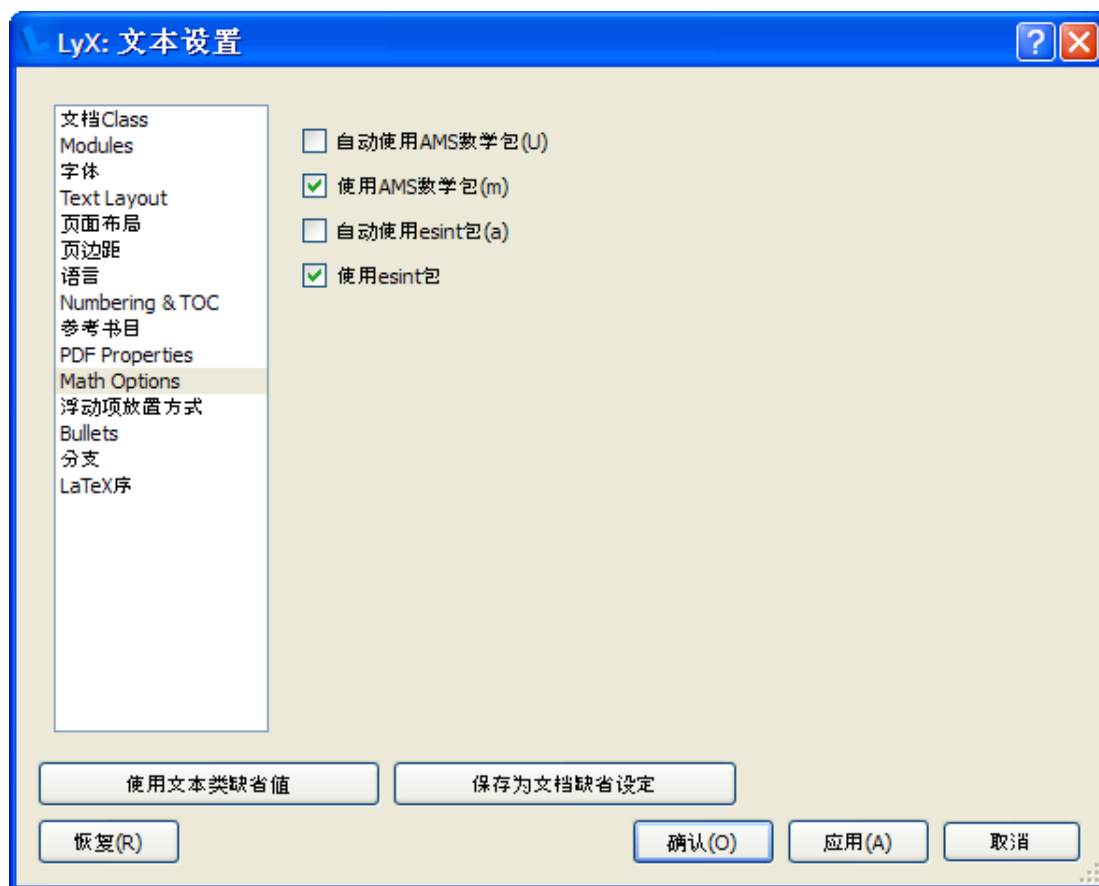


图 11: 打钩导入 AMS 和 esint 宏包

## 4.12 输出 pdf 的设置

我们可以在文档首选项中的“PDF properties”指定是否使用 hyperref 包来使输出的 pdf 更加具有可读性，比如 pdf 目录层次的结构，以及文档链接颜色的高亮等，当你撰写中文文档时，必须要在“Additional option”（附加参数）中指定参数 unicode 为 false，否则 pdf 的目录会出现乱码。

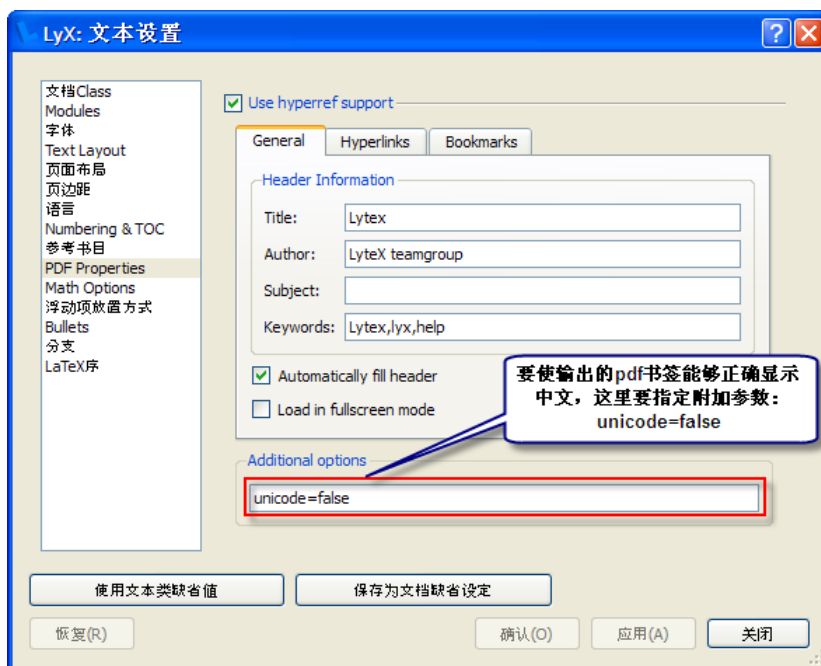


图 12: 设置 pdf 输出选项

### 4.13 列表符号的设置

如果对文档的列表符号有特殊要求，可以在文档首选项里的“bullets”中设定列表的符号形状，由于 LyX 中内置了大量不同种类的符号，因此我们不需要在 LaTeX 序言中设定，这样能够节省不少时间。

