

LibreOffice Math 3.5.x 导引

版权声明

本文档版权(2011~2012)由下列贡献者所有。您可以在 GNU 通用公共许可协议版本 3 或更新版本，或是知识共享署名许可协议版本 3.0 或更新版本所允许的范围内对本文档进行分发或修改。本手册中的所有商标属于它们的合法拥有者。

贡献者

Jean Hollis Weber

Hazel Russman

Laurent Balland-Poirier

汉化

Ran Liang, Ph.D.

rliang06@ruc.edu.cn

LibreOffice Math 3.5.x 导引

0. 什么是 LibreOffice 和 LibreOffice Math

LibreOffice 是一种可自由获得的、全功能的办公套件，它可以运行在 Windows、Linux、Mac OS X 等操作系统上。LibreOffice 已被世界上部分地区的教育、行政、商务部门以及个人用户接受并使用。

LibreOffice Math 是一个简易的公式编辑器，它使您能够以标准格式快速创建、编排并显示数学、化学、电子及其他自然科学的公式和方程式。Math 最常用于 Writer 文字处理程序的文档中，也可用于其他类型的文档或单独使用。

如果您从未使用过 LibreOffice，或者您需要查阅关于 LibreOffice 全部组件的总体介绍，您可能需要首先查阅 LibreOffice 入门教程：*Getting Started with LibreOffice*。

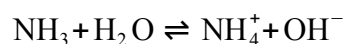
1. Math 简介

Math 是用于创建数学和化学公式或方程式的 LibreOffice 组件。它最常用作文本文档中的公式编辑器，也可用于其他类型的文档或单独使用。当您在 Writer 文字处理软件中调用 Math 时，您所创建的公式将被视为嵌入文档中的对象处理。

Math 公式编辑器用于以符号形式创建如下所示的公式。如果您需要进行数值评估，请参阅 Calc 导引。

公式示例：

$$\frac{d f(x)}{d x} = \ln(x) + \tan^{-1}(x^2)$$



2. 开始使用 Math

您可以作为独立文档创建一个公式，也可将公式嵌入 Writer 文本文档或其他 LibreOffice 组件的文档中。

2.1 作为独立的文档创建公式或方程式

如需作为独立的文档创建公式或方程式，可以选用以下方式之一打开 LibreOffice Math 组件，即可出现空白的公式文档界面(图 1)：

- 在 LibreOffice 菜单栏上选择“文件→新建→公式”
- 在“标准”工具栏上，点击“新建”图标右侧的三角形，从下拉菜单中选择“公式”
- 在 LibreOffice 启动中心界面点击“公式”

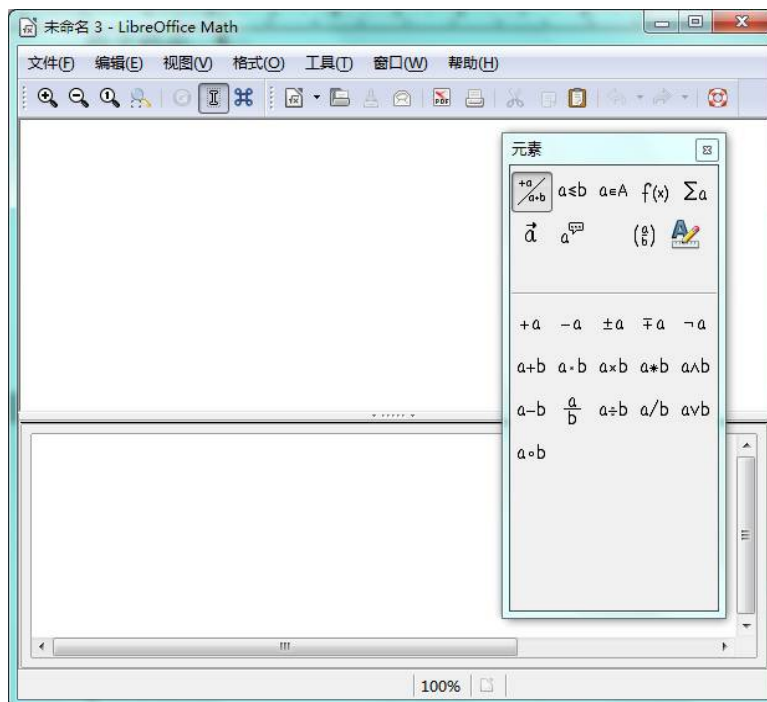


图 1 空白公式文档界面

以上界面的上半部分窗口为公式预览窗口，您所输入的公式内容将会即时显示在此区域。下半部分窗口为公式编辑区域，您可以在该区域输入公式的标记语言(markup language)代码。同时在 Math 界面中还有一个浮动的元素窗口。

2.2 向 Writer 文本文档中插入公式

如果需要向 Writer 文本文档中插入公式，您需要首先打开该文本档，选择菜单栏中的“插入→对象→公式”，公式编辑器将会出现在 Writer 窗口下方，同时出现浮动的元素窗口。在文本文档中还可以看到一个带有灰色边框的对象框，您所插入的公式将会显示于此(图2)。

提示：如果元素窗口没有出现，您可以在 Math 的菜单栏选择“视图→元素”。

当您完成公式输入时，您可以选择以下方式之一退出公式编辑器，返回文档编辑界面：按下 Esc 键或点击主文档中任何公式以外的区域。双击公式对象将会重新打开公式编辑器，您可以继续编辑或修改公式。

在 LibreOffice 文档中，公式被视为 OLE (object linking and embedding)对象插入文档中。对于 Writer 文档，公式被视为单个字符锁定(anchor)在文档中，可看做嵌入在运行文字(running text)中。同其他类型的 OLE 对象相同，您可以改变公式的锁定方式，使其成为浮动的 OLE 对象。在 Calc、Impress 和 Draw 等组件中，公式作为浮动 OLE 对象嵌入在文档中。

如果您在编辑文本文档过程中经常需要插入公式，您可能希望将“公式”按钮添加至“标准”工具栏或创建“插入公式”快捷键，请查阅“添加按钮至工具栏”或“添加快捷键”。(9.2~9.3 节)