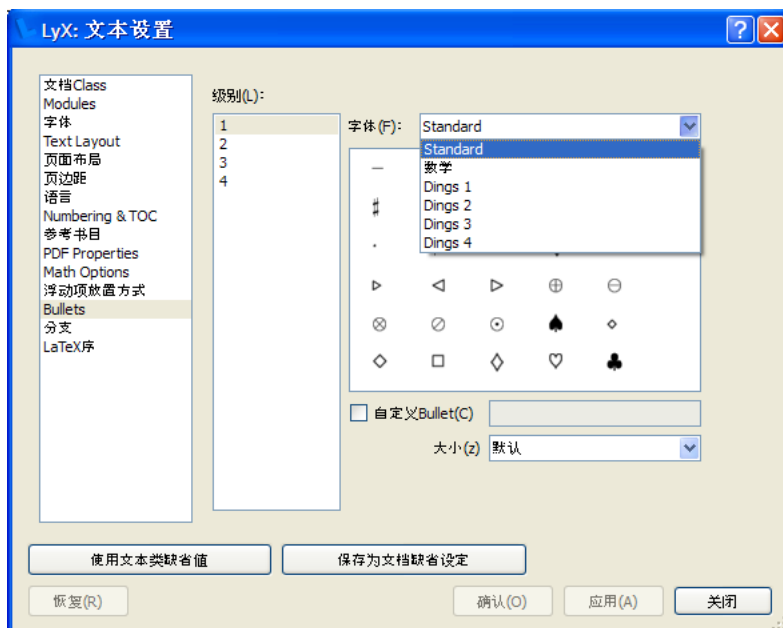


图 13: 设置列表符号形状

## 4.14 使用分支

在 LyX 里，分支是一个非常有特色的选项，简单来说，分支就相当于一个环境或者说一个可编译或不编译的 LyX 文档片段，我们可以设定被某一个分支所包含的内容在最终的输出中是否应该被编译，例如你可以使用 LyX 中撰写一份试卷，其中包括问题和答案，对于学生而言，可以不编译答案分支（不激活分支编译），在考试结果出来之后，然后设置答案在 pdf 输出中应该被编译（激活分支编译），这样就很方便使用一份文档来管理问题和答案。我们可以通过在文档首选项中的“分支”中添加分支以及设定分支的颜色以示区别，然后就可以很简单在文档中插入所需要分支了。

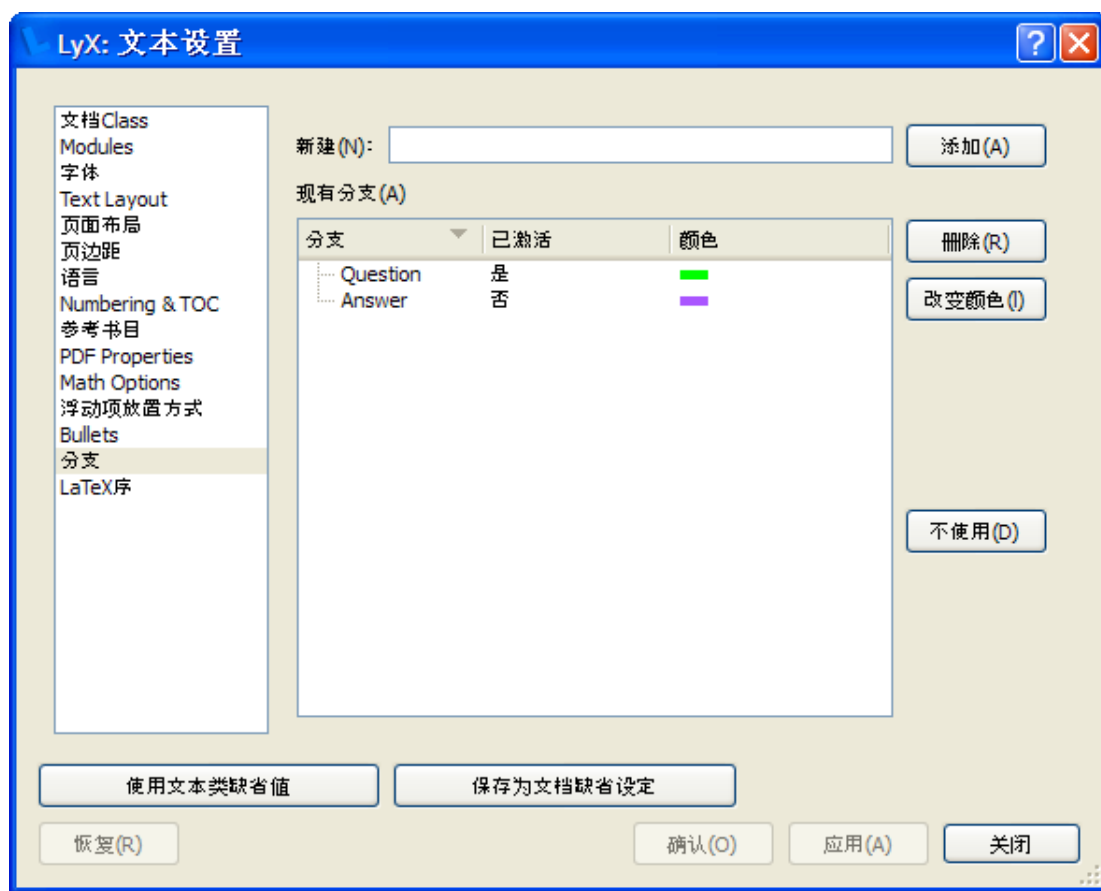


图 14: 添加自定义的分支

## 5 Lyx 内容排版

这部分主要介绍一些常用的编辑和排版技巧，以便提高使用 LyX 的效率。如果要查看详尽的使用手册可以通过点击“帮助”▷“入门教程”或者“用户手册”。

### 5.1 快捷键

想要在 LyX 中高效排版，首先必须要了解几个非常重要的默认快捷键。

快捷键	说明	使用频率
Ctrl+B	使文本变成粗体	中
Ctrl+E	使文本变成斜体 (强调)	中
Ctrl+L	快速插入 $\text{\LaTeX}$ 代码	高
Ctrl+M	快速插入数学输入环境	高
Ctrl+P	查看 Xetex 编译的 pdf 文件	高
Ctrl+S	快速保存正在编辑 (激活) 的文档	高
Ctrl+U	给文本添加下划线	中
Ctrl+Z	撤销上一次操作	高
Ctrl+return	换行 (在数学环境下也可用)	高

表格 2: 重要的常用快捷键

说明:

- “Ctrl+L” 可以插入  $\text{\TeX}$  代码, 例如 `\begin{grayBox}` 插入已自定义的环境。
- 使用快捷键 “Ctrl+M” 就可以立刻打开数学输入支持环境, “Ctrl+Enter” 可以在数学环境中插入新行, 特别适合在插入了 AMS 对齐的环境下, 可以使用此快捷键来换行输入公式化简等。
- 使用 “Ctrl+P” 可以直接预览 Xetex 生成后的 pdf 文件, 非常方便 (注意, “Ctrl+P” 并不是  $\text{\LaTeX}$  官方网站所发行的  $\text{\LaTeX}$  中所默认的快捷键, 而是  $\text{\LaTeX}$  套件重新编译后添加的, 因为官方版本默认并不支持 Xetex 编译)