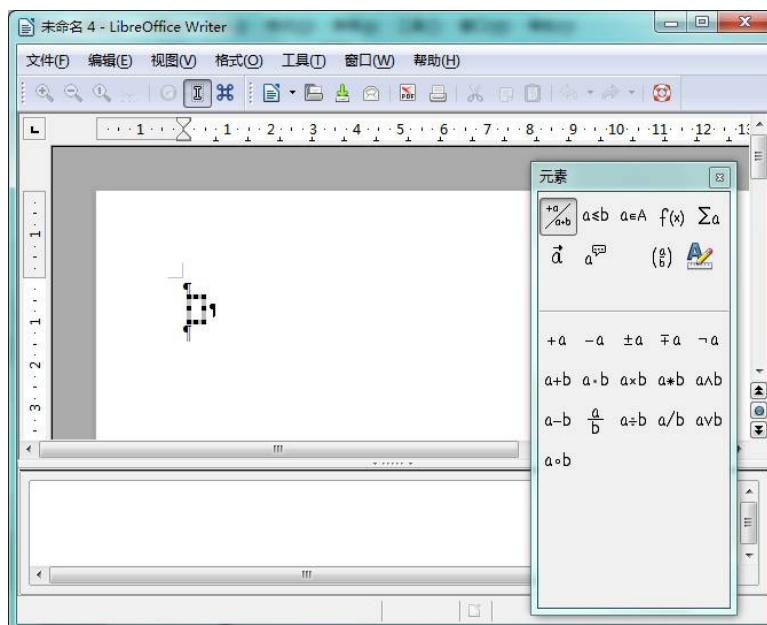


图 2 向 Writer 文档中插入公式

3. 输入公式

Math 公式编辑器采用一种标记语言来表达公式。例如，代码"%beta"将会创建希腊字母 β 。这种标记语言被设计成与公式的英语读法尽可能相似。例如，"a over b"将会创建一个分式： $\frac{a}{b}$ 。

您可用以下方式之一输入公式：

- 从元素窗口中选择所需符号
- 右键点击公式编辑器，从弹出的上下文菜单中选择所需符号
- 直接使用标记语言代码在公式编辑器中输入公式

使用上下文菜单或元素窗口将会在公式编辑器中插入对应于所选符号的标记语言代码，这提供了一种学习 LibreOffice Math 的标记语言的简便方式。

3.1 元素窗口

输入公式的最简便的方法是使用元素窗口(图 3)。

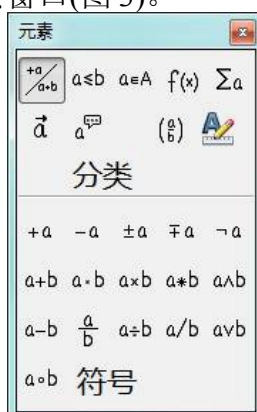


图 3 元素窗口示意，符号被分类显示

元素窗口分为两个主要部分：上半部分为符号分类，通过点击此部分的图标以指定下半部分出现的符号列表；下半部分为当前分类下的符号列表。

提示：您可以通过选择菜单栏中的“视图→元素”来显示或隐藏元素窗口。

示例 1：输入公式 5×4 。

在本例中，我们将输入一个简单的公式： 5×4 ，在元素窗口中执行如下操作：

1. 选择分类列表中的左上角图标；
2. 选择乘法符号(图 4)。

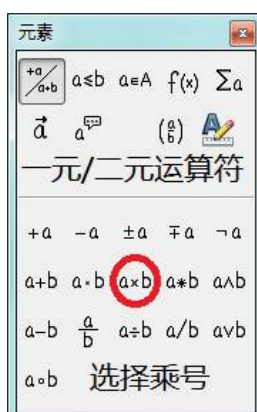


图 4 在元素窗口中选择乘号

当您在元素窗口中点击乘号图标时，将会发生两个变化(图 5)：

- 公式编辑器中出现标记语言代码："`<?> times <?>`";
- 主文本中出现如下的灰色公式框： $\square \times \square$ 。

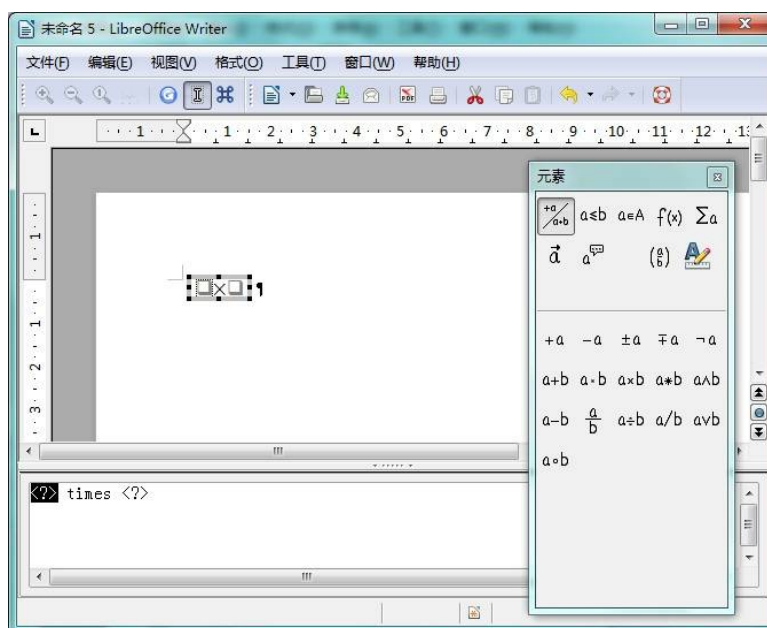


图 5 在元素窗口中选择乘号出现的结果

图 5 所示"`<?>`"符号为占位符(placeholder)，您可以用其他内容替代，例如本例中的 5 和 4。当您更改其内容时，相应的公式显示将会自动更新，您将看到如图 6 所示的结果：