## 5.2 文本样式排版

如果想要加粗 (快捷键 “Ctrl+B”) 或者强调 (快捷键 “Ctrl+E”) 或者给文本加下划线 (快捷键 “Ctrl+U”), 可以直接使用他们的快捷键, 或者你想一步实现所有的设置, 可以 “右键单击选择的文本或者段落” ▷ “文本样式” ▷ “自定义” 中对文本的各种样式参数进行设置。



图 15: 文本样式设置

### 5.3 文章段落排版

在对文章的段落进行排版时，可以很容易的对段落进行首行缩进，行间距以及段落对齐方式设置，只需通过“右键单击所要设置的段落”▷“段落设置”即可。

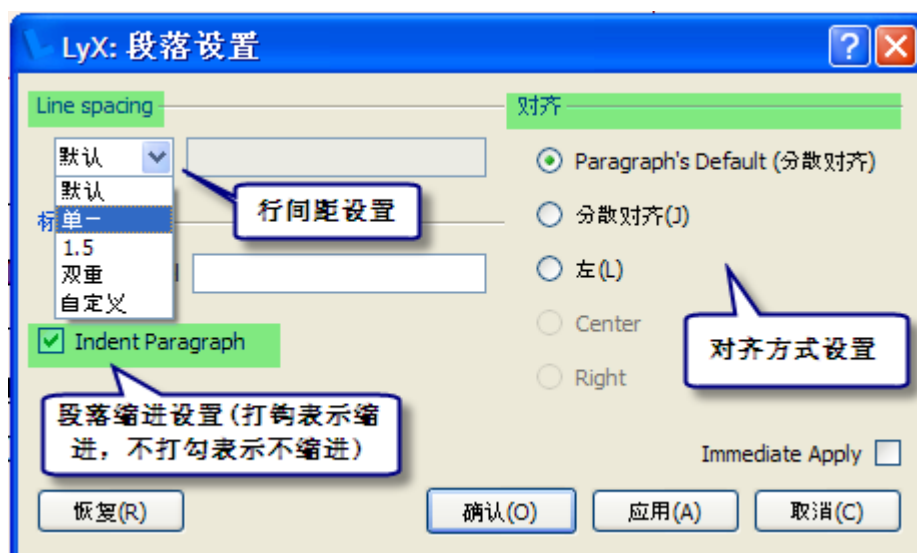


图 16: 设置段落



## 5.4 浮动项目排版

在 LYX 中插入浮动项的首选设置已经在前面4.5已经说明，这里主要简单介绍对算法，图片和表格操作的一些方法和技巧。

### 5.4.1 插入算法

可以通过点击“插入”▷“浮动项”▷“算法”，插入算法伪代码（请尽量不要插入任何其他具体编程语言的算法如 C 或者 Ruby），由于 LYX 默认提供的算法功能太弱，建议使用 `algorithmic` 宏包，在序言中加入以下代码：

```
% 添加算法支持宏包
\usepackage{algorithms}
\usepackage{algorithmic}
% 下面重新定义命令以支持中文 (默认是英文)
% 我们不建议这样做，除非是为了做演示，因为任何学习计算机的人都能看懂伪代码
\renewcommand{\algorithmicrequire}{\textbf{ 需要:}}
\renewcommand{\algorithmicensure}{\textbf{ 前提:}}
\renewcommand{\algorithmicend}{\textbf{ 算法结束}}
\renewcommand{\algorithmicif}{\textbf{ 如果}}
\renewcommand{\algorithmicthen}{\textbf{ 那么}}
\renewcommand{\algorithmicelse}{\textbf{ 否则}}
\renewcommand{\algorithmicelsif}{\textbf{ 否则}}
\renewcommand{\algorithmicendif}{\textbf{ 条件语句结束}}
\renewcommand{\algorithmicfor}{\textbf{ 对}}
\renewcommand{\algorithmicforall}{\textbf{ 对所有的}}
\renewcommand{\algorithmicdo}{\textbf{ 进行如下}}
\renewcommand{\algorithmicendfor}{\textbf{for 循环体结束}}
\renewcommand{\algorithmicwhile}{\textbf{ 当}}
\renewcommand{\algorithmicendwhile}{\textbf{while 循环体结束}}
\renewcommand{\algorithmicloop}{\textbf{loop 循环开始}}
```

```

\renewcommand{\algorithmicendloop}{\textbf{loop 循环结束}}
\renewcommand{\algorithmicrepeat}{\textbf{反复做}}
\renewcommand{\algorithmicuntil}{\textbf{直到}}
\renewcommand{\algorithmicprint}{\textbf{打印}}
\renewcommand{\algorithmicreturn}{\textbf{返回}}
\renewcommand{\algorithmictrue}{\textbf{真值}}
\renewcommand{\algorithmicfalse}{\textbf{假值}}

```

为了操作更简便，我们可以先在 L<sup>A</sup>T<sub>E</sub>X 中通过“插入”▷“浮动项”▷“算法”来插入算法，然后调整版面，最后在其中插入 algorithmic 环境代码和算法的具体内容，一个简单的例子如下<sup>8,9</sup>：

---

#### 算法 1 状态空间中的深度优先递归算法

---

```

1: if current_state is a member of closed then
2:   return SUCCESS
3:   add current_state to closed
4: end if
5: while current_state has unexamined children do
6:   Child:=next unexamined child
7:   if child not member of closed and depthsearch(child)=SUCCESS then
8:     return SUCCESS
9:   else
10:    return FAIL
11:   end if
12: end while

```

---

所插入的相应 T<sub>E</sub>X 源代码：

```

\begin{algorithmic}[1]
\IF{current\_state is a member of closed}
\RETURN SUCCESS

```

---

<sup>8</sup>参考了 algorithm 宏包 (下载地址: <http://ftp.join.uni-muenster.de/pub/software/CTAN/macros/latex/contrib/algorithms.zip>) 中的说明手册

<sup>9</sup>算法改编自 George F.Luger(美) 编写的 Artificial Intelligence Structures and Strategies for Complex Problem Solving(Fifth Edition)

```

\STATE add current\_state to closed
\ENDIF
\WHILE{current\_state has unexamined children}
\STATE Child$:= $next unexamined child
\IF{child not member of closed and depthsearch(child)=SUCCESS}
\RETURN SUCCESS \ELSE
\RETURN FAIL
\ENDIF \ENDWHILE
\end{algorithmic}

```

在上面插入的代码中，注意开始的`\begin{algorithmic}[1]`的“1”表示每隔一行便显示行号，为了使标题 `algorithm` 显示成中文，在整个算法的前面插入如下代码：

```

\floatname{algorithm}{ 算法}