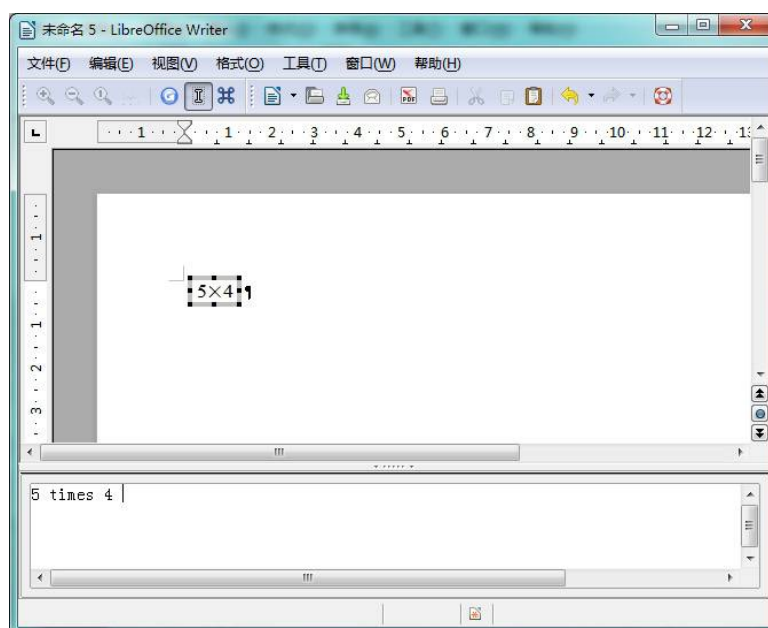


图 6 在乘号两边分别输入 5 和 4 得到的结果

提示：当您向文档中添加公式时，保留的占位符将会以方框显示于公式主体中，并且以符号"<?>"显示在标记语言编辑窗口中。您可以通过 F4 和 Shift+F4 按键导航公式中的各个占位符。

提示：如果您不希望公式显示自动更新，您可以从菜单栏中取消“视图→自动更新视图”选项。如果需要手动更新视图，您可以按下 F9 键或选择“视图→更新”。

3.2 右键上下文菜单

右键点击公式编辑器，将会弹出上下文菜单，这是使用数学符号的另一种方式。弹出的上下文菜单如图 7 所示。上下文菜单中列出的符号与元素窗口中的符号对应，并且还带有一些额外的命令。

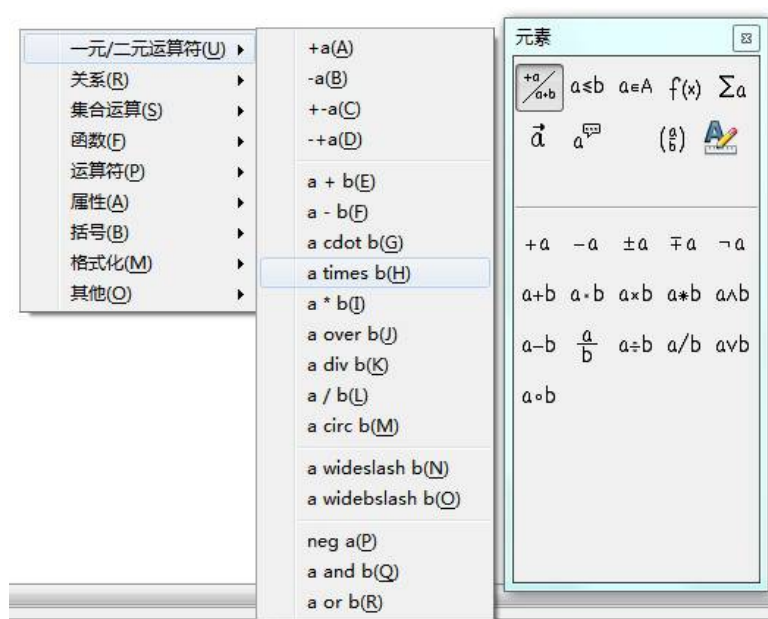


图 7 利用上下文菜单插入乘法符号

提示：元素窗口和右键上下文菜单都不包括所有可用的符号或命令。对于较少使用的符号，您需要通过输入标记语言代码来实现。完整的命令代码列表可以在本手册附录中找到。

3.3 标记语言

您可以直接在公式编辑器中输入标记语言。例如，您可以输入"5 times 4"直接得到 5×4 。如果您熟悉标记语言代码，它将成为最为快捷的公式编辑方式。

下表中列出了常用的公式符号及其对应的标记语言代码。

显示	命令	显示	命令
$a=b$	a=b	\sqrt{a}	sqrt {a}
a^2	a^2	a_n	a_n
$\int f(x)dx$	int f(x) nitalic {d} x	$\sum a_n$	sum a_n
$a \leq b$	a<=b	∞	infinity
$a \times b$	a times b	$x \cdot y$	x cdot y

3.4 希腊字母

希腊字母(α 、 β 、 γ 、 δ 等等)是数学公式中的常用符号，这些符号并不包括在元素窗口或右键上下文菜单中。不过，希腊字母的标记语言代码相对简单，只需输入%符号加上对应希腊字母的英文名称。

- 欲输入小写希腊字母，以小写字母输入该希腊字母的英文名称；
- 欲输入大写希腊字母，以大写字母输入该希腊字母的英文名称；
- 欲输入斜体希腊字母，在%和希腊字母名称之间添加 i。

完整的希腊字母列表可以在本手册附录中找到。

另一种输入希腊字母的方法是使用图标窗口。选择“工具→图标”，出现的窗口如图 8 所示。在“图标组”下拉菜单中选择“希腊语”，然后双击列表中出现的希腊字母，即可实现向公式中插入希腊字母。同时，该字母对应的标记语言显示于列表下方。

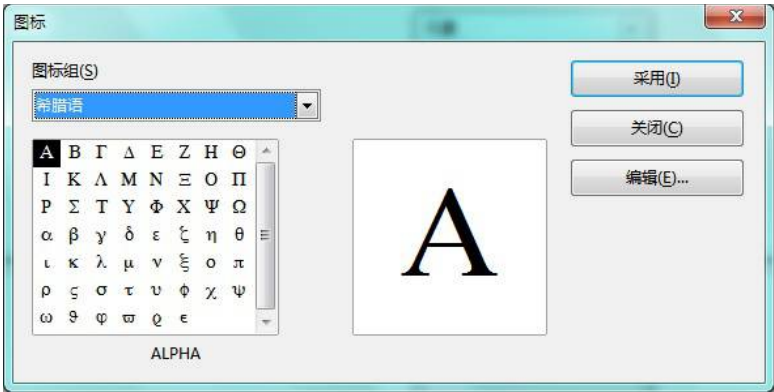


图 8 用于插入希腊字母及其他特殊符号的图标窗口

例 2 输入公式 $\pi \approx 3.14159$

在本例中，假设：

- 您想要输入上述公式(圆周率保留小数点后 5 位的近似值)；
- 您知道希腊字母 π 的英文名称 **pi**；
- 您不知道约等号 \approx 的标记语言代码。

您可以按照以下步骤输入此公式：

1. 输入%紧跟文字 **pi**，即"%pi"，得到希腊字母 π ；
2. 打开元素窗口(视图→元素)；
3. “约等于”是一种关系运算符。点击“关系”分类按钮，如果您将鼠标光标悬在按钮上方，您将看到工具提示“关系”(图 9)。图 10 显示按下“关系”按钮后的元素窗口效果，您所需的关系运算符位于圈中。
4. 选择 $a \approx b$ 按钮，公式编辑器中将会出现对应标记语言：`%pi <?> approx <?>`。
5. 删除第一个"%pi"占位符，然后按 F4 切换至公式末尾的占位符，输入 3.14159，您将得到最终的标记语言"`%pi approx 3.14159`"，如图 11 所示。

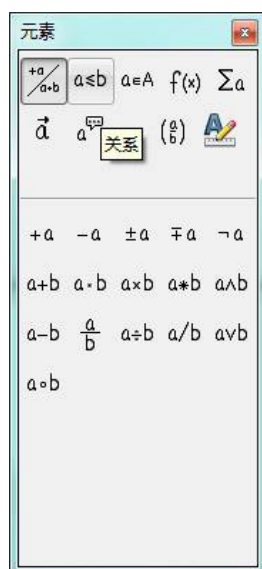


图 9 工具提示“关系”

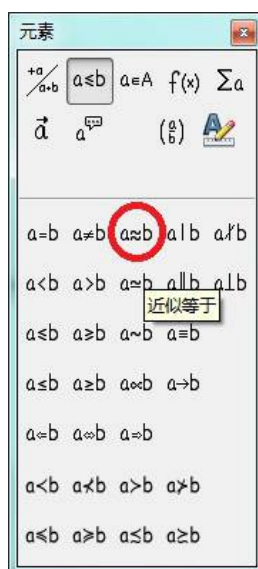


图 10 选择“关系”之后

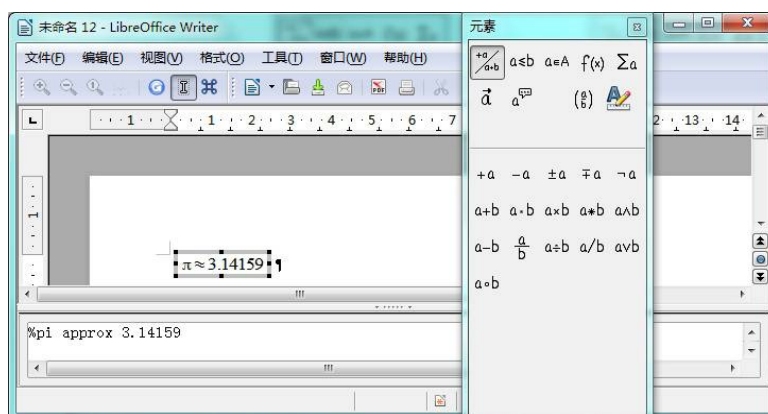


图 11 最终效果

4. 修改公式

您可以在任何时间对公式进行修改。如需进入公式编辑模式，您可以双击文档中的公式对象。

4.1 选择修改区域

如需选择标记语言中的适当区域进行修改，可以采取以下方式之一：

- 在公式编辑器中，点击欲修改的位置；
- 拖动鼠标选取标记语言中欲修改的位置；
- 点击预览区域中的符号元素，光标将会自动移至公式编辑器中标记语言的对应位置；
- 双击预览区域中的符号元素，公式编辑器中标记语言的对应位置会被选中。

如果您需要在独立的 Math 窗口中对预览区域进行操作，您需要启用“公式光标”，可以在 Math 工具栏中找到相关按钮。

4.2 执行修改

您可以通过重写已选定的标记语言代码或在光标位置插入新的标记语言代码的方式来对公式进行修改。

提示：LibreOffice Math 还提供直接在预览区域进行公式修改的功能，此功能尚在测试中。如需使用该项功能，您需要进行以下设置：选择“工具→选项→LibreOffice→一般”，选中“启用实验性(不稳定的)功能”复选框。如果您启用此设置，您将不能通过点击预览区域符号元素以选定对应标记语言代码。

5. 公式布局

使用 LibreOffice Math 最为困难之处在于创建复杂的数学公式。这部分内容可能会为您提供一些建议。

5.1 善于使用括号

除了通过识别括号，Math 不会自动识别运算符的优先级。您需要通过使用括号来明确声明运算符的优先级。考虑以下范例：

标记语言	结果	标记语言	结果
2 over x+1	$\frac{2}{x}+1$	2 over {x+1}	$\frac{2}{x+1}$
-1 over 2	$\frac{-1}{2}$	-{1 over 2}	$-\frac{1}{2}$

在第一个例子中，Math 将"over"前面的 2 和后面的 x 视为分式的组成部分，并基于此给出公式显示。如果您需要使 $x+1$ 作为分母而不是 x ，您需要使用括号将 $x+1$ 组合在一起，使其整体作为分母。在第二个例子中，Math 将负号视为 1 的前缀，而将负号置于分子。如需表示整个分式的负值，您需要将分式整体置于括号内，Math 才会将分式作为整体处理。

提示：括号字符 "{" 和 "}" 仅作为公式标记语言的结构标识符，并不会被显示在公式中。如果您需要在公式中显示 "{" 和 "}"，您需要使用命令 lbrace 和 rbrace，考虑以下范例：

标记语言	结果	标记语言	结果
<code>x over {-x+1}</code>	$\frac{x}{x+1}$	<code>x over lbrace -x+1 rbrace</code>	$\frac{x}{\{-x+1\}}$

5.2 在矩阵中使用括号

作为背景，首先介绍矩阵的标记语言代码。

标记语言	结果
<code>matrix {a # b ## c # d}</code>	$\begin{matrix} a & b \\ c & d \end{matrix}$

提示：矩阵的两行由两个#号分隔，不同列的元素由一个#号分隔。

第一个常见的问题是，括号不随着矩阵的维度改变而改变大小。

标记语言	结果
<code>(matrix {a # b ## c # d})</code>	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Math 提供可变尺度的括号，即括号会根据其中内容自动调整尺度以适应内容大小。您可以使用命令 `left`(和 `right`)来输入可变尺度的括号。