```

#### 5.4.2 插入浮动图片

通过点击“插入”▷“浮动项”▷“图”插入浮动项之后，在标题的上一行或者按“Enter”键可以使插入的图片在标题上面显示，反之亦然。插入之后我们可以通过调整段落，见段落排版5.3从而设置图片的对齐方式，比如我们经常让图片经常居中显示，则可以打钩“**center**”选项。

在浮动图内还可以继续插入浮动子图，而且每个浮动子图都会被阿拉伯数字和小写英文自动编号。如果想使用改变浮动图的编号，可以在 LaTeX 序言中插入以下代码：

```

% 按章节编号
\numberwithin{figure}{section}
% 按子章节编号
\numberwithin{figure}{subsection}

```

由于使用了`\numberwithin`命令，所以一定要在 LaTeX 序言中要导入 AMS 数学宏包或者在文档首选项中打钩“使用 AMS 数学包 (m)”

### 5.4.3 插入浮动表格

插入浮动表格操作与插入浮动图片类似，这里不详述。

#### 5.4.3.1 长表格

对于很长的表格 (可能要跨页)，我们可以在“右键单击表格”▷“首选项”▷打钩“长表格”来设置表格，其中参数解释如下：

1. 页首：当前行被作为标志行，只被显示在每次分页之后的表格首行中
2. 第一页首：当前行作为标志行，只被显示在当前页面的表格首行中，在后续页面的表格中不会被显示
3. 页脚：当前行作为末行，会被显示在其他所有的延续表格中 (除了最后一行，除非自己另外定义了末行)
4. 最后页尾：当前行作为末行，只会出现在最后一个延长表格的页面中
5. caption(标题)：

注意：在设置长表格的时候，一定要注意鼠标的光标落在的表格的行（标志行？行首？行尾？），否则可能会出现排版差错。

#### 5.4.3.2 长表格的位置调整

由于长表格默认位置是据中，要改变其位置，我们可以通过在长表格的前面插入以下 $\text{\TeX}$  代码来达到目的：

```
%value 为所指定的长度单位值
% 设定表格和页面左边的距离
\setlength{\LTleft}{value}
% 设定表格和页面右边的距离
\setlength{\LTright}{value}
% 返回默认值
\setlength{\LTleft}{\fill} % 如果你修改了\LTleft, \fill 是默认值
或者
\setlength{\LTright}{\fill} % 如果你修改了\LTright
```

比如我们设置下面表格和页面左边距对齐，也就是说距离为0：

请注意，此调整只对长表格有效！

1	2	3
4	5	6
7	8	9

表格 4: 靠齐页面左边距的表格

## 5.5 Box(盒子)

Box 有两种格式属性可供选择:

- MiniPage: 有和母版完整功能的迷你型页面
- Parbox: 类似于 word 中的文本框不能够有自己的脚注, 只是一个  $\text{\LaTeX}$  环境而已

Box 中的内容可以选择垂直对齐方式, 总共有:

上: 例如: 

this is a test!
-----------------

中间: 例如, 

this is a test!
-----------------

下: 例如, 

this is a test!
-----------------

拉伸: 例如 

this is a test!
-----------------

除此之外, 盒子本身也可以类似地设置对齐方式, 这里不详述了。

### 5.5.1 自定义 box

Box 的一些常用属性会在下面的5.5.2和5.5.3谈到, 这里我们看一看如何使用自己定义的 box。

#### 5.5.1.1 设置边框的厚度

我们可以通过以下插入代码来改变 box 边框的厚度:

```
\setlength{\fboxrule}{value}    %value 为边框的厚度
```

要返回默认值, 插入以下代码:

```
\setlength{\fboxrule}{0.3pt}    % 设置为默认值, 0.3pt 为边框的默认厚度
```

例如:

this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!

\setlength{\fboxrule}{3pt}

### 5.5.1.2 设置边框到内容的距离

可以通过插入以下代码达到目的:

```
\setlength{\fboxsep}{value}    %value 为内容到边框的距离
```

要返回默认值, 插入以下代码:

```
\setlength{\fboxsep}{3pt}    % 设置为默认值, 3pt 为默认值
```

例如:

this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!

\setlength{\fboxsep}{20pt}

### 5.5.1.3 设置阴影大小

当 box 使用阴影外观时, 可以插入以下代码控制阴影的厚度或者大小:

```
\setlength{\shadowsize}{value}    %value 为说要设置的值, 默认值为 4pt
```

例如:

this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!this is a test!

\setlength{\shadowsize}{1.5pt}

#### 5.5.1.4 设置背景颜色

如果选择了 shaded background, 可以通过“工具”▷“首选项”▷“Look & Feel”▷“颜色”来设置背景颜色, 或者插入以下代码达到目的:

```
% 前提是要导入 color 宏包
\definecolor{shadecolor}{cmyk}{0.5,0,1,0.5}
```

例如:

this is a test!

#### 5.5.2 迷你页面 (MiniPage)

迷你页面可以算的上一个拥有母页几乎所有的功能的页面, 通过点击“插入”▷“Box”来操作 (插入的 Box 默认属性是 minipage)。通过点击右键可以设置迷你页面的外观属性:

1. Frameless: 没有边框效果
2. simple frame: 这是 L<sup>A</sup>T<sub>E</sub>X 中默认的
3. simple frame, page breaks: 和 2 类似, 只不过可以换页
4. Oval,thin: 圆矩形, 有比较薄的边框, 边框厚度为 0.4pt
5. Oval,thick: 圆矩形, 有比较厚的边框, 边框厚度为 0.8pt
6. drop shadow: 背景有投影而是实心黑色的盒子, 阴影厚度为 4pt
7. shaded background: 背景是全红色的, 而且使用整个列的宽度, 类似于本帮助文档中的 grayBox
8. double frame: 双层边框,

要使用这些炫彩的外观, 在 LateX 序言中需要导入 fancybox 宏包, 下面是一些外观效果:

默认的 frameless

simple frame

oval thin

drop shadow

shaded background

double frame

以下是两个完整的 minipage，左边使用了德语，而右边使用美国英语。

Die Modelle der SyncMaster 70er Serie beeindrucken mit edlem Design und moderner Technik. Dabei macht das Touch of Color Design in elegantem Schwarz diese Modelle zu einem echten Blickfang. Dank hochwertiger Komponenten konnte der Stromverbrauch reduziert werden. Dies trägt positiv zum Schutze der Umwelt bei und schont das eigene Budget. Das Modell kommt im 16:9-Format daher und bieten Full HD Auflösung (1.920 x 1.080 Pixel). Das sehr hohe dynamische Kontrastverhältnis von 50.000:1 und die schnelle Reaktionszeit von nur 2 ms (G/G) sorgen stets für klare und detailreiche Bilder - auch bei schnell bewegten Bildinhalten.<sup>a</sup>

---

<sup>a</sup>Das ist Deutsch

Intel Technology Intel® QuickPath Interconnect. This Intel-exclusive innovation makes it easier for data to move quickly from the processor to the Intel X58 chipset, delivering better performance and responsiveness with the most demanding applications. Intel® Hyper-Threading Technology. By allowing each of the four processing cores to process two application “threads” simultaneously, Intel Core i7 processors can accomplish more simultaneously—allowing users to run more demanding applications simultaneously while experiencing superior responsiveness. Intel® Turbo Boost Technology. This new technology dynamically increases ... <sup>a</sup>

---

<sup>a</sup>This is english

### 5.5.3 文本框 (Parbox)

我们可以通过鼠标右键单击“插入的 Box”▷“首选项”▷“内框”设置为 Parbox，如下图所示：





图 17: 设置内框为 Parbox

Parbox 的垂直对齐方式前面已经讲过，这里举一个简单的例子：

this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!

this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!  
 this is a test! this is a test! this is a test!  
 a test! this is a test! this is a test!

## 5.6 有颜色的 Box

前面我们讨论过当使用 shaded background 的 box 时，可以设置 box 的背景填充颜色，但是如果对于普通的 box 该如何设置内容的背景颜色，这时候我们可以使用 `\colorbox` 命令。

```
\colorbox{color}{内容...}
```

%color 的预设值有 black, blue, cyan, green, magenta, red, white, and yellow.

比如: 女士们，先生们，请注意！

如果要想使用框架的 colorbox，可以使用命令\colorbox:

```
\colorbox{frame color}{box color}{内容...}
```

例如:

如果想进一步调整边框厚度以及边框和内容的距离，请参照5.5.1。

请注意: 在 colorbox 和 fcolorbox 中文字不能断行! 如果想要在其中插入段落或者更复杂的内容，请在它们内部插入 minipage, 或者 parbox.

测试:

显然断行有问题! 因此我们在 colorbox 内部插入 minipage 和 parbox:

1. 这是在 clolorbox 内部的一个 Minipage，背景为灰色
2. 有一个 $4 \times 4$ 的表格

断行示例: 这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

断行示例: 这是一个在 fcolorbox 里面的一个 parbox(frameless).

断行示例: 这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试，这是一个测试。

00	010	01101	11010
----	-----	-------	-------

## 5.7 自定义有颜色的 box

如果不使用 LyX 中自带的 Box 而要如何定义一个有着类似外观而且可以换行的自定义 box 呢? 本帮助文档所使用的 grayBox 就是一个例子，首先看看 grayBox<sup>1</sup>环境是如何定义的:

```
%grayBox,needs shadecolor, \boxwidth and the framed package
\newenvironment{grayBox} % 新环境定义的开始
{\def\FrameCommand{ % 定义边框命令
\fbboxsep=\FrameSep % 使 box 和 frame 与内容距离相等, 请看5.5.1
\fbboxrule=\FrameRule % 使 box 和 frame 的边框厚度相等
\fcolorbox{white}{shadecolor}} % 使用了 fcolorbox, 关于 fcolorbox 的详细请看上节
%FrameCommand 定义结束, 以下开始定义命令 MakeFramed
\MakeFramed % 开始构建 frame
{\setlength{\boxwidth}{\textwidth} % 设置 box 的宽度和文本宽度一样
\addtolength{\boxwidth}{-2\FrameSep} % 调整 box 的宽度
\addtolength{\boxwidth}{-2\FrameRule} % 同样调整 box 的宽度
\setlength{\hspace}{\boxwidth} \FrameRestore}} % 设置整个宽度和 box 的宽度一样
\endMakeFramed} % 定义\endMakeFramed 结束
```

<sup>1</sup>此例来自于 latex community, <http://www.latex-community.org/>

比如, 我们可以从此 grayBox 修改为一个 yellowBox(以示区别), 比如:

```
\newenvironment{yellowBox}
{\def\FrameCommand{
\fbboxsep=\FrameSep
\fbboxrule=\FrameRule
\fcolorbox{white}{lightyellow}} 边框颜色默认, 内容背景色为浅黄色
%FrameCommand
\MakeFramed
{\setlength{\boxwidth}{\textwidth}
\addtolength{\boxwidth}{-2\FrameSep}
\addtolength{\boxwidth}{-2\FrameRule}
\setlength{\hspace}{\boxwidth} \FrameRestore}}
\endMakeFramed}
```

效果如下:

我们以 grayBox 为例, 自己修改的一个背景为浅黄色的 box, 当然你可以根据自己的喜好修改边框颜色和厚度, 以及内容与边框的距离。

如果要修改边此 box 的框大小和厚度, 就要在 box 之前插入以下命令 (和 box 的 `\fboxrule` 以及 `\fboxsep` 命令类似):

```
\setlength{\FrameRule}{value}    % 修改边框的厚度为 value, 默认值为 0.4pt  
\setlength{\FrameSep}{value}    % 修改边框和内容的距离为 value, 默认值为  
9pt
```

例如:

此 box 有边框, 而且边框厚度为 0.1cm, 边框与内容的距离为 1cm

此外, 我们还可以改变此边框的所占的列宽

这个 box 和上面一个类似, 只不过是一个所占列宽改变了。换行测试: 这是一个自定义 box, 这是一个自定义 box, 这是一个自定义 box, 这是一个自定义 box, 这是一个自定义 box, 这是一个自定义 box, 这是一个自定义 box.

注意: 此类 Box 永远居中!