标记语言	结果
<code>left(matrix {a # b ## c # d} right)</code>	$\left(\begin{matrix} a & b \\ c & d \end{matrix} \right)$

提示：您可以使用`left[`和`right]`输入可变尺度的方括号，可用的括号代码可以在本手册附录中找到。

如果您需要使所有括号均为可变尺度，您可以选择“格式→间隔”，在“分类”下拉菜单中选择“括号”，然后选中“所有括号使用显示比例”复选框。

除矩阵外，可变尺度的括号还可应用于分式、根式等结构。

5.3 孤立与未成对的括号

一般情况下，**Math** 期望对于每一个左括号都存在与之成对的右括号。如果您遗忘了半对括号，**Math** 会在相应位置显示一个倒置的问号。当所有括号都成对时，这些错误符号将会消失。有时，遗忘半对括号将会使得公式结构不整。然而，在某些特殊情况下，需要使用未成对的括号。在此情况下，您可以采取以下方式之一：

- 对于固定尺度的括号，在括号前添加反斜线“\”，以表示与之相邻的字符不被识别为运算优先级标识符，而仅作为普通字符处理。例如，半开区间 $[a; b]$ 可表示为“`[a;b\[`”，与“`[a;b[`”对比；
- 可变尺度的括号也可不成对，同样的半开区间可表示为“`left [a;b right [`”。

对于可变符号的括号，还可以使用命令 `none` 表示与前面的括号成对但不需显示的括号。

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

以上公式可表示为：

```
abs x=left lbrace stack {x "for" x>=0 # -x "for" x<0}right none
```

5.4 Math 中函数的识别

在基本配置下，**Math** 会将变量显示为斜体。如果您输入一个函数，**Math** 一般会自动识别，并以正体显示函数名，**Math** 内置的函数列表可以在本手册附录中找到。如果 **Math** 没有自动识别您输入的函数，您可以通知 **Math** 您所输入的内容为函数而非变量。在函数名称前加上 `func`，紧邻的字符将被识别为函数。

某些函数需要跟随数字或变量作为参数，如果这些必需的参数缺失，**Math** 将会在缺失的参数位置显示一个倒置的问号。您可以通过补全所需参数或放置空的 `{}` 括号作为占位符的方式消除此错误。

提示：您可以通过 F3 和 Shift+F3 来切换错误信息。

5.5 跨行公式

假设您需要输入一个简单的跨行公式：

```
x=3
y=1
```

您的第一反应可能是简单地通过 `Enter` 键输入换行符。然而，如果您简单地按下 `Enter` 键，标记语言代码转入下一行，而预览区的公式并不断行。您需要具体指定 `newline` 命令，如下表所示：

标记语言	结果	标记语言	结果
x=3 y=1	$x=3\ y=1$	x=3 newline y=1	$x=3$ $y=1$

一般情况下，您不能直接利用 **Math** 进行脱式计算公式的书写，而不编辑完整的新公式。由于等号是二元运算符，**Math** 将会期待等号左侧的项。您可以采取以下方式之一(在等号左侧添加以下符号之一)：

- 空引号：`"`。这将使得下一行公式自动左对齐；
- 空括号：`{}`。下一行公式将会居中对齐；
- 间隔符：`"~"`或`"^"`。下一行公式将会居中，并带有对应的间隔。

对应的等号对齐状况参见 5.10 节。

同样，公式中的元素之间的间隔不是通过空格字符实现的，您需要在标记语言中使用间隔符：`"^"`表示较小的间隔；`"~"`表示较大的间隔。另一种方式是在引号中添加空格字符，它将被视为普通文本。位于公式行末尾的间隔符将被略过(参见 9.6 节)。

5.6 为积分/求和符号添加上下限

sum 和 int 命令带有可选参数 from 和 to，分别表示求和/积分的下限和上限，这些参数可以单独或联合使用。

标记语言	结果
sum from k=1 to n a_k	$\sum_{k=1}^n a_k$
int from 0 to x f(t) nitalic {d}t	$\int_0^x f(t) \mathrm{d}t$
int_0^x f(t) nitalic {d}t	$\int_0^x f(t) \mathrm{d}t$
int from Re f	$\int_{\Re} f$
sum to infinity 2^{-n}	$\sum 2^{-n}$

5.7 书写微分式

在 Math 中书写微分式如同书写普通分式，即使用 over 命令，将 over 与全微分算符 d (在数学期刊或教材中一般使用正体，藉此和变量区分开来)或偏微分算符 ∂ (partial 命令)相结合。

标记语言	代码
{nitalic {d}y} over {nitalic {d}x}	$\frac{\mathrm{d}y}{\mathrm{d}x}$
{partial f} over {partial y}	$\frac{\partial f}{\partial y}$
{partial^2 f} over {partial t^2}	$\frac{\partial^2 f}{\partial t^2}$

如果需要使用撇号"'"书写微分式，您首先需要向图标列表中添加撇号(参见 9.4 节)。直接使用引号代替撇号是不美观并且不规范的。

5.8 作为普通字符的标记字符

用于控制标记语言结构的字符一般不可直接作为普通字符输入，存在此类问题的符号包括：%、{、}、&、|、_、^和"。例如，您不能通过直接输入 2%=0.02 或 1"=2.56 cm。有两种方式可用于克服这种限制：

- 用双引号将特殊字符标记为普通字符，例如：2""=0.02。此法不适用于双引号本身。
- 将所需符号添加至图标列表(参见 9.4 节)。

在某些情况下，您可以使用命令：

- 输入 lbrace 和 rbrace 可得到 {}。
- 输入 mline 可得到竖线。例如，输入"2 mline 4"可得到 2|4。

像 HTML 语言那样转换字符实体(character entity)或使用 Esc 字符在 Math 中不可实现。

5.9 公式中的文本

如果您需要向公式中插入文本，可以将文本输入在直双引号中(参见 5.3 节绝对值定义式)。除了双引号本身以外的任何字符都可以出现在文本中，而目前尚无法通过特殊字符对话框输入双引号。如果确实需要在公式中使用双引号，您可以将文本输入到文字处理软件中，然后通过剪切板粘贴到 Math 中，通过这种方式您可以将直双引号转换为分左右的双引号，从而实现向 Math 中输入“智能引号”，如图 12 所示：