## 6 数学公式排版

### 6.1 使用数学工具条

在进行数学输入之前，强烈建议在 LyX 中打开数学工具条视图以方便的选择所需要的数学符号和模版，通过点击“视图”▷“工具条”▷“数学工具条 (自动)”和“数学 (自动)”来打开数学面板。



图 18: 选择显示数学工具条

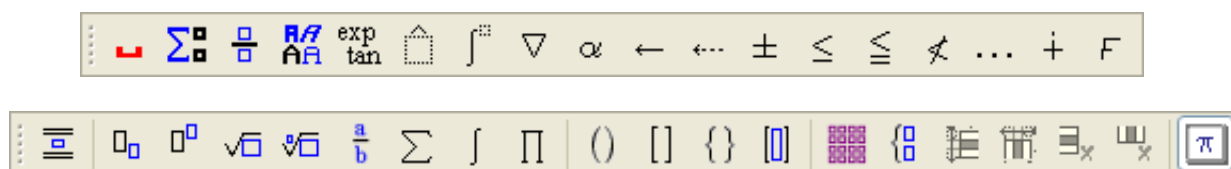


图 19: 数学工具条 (可以拖动)

### 6.2 插入行内或者显示公式

当插入公式时，可以选择插入“行内公式”或者“单独显示公式”，在定理以及引理中，最好使用单独显示公式便于显示和编号，在段落中引用到数学函数或者公式时，可以使用“行内公式”。建议使用打印 pdf 来查看数学公式的最终效果，然后再决定使用哪一种显示方式最适合你的论文。

$$\sum_{i=1}^N 2^i = \dots\dots\dots \text{这是一个行内公式}$$

$$\sum_{i=1}^N 2^i = \dots\dots \text{这是一个单独显示公式}$$

### 6.3 公式编号

给数学公式编号，一般有两种方法：

1. 通过单击“插入”▷“公式”▷“编号公式”(快捷键 Ctrl+Alt+N), 来直接插入带有编号的公式，比如：

$$z^2 = x^2 + y^2 \quad (1)$$

2. 通过右键单击公式，然后选择“Toggle Labeling/Numbering”就可以给公式编号了，也可以选择“编辑”▷“数学”▷“Number this line”(给此行进行编号)或者“Number whole formula (对整个公式进行编号)”。



图 20: 公式编号

### 6.3.1 内置公式编号样式

AMS 宏包内置了一些数学公式编号的样式，比如改成罗马数字编号或者字母编号，可以在 LaTeX 序言中加入或者插入如下代码：

```
\renewcommand{\theequation}{\roman{equation}}
```

```
% 罗马数字小写编号
```

```
% 效果
```

$$z^2 = x^2 + y^2 \quad (\text{ii})$$

```
\renewcommand{\theequation}{\Roman{equation}}
```

```
% 罗马数字大写编号
```

```
% 效果如下：
```

$$z^2 = x^2 + y^2 \quad (\text{III})$$

```
\renewcommand{\theequation}{\alph{equation}}
```

```
% 拉丁语小写编号
```

```
% 效果如下：
```

$$z^2 = x^2 + y^2 \quad (\text{d})$$

```
\renewcommand{\theequation}{\Alph{equation}}
```

```
% 拉丁语大写编号
```

```
% 效果如下：
```

$$z^2 = x^2 + y^2 \quad (\text{E})$$

如果想要返回默认的格式（阿拉伯数字编号），只需在后面插入如下代码：

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

### 6.3.2 按章节或段落对公式进行编号

对于如果想数学公式按照所在的章节进行编号，可以使用 AMS 宏包中的 “\numberwithin” 命令，如果你把此代码加入文档的 LaTeX 序中，则所有的公式将按照章节进行编号，如果你只想应用在某一公式上，可以在公式前插入 T<sub>E</sub>X 代码，例如：

```
\numberwithin{equation}{section}
```

% 或者

```
\numberwithin{equation}{paragraph}
```

(公式输入).....

按章节进行编号效果:

$$z^2 = x^2 + y^2 \quad (6.6)$$

按段落进行编号效果:

$$z^2 = x^2 + y^2 \quad (6.3.2.0.7)$$

如果要使用原来默认的编号，则可以使用重置命令：

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

```
\makeatletter
```

```
\@removefromreset{equation}{paragraph}
```

```
\makeatother
```

% 由于\numberwithin 自动使用了\addtoreset 命令，因此我们要移除它

% 重置后效果:

$$z^2 = x^2 + y^2 \quad (8)$$

### 6.3.3 对子公式编号

有时候，我们有一个母公式，然后要对由此母公式所导出的子公式进行编号，我们可以使用 AMS 的 “\subequations” 环境，例如以下例子<sup>10</sup>：

<sup>10</sup>此例子摘自 L<sub>A</sub>T<sub>E</sub>X 的数学英文帮助文档 19.3 部分



$$A = C - B \quad (9)$$

$$B = C - A \quad (9a)$$

$$C = A - B \quad (9b)$$

注意：先要输入此类公式分三步：

1. 先输入编号的母公式，然后紧接着插入代码 “`\addtocounter{equation}{-1}`” 和 “`\begin{subequations}`” (可以 L<sup>A</sup>T<sub>E</sub>X 快捷键 “Ctrl+L” 来快速插入代码)
2. 然后输入编号的子公式
3. 输入子公式环境结束代码 “`\end{subequations}`”

注意，上面子公式的编号的都是以“数字-英文字母”格式的，如果子编号的式子数目超过了 26，就可能不符合要求了，因此我们还可以重新定义子编号的格式为“数字 # 数字”的形式的，我们可以在 LateX 序言中自定义一个子公式输入环境，添加如下代码：

```
\newenvironment{mysubeq}
{\begin{subequations}\renewcommand{\theequation}
{\theparentequation\#\arabic{equation}}\}
{\end{subequations}}}
```

% 示例：

$$A = C - B \quad (10)$$

$$B = C - A \quad (10\#1)$$

$$C = A - B \quad (10\#2)$$

### 6.3.4 自定义公式编号样式

尽管 AMS 宏包提供了数字，英文，罗马数字等编号，但是如果想给公式提供自己定义的编号该怎么办呢？我们可以简单的使用 `\tag` 命令达到目的，在 L<sup>A</sup>T<sub>E</sub>X 数学环境中公式的末尾键入 `\tag`，L<sup>A</sup>T<sub>E</sub>X 就会自动显示出一个 tag 框，然后输入编号的内容即可，但是总体格式仍然是一对括号所包括的编号内容)，如果甚至要修改括号，我们可以在 LaTeX 序言中添加以下代码<sup>11</sup>，比如 (`#1` 表示宏命令中编号的参数，表示第一个参数，尖括号可以任意被替换)：

```
\def\tagform@#1{\maketag@@@{<#1>}}
% 此代码加入 LATEX 序言中, 这将会导致所有的方程编号都是在尖括号 <...> 内而不是默认的圆括号 (...)
```

或者在文档中插入以下代码：

```
\makeatletter
\def\tagform@#1{\maketag@@@{<#1>}}
\makeatother
```

为了演示，我们只在文档中插入此代码，于是上一个方程的编号效果就会变成如下（必须要使用 `\tag` 命令）：

$$A = C - B \tag{<自定义方程编号>}$$

要返回原来的编号，只需要插入如下代码：

```
\makeatletter
\@addtoreset{equation}{section}
% 或者 \@addtoreset{equation}{part}
\makeatother
% 由于使用 \@ 内部命令符号，所以必须把使用 \makeatletter 和 \makeatother, 而且还要使用 remreset 宏包
```