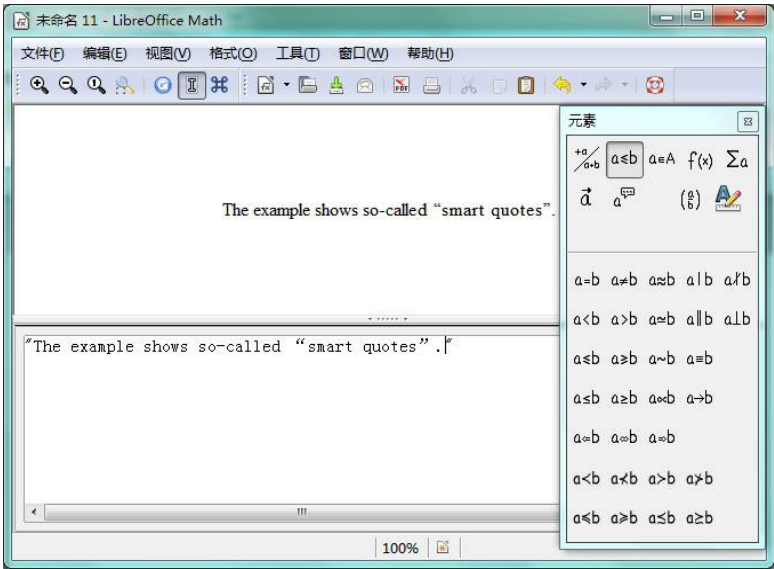


图 12 通过从 Writer 复制粘贴实现输入智能引号

公式中的字体将按照“字体”对话框中的“文字”列表框所指定的字体显示(参见 6.2 节)。如需使用公式编辑器下方窗口中的字体，在文本之前设置属性 Sans、Serif 或 Fixed。

在默认条件下，文本采用左对齐方式，您可以使用 alignc 或 alignr 改变文字对齐方式。文本中的命令将不会被表达，如果您需要在文本中间使用格式化命令，需要用引号将文本断开。例如：

"In " color blue bold "isosceles" " triangles, the base angles are equal."

将会得到：

In **isosceles** triangles, the base angles are equal.

5.10 将公式按照等号位置对齐

Math 并不提供基于某一特定字符对齐的命令，不过您可以通过矩阵来实现：

标记语言	结果
<pre>matrix { alignr x+y # {}={} # alignl 2 ## alignr x # {}={} alignl 2-y }</pre>	$\begin{matrix} x+y & = & 2 \\ x & = & 2-y \end{matrix}$

注意，等号两侧的空括号是必需的，因为等号是二元运算符，两边都需要有参数。您可以将它们替换为间隔符`或~。

您可以通过调整矩阵的列之间的间隔来减少等号两侧的间隔：

1. 在打开公式编辑器的条件下，从菜单栏选择“格式→间隔”；
2. 在“间隔”对话框(图 13)，点击“分类”按钮并从下拉菜单中选择“矩阵”；
3. 在“列间隔”输入 0%，然后点击“确定”。

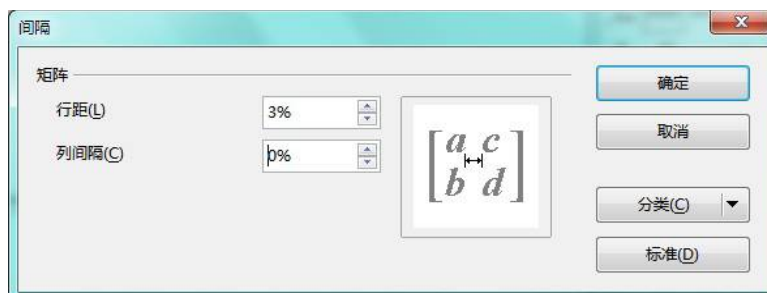


图 13 设定矩阵中的间隔

6. 改变公式外观

6.1 改变字体大小

这个问题是 LibreOffice Math 使用者最常提出的问题之一，答案是简单的，但并不直观。

1. 启动公式编辑器，选择“格式→字体大小”；
2. 在最上边的“基本大小”选项中选择较大的字体(图 14)。



图 14 编辑基本字体大小

结果如图 15 所示：

如果您需要改变公式中的部分字符的大小，可以使用 size 命令，例如：“b size 5 {a}”将会得到 b_a 。在元素窗口中“属性”一类的“更改大小”图标对应于 size 命令。size 命令后面的参数可以是绝对数值或相对于上下文字体大小(默认为基本大小)的相对值，例如 +6、-3、*2 或/2。

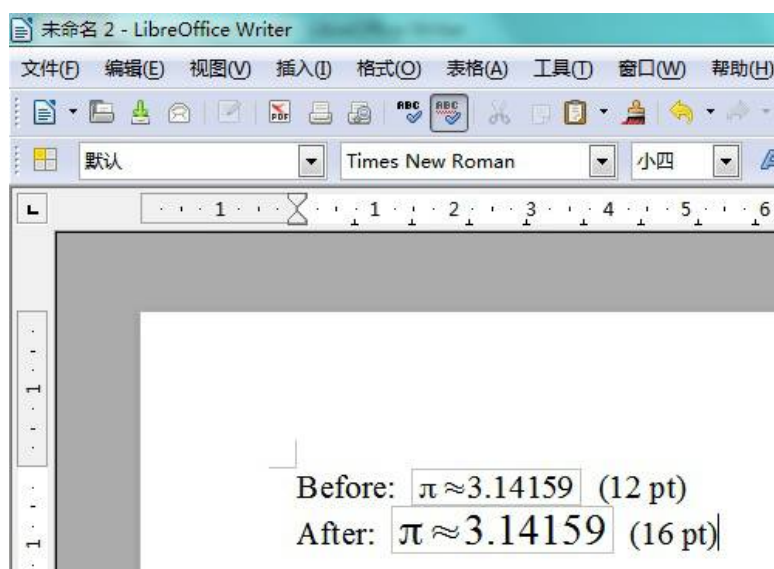


图 15 改变基本字体大小的效果

6.2 改变字体

您可以通过“格式→字体”选项来改变公式中的字体(图 16)。

“字体”对话框中的“公式字体”部分包括四种公式元素(变量、函数、数字和文字)的字体设定，而运算符、关系符和括号的字体不受这些选项影响，它们一般来源于 OpenSymbol 字体。类似地，通过“图标”界面选取插入的符号仍以先前添加符号(参见 9.4 节)时指定的字体显示。