<sup>11</sup>参考 L<sup>A</sup>T<sub>E</sub>X 数学英文帮助的 19.4 部分

## 6.4 数学公式的对齐方式

在  $\text{L}_\text{X}$  中, 有 5 种内置的 AMS 公式环境供使用, 如果要指定所有数学公式在文档中的缩进可以在导言中插入以下代码:

```
% 制定数学公式缩进为 10mm
\setlength{\mathindent}{10mm}
```

### 6.4.1 AMS align 环境

当选择了“插入”▷“公式”▷“AMS align 环境”时, 将会显示列对齐方式的公式: 奇数列向右对齐, 偶数列向左对齐 (简称“奇右偶左”)。如果公式较长可以使用快捷键“Ctrl+Enter”添加新行 (换行), 例如在第一列中我们输入  $a = b$ , 第二列中我们输入  $c = d$  (因为总共只有一行两列), 显示效果如下:

$$a = bc = d \quad (11)$$

效果看起来几乎两列之间没有间距调整。

注意到, 插入“AMS align 环境”后的默认环境是一行两列, 可以通过数学工具栏<sup>12</sup>上的“添加行”, “添加列”等按钮进行编辑。如果在这个一行两列的默认环境成编辑数学公式所显示出来的效果是比较差的, 必须要添加新列来调整, 比如我们把上面的公式稍微再调整一下, 添加一列, 也就是说一行三列, 依次输入  $a = b$ ,  $c$ ,  $= d$ , 第二列中我们输入, 第三列中输入, 显示效果如下:

$$a = bc \qquad = d \quad (12)$$

根据此环境对齐的规则,  $bc$  应该是挨着, 因为  $c$  在偶数列 (向左靠齐),  $= d$  在奇数列 (向右靠齐) 应该在最右边。与方程 11 一样,  $c$  与  $= d$  的间距特别大, 似乎被连续的空白给填充了 ( $\backslash\text{hfill}$ ), 我们再继续修改, 把第二列留空, 在第三列输入  $c = d$ , 显示效果变成:

$$a = b \qquad c = d \quad (13)$$

最后我们把 11 修改成一行四列, 依次输入  $a, = b, c, = d$  这是个数学符号, 效果如下:

<sup>12</sup>如果数学工具条未显示, 可以点击“视图”▷“工具条”▷“数学 (自动)”打钩

$$a = b \qquad c = d \qquad (14)$$

最后我们发现<sup>13</sup>和<sup>14</sup>的显示效果几乎一样，因为标准的 `align` 环境会默认在每个水平列后自动添加一些空白而使之自动对齐。

#### 6.4.2 AMS `alignat()` 环境

此对齐方式与上面的<sup>6.4.1</sup>类似，只不过可以自己制定列间的距离（使用`\hspace` 命令<sup>13</sup>），比如我们修改<sup>14</sup>如下效果（在 $b$ 的后面键入`\hspace` 双击会自动显示对话框<sup>14</sup>，比如我们设置间距为 1cm）：

$$a = b \qquad c = d \qquad (15)$$

对于多行的公式也有效，比如：

$$a = b \qquad c = d \qquad (16)$$

$$e = f \qquad g = h \qquad (17)$$

AMS `alignat` 环境和标准的 AMS `align` 环境一样，也可以给每行进行编号。

#### 6.4.3 AMS `falign` 环境

在此环境中，前两列会被尽可能的安排在最左边，而最后两列则尽可能的被安排在最右边，同样，例如下面一个两行四列的 `falign` 环境中，列 $a = b$ 和 $c = d$ 被放置在左边，而列 $e = f$ 和 $g = h$ 被放置在右边：

$$a = b \qquad e = f \qquad (18)$$

$$c = d \qquad g = h \qquad (19)$$

同样，在 AMS `falign` 环境中，可以对每行进行编号。

<sup>13</sup>点击“插入”▷“Horizontal space”（水平连续空格距离）来设置

<sup>14</sup>通过菜单命令插入 Horizontal space 无效，可能是 LyX1.6 的一个 bug

#### 6.4.4 AMS gather 环境

在此环境中，只有一列，而且被放置在中间 (不能够添加新列，只能够添加新行)，例如：

$$a = b \quad (20)$$

$$c = d \quad (21)$$

$$z^2 = x^2 + y^2 \quad (22)$$

其中，每行都可以被编号。

#### 6.4.5 ANS Multiline 环境

首先确保你在文档选项中打钩导入了 esint 包。此环境类似于 AMS gather 环境，也只有一列，只是第一行左置，最后一行右置，其余的行放在中间，比如：

$$\begin{aligned} & a^2 + b^2 + 2ab + x^2 + 2x + 1 + y^2 + 2y + 1 + z^3 \\ & \quad + 3z + 3z^2 + 1 + P + Q + R + S \\ & \quad = (a + b)^2 + (x + 1)^2 + (z + 1)^3 + P + Q + R + S \end{aligned} \quad (23)$$

在此环境下，不能对每行进行编号，只能对整个公式编号；如果为了使公式更加美观，我们可以使用命令 `\setlength` 来改变公式首行与页面左边距的距离，使用命令 `\shoveright` 或者 `\shoveleft` 使公式中间一些行向右或向左靠齐，

```
% 控制公式首行缩进
\setlength{\multlinegap}{4cm}
% 向左靠齐
\shoveright{ 内容}
% 向右靠齐
\shoveleft{ 内容}
```

比如我们改变公式23，让首行缩进 1cm，并在公式的第二行最左边插入代码 `\shoveleft`，让第二行向左对齐：

$$\begin{aligned} & a^2 + b^2 + 2ab + x^2 + 2x + 1 + y^2 + 2y + 1 + z^3 \\ & \quad + 3z + 3z^2 + 1 + P + Q + R + S \\ & \quad = (a + b)^2 + (x + 1)^2 + (z + 1)^3 + P + Q + R + S \end{aligned} \quad (24)$$

### 6.4.6 Eqarray 环境

在 Eqarray 环境，第一输入框的内容会向右靠齐，而最后一个输入框的内容会向左靠齐，中间输入框的内容字体比左右框的稍微小一些（在 L<sup>A</sup>T<sub>E</sub>X 编辑环境中不会显示），而且会被自动的安排在中间。在此环境中，只有三列，而且只能添加新行，例如：

$$\begin{array}{ccc} \frac{a}{b} & \frac{a}{b} & \frac{a}{b} \\ abc & abc & abc \end{array}$$