窗口下面的“使用者设定的字体”决定了在标记语言代码中使用属性 font serif、font sans 和 font fixed 时将对应何种字体。

图 16 下方显示的 Serif、Sans、Fixed 分别表示衬线字体、无衬线字体和等宽字体。在 Windows 系统中，最常使用的上述三类字体分别是 Times New Roman、Arial 以及 Courier New，这三种字体也是科技文献写作中最为常用的字体。等宽字体一般用于书写程序代码，便于辨识空格和符号。(译者注)

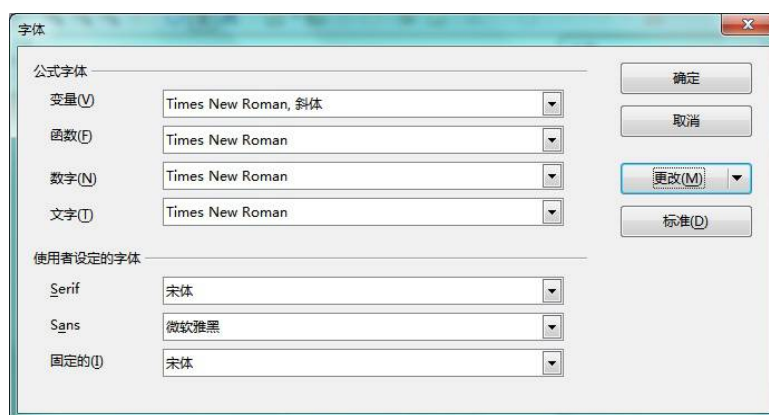


图 16 字体对话框

如果您需要更改字体，您可以点击“更改”，然后在弹出的下拉菜单中选择需要改变字体的项目(变量、函数、数字、文字等)，这将会打开一个选择字体的窗口(图 17)，它将显示您的操作系统中所有可用的字体。



图 17 选择字体的对话框

如果您在窗口最上方的“字体”文本框中输入您所需的字体名称的首字母，字体列表框将跳转至您所输入的首字母的位置。继续输入几个额外的字母，您可以找到您所需使用的字体。如果您不清楚您所需使用的字体的具体名称，您可以拖动列表框右侧的滚动条以遍历所有已安装的字体。点击列表框中的任何字体，窗口下方将会显示该字体的预览效果。

在数学公式中，变量名应使用斜体书写，请确保对应于“变量”的字体属性中的“斜体”复选框被选中。对于公式中的其他元素(函数名、量纲等)，一般应使用正体。您也可以在公式的标记语言中设定字体风格，使用 *italic* 或 **bold** 将会设定斜体或粗体(可同时使用以设置粗斜体，例如向量)，而 *nitalic* 或 **nbold** 将会设定非斜体或非粗体。

当您选择新的字体时，曾经使用过的字体仍然保留在相应元素的“字体”下拉列表中以供重新选择。这种更改设定仅对当前 Math 会话有效，原有的字体设置不会永久保留。

您可以选择任何喜爱的字体，但如果您需要与其他人共享交互文档，您应选择您的同事的计算机系统已安装的字体，以便于显示和进一步编辑。

6.3 更改间隔

如果您需要更改公式中的间隔，可以按以下步骤操作：

1. 选择“格式→间隔”；
2. 在“间隔”对话框(图 18)中，点击“分类”按钮右侧三角形，在下拉菜单中选择欲修改的间隔分类；
3. 选择适当的间隔数值(可修改的间隔参数取决于所选分类)，点击“确定”。

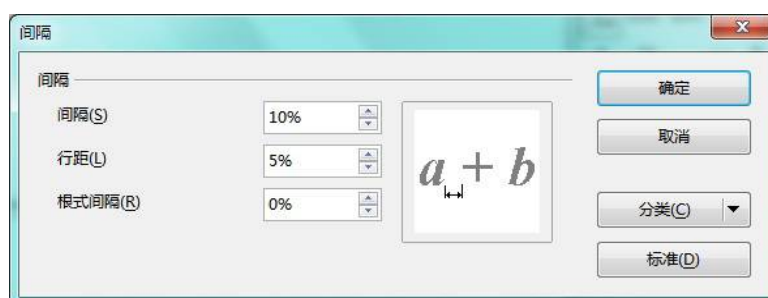


图 18 间隔对话框

各个分类下可修改的间隔参数列于下表：

分类	可能的参数调整			
间隔	字符间隔	行距	根式间隔	
指数	上标		下标	
分式	分子		分母	
分数线	两侧空余长度		分数线宽度	
极限	上限		下限	
括号	上下空余量		与括号内字符的间隔	
矩阵	行距		列间隔	
符号	主要高度		与字符顶端最小间隔	
算符	空余高度		与参数间隔	
边框	左边距	右边距	顶边距	底边距

6.4 改变对齐方式

对齐方式设置决定公式中的元素在上下及左右方向如何与其他元素排布，选择“格式→对齐”，在“对齐”对话框(图 19)中选取在全局基础上的对齐方式：左对齐、居中或右对齐。



图 19 对齐对话框

在此您仍然可以通过点击“标准”按钮将当前更改设为默认属性，这样将会影响到以后创建的任何公式，而限于当前编辑的公式。不同对齐设置下的显示效果如下所示：

对齐方式	范例(分式、矩阵及余弦定理公式)		
左对齐	$\frac{x^2-9}{x}$	$\begin{pmatrix} 100 \\ 7 \end{pmatrix}$	$\begin{aligned} c^2 &= a^2 + b^2 - 2ab \cos \gamma \\ \cos \gamma &= \frac{c^2 - a^2 - b^2}{-2ab} \end{aligned}$
居中(默认)	$\frac{x^2-9}{x}$	$\begin{pmatrix} 100 \\ 7 \end{pmatrix}$	$\begin{aligned} c^2 &= a^2 + b^2 - 2ab \cos \gamma \\ \cos \gamma &= \frac{c^2 - a^2 - b^2}{-2ab} \end{aligned}$
右对齐	$\frac{x^2-9}{x}$	$\begin{pmatrix} 100 \\ 7 \end{pmatrix}$	$\begin{aligned} c^2 &= a^2 + b^2 - 2ab \cos \gamma \\ \cos \gamma &= \frac{c^2 - a^2 - b^2}{-2ab} \end{aligned}$