## 6.5 数学长公式输入

对于比较长的数学公式，我们可以使用 ANS Multiline 环境，见 6.4.5，此外，我们还可以通过利用 Eqarray 环境输入长公式，例如：

$$\begin{aligned} f(a, b, x, y, P, Q, R, S) &= a^2 + b^2 + 2ab + x^2 + 2x + 1 + y^2 + 2y + 1 + z^3 \\ &\quad + 3z + 3z^2 + 1 + P + Q + R + S \\ &= (a + b)^2 + (x + 1)^2 + (z + 1)^3 + P + Q + R + S \end{aligned} \quad (25)$$

比较遗憾的是，上面公式的第二行开头的+和3z之间似乎没有空格（几乎挨着），于是我们必须在它们之间手动插入\hspace，然后设置值为 3pt，效果如下：

$$\begin{aligned} f(a, b, x, y, P, Q, R, S) &= a^2 + b^2 + 2ab + x^2 + 2x + 1 + y^2 + 2y + 1 + z^3 \\ &\quad + 3z + 3z^2 + 1 + P + Q + R + S \\ &= (a + b)^2 + (x + 1)^2 + (z + 1)^3 + P + Q + R + S \end{aligned} \quad (26)$$

如果我们对第二行和第三行的缩进有要求，可以在行首使用\lefteqn<sup>15</sup>命令，此命令会自动缩进后面每行列中延长的公式部分，比如：

$$\begin{aligned} \lefteqn{f(a, b, x, y, P, Q, R, S)} &= a^2 + b^2 + 2ab + x^2 + 2x + 1 + y^2 + 2y + 1 + z^3 \\ &\quad + 3z + 3z^2 + 1 + P + Q + R + S \\ &= (a + b)^2 + (x + 1)^2 + (z + 1)^3 + P + Q + R + S \end{aligned} \quad (27)$$

结果是后面的延续（第二行和第三行）被自动缩进了（与公式 26 比较）。

## 6.6 长公式中的括号

有时候，比如我们输入以下长公式：

<sup>15</sup>使用快捷键“Ctrl+L”，然后键入\lefteqn，会出现粉红色的小框，在此框内输入第一行的内容

$$f(n) = n^5 + n^3 + \left[ \sum_{i=1}^n i + \dots \right. \\ \left. \dots + n + 1 \right] \quad (28)$$

方程28的第二行右括号与第一行左括号明显不匹配，为了使之匹配，我们在第二行的 `\left(` (在  $\text{L}_\text{YX}$  编辑器中显示的是一条竖直的虚线) 后添加以下代码:

```
\vphantom{\sum_{i=1}^n i}
```

其实我们没有必要这样输入，因为比较繁琐，最简单的步骤如下:

1. 手动在 `\left(` (虚线) 后输入 `\vphantom` (不用全部输入) 然后会自动出现对话框，让你选择 `\vphantom` 命令，之后会出现一个红色下划线的对话框
2. 然后我们只需要复制与左括号最匹配的数学式子 (这里显然是  $\sum_{i=1}^n i$ )，然后直接粘贴在第一步中的对话框中即可

$$f(n) = n^5 + n^3 + \left[ \sum_{i=1}^n i + \dots \right. \\ \left. \dots + n + 1 \right] \quad (29)$$

## 6.7 数学特殊字符的输入

数学字符可以通过“插入”▷“特殊字符”▷... 来选择，有任何问题可以参考  $\text{L}_\text{YX}$  自带的数学用户手册。

## 6.8 数学脚本的使用

$\text{L}_\text{YX}$  中一个非常强大的数学输入功能就是使用数学输入脚本，当然你可以使用 `\newcommand` 命令来代替，但是使用 `\newcommand` 命令又要撰写大量的  $\text{T}_\text{E}\text{X}$  代码，很浪费时间，而且对于复杂的公式，很容易混淆参数位置而出错 (因为没有直观的图形显示)，比如我们要求解标准一元三次方程的根:

$$x^3 = ax^2 + bx + c \quad (30)$$

那么它的标准解为:

$$\begin{aligned}
 x = & \\
 & -\frac{b}{3a} + \sqrt[3]{-\frac{27a^2 - 9abc + 2b^3}{54a^3}} + \sqrt{\left(\frac{27a^2 - 9abc + 2b^3}{54a^3}\right)^2 + \left(\frac{3ac - b^2}{9a^2}\right)^3} \\
 & + \sqrt[3]{-\frac{27a^2 - 9abc + 2b^3}{54a^3}} - \sqrt{\left(\frac{27a^2 - 9abc + 2b^3}{54a^3}\right)^2 + \left(\frac{3ac - b^2}{9a^2}\right)^3} \quad (31)
 \end{aligned}$$

如果文档中有很多地方要求解, 需要手动输入这样的公式, 要浪费大量的时间, 因此我们可以使用 L<sup>A</sup>T<sub>E</sub>X 的数学脚本来简化我们的输入, 通过“插入”▷“公式”▷“Macro”来编辑我们的数学脚本 (也可以称之为模版, 注意不要插入过于长的公式, 会导致无法编辑<sup>16</sup>)。

然后我们通过在数学环境中直接键入 \rootpartOne 和 \rootpartTwo 来引用这两个数学脚本 (在键入 \rootpart 的时候, L<sup>A</sup>T<sub>E</sub>X 会自动显示 \rootpartOne 和 \rootpartTwo 两个脚本命令, 因而我们可以很轻易的选择所需要的脚本模版), 比如我们有另外一个一元三次方程,

$$x^3 = px^2 + qx + r \quad (32)$$

那么求解过程为:

$$\begin{aligned}
 x = & \\
 & -\frac{q}{3p} + \sqrt[3]{-\frac{27p^2 - 9pqr + 2q^3}{54p^3}} + \sqrt{\left(\frac{27p^2 - 9pqr + 2q^3}{54p^3}\right)^2 + \left(\frac{3pr - q^2}{9p^2}\right)^3} \\
 & + \sqrt[3]{-\frac{27p^2 - 9pqr + 2q^3}{54p^3}} - \sqrt{\left(\frac{27p^2 - 9pqr + 2q^3}{54p^3}\right)^2 + \left(\frac{3pr - q^2}{9p^2}\right)^3} \quad (33)
 \end{aligned}$$

## 7 其他排版技巧

<sup>16</sup>L<sup>A</sup>T<sub>E</sub>X1.6 中还不支持复杂数学公式的脚本, 比如很长的公式, 会无法折行