注意，不可将公式按照某一特定字符对齐，参见 5.10 节范例。

对齐方式的默认设置对文字元素无效，它们总是按照左对齐方式排列。在下例中，默认对齐方式为居中，而第一个例子中的第二行以空字符串开头，因而仍为左对齐。

标记语言	结果
<code>{1+2+3+4} over 5 + 2 over {60+70+80+90} newline ""=2+1 over 150</code>	$\frac{1+2+3+4}{5} + \frac{2}{60+70+80+90}$ $= 2 + \frac{1}{150}$
<code>{1+2+3+4} over 5 + 2 over {60+70+80+90} newline {}=2+1 over 150</code>	$\frac{1+2+3+4}{5} + \frac{2}{60+70+80+90}$ $= 2 + \frac{1}{150}$

与默认设置相独立的是，您可以利用以下命令具体指定对齐方式：`alignl`、`alignc`、`alignr`。这些属性设置命令对文字对象也有效。

6.5 更改字体颜色

您可以使用 `color` 命令来改变公式中的部分文本的颜色，例如标记语言 `color red ABC` 将得到 **ABC**。可选的 8 种颜色代码为：`white` (白色)、`black` (黑色)、`red` (红色)、`green` (绿色)、`blue` (蓝色)、`cyan` (青色)、`magenta` (品红色)、`yellow` (黄色)。

您可以为公式中用括号 `{}` 或其他括号整合起来的部分统一设置颜色，例如标记语言 `A B color green {C D} E` 将得到 **ABCDE**。

如果在一段标记语言中，多种颜色代码被使用，对于给定字段，位于最内层括号的颜色代码将会生效，例如标记语言 `color blue {A B color cyan C D}` 将得到 **ABCD**。

在 **Math** 中，您不能为公式对象指定背景颜色，**Math** 中的公式对象的背景总是无色(透明)的。整个公式的背景效果由它所在的文档或框架(例如在一个 **Writer** 文本文档中)的背景设置决定。例如，在 **Writer** 中，您可以使用对象属性方法(右键→对象)来指定用于整个公式对象的背景色(参见 7.5 节)。

7. Writer 文档中的公式

7.1 公式编号

公式编号是 **Writer** 最为实用的几个隐藏功能之一。插入公式编号的步骤并不复杂，但不易发现。

- 1. 开始新的一行；
- 2. 输入字符 `fn`，然后按下 `F3`。

字符 `fn` 将会被一个带有编号的公式替代，例如：

$$E=mc^2 \quad (1)$$

您可以双击公式进行编辑。

您还可以通过以下步骤引用一个公式：

1. 选择“插入→交叉引用”；
2. 在“交叉引用”对话框中的“引用”选项卡下的“字段类型”一栏中选择“文字”；
3. 在“选择”一栏中选择欲使用的公式编号；
4. 在“插入引用”一栏中选择“引用”；
5. 点击“插入”(图 20)。

如果您向文档中已插入引用的公式之前添加其他公式，所有公式将会重新编号，交叉引用也会自动更新。

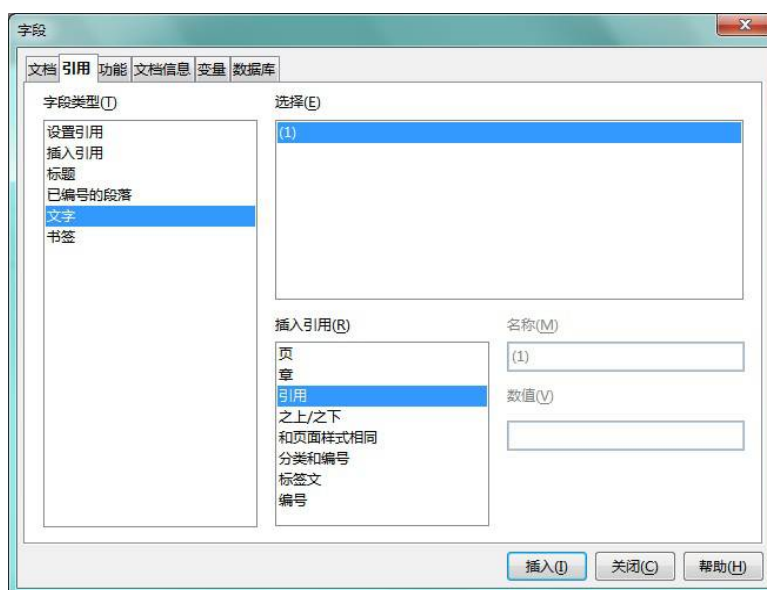


图 20 向公式编号插入交叉引用

提示：如果您想要插入不带括号的数字编号，在“插入引用”一栏中选择“编号”。

按照上述步骤插入的自动图文集(AutoText)包括一个 1 行 2 列的矩阵，左栏用于存储公式内容而右栏用于存储公式编号，这个编号是一个可自动更新的计数器。您可以按需编辑自动图文集的格式，例如，您希望使用方括号代替圆括号来表示公式编号，或是您希望公式和编号之间以制表符(Tab)相隔，而不是将二者作为表格处理，您可以查阅 Writer 导引第三章有关使用自动图文集的章节内容。

7.2 位置

默认设置下，公式被作为字符锁定在 Writer 文档中。但是，与任何其他类型的 OLE 对象相似，您可以通过改变它们的锁定方式来改变公式在文档中的嵌入方式。

默认条件下，作为字符方式锁定在文档中的公式对象将会基于其周围的文字基线垂直对齐。如果您需要手动改变公式的对齐方式，可以选择“工具→选项→LibreOffice Writer→格式化辅助”，取消“数字基线对齐”的复选框。此项设定对当前文档中的所有公式有效。新建文档将会使用此对话框的当前设置。

7.3 两侧空白

插入文档中的 Math 公式对象左右两侧都会有空白，用以同周围环绕的文字分隔开来。如果您不需要这种两侧空白，最好的方法是改变公式的框架样式。此项设置的更改将会对文档中已有的以及将要向文档中新增的公式立即生效，操作步骤如下：

1. 按下 F11，弹出“样式和格式”对话框；
2. 选择“框架样式”选项卡(将鼠标指针悬停在按钮上方可以看到工具提示)；
3. 右键点击“公式”；
4. 从弹出的上下文菜单中选择“修改”，将会出现“框样式：公式”对话框；
5. 选择“环绕”选项卡，更改左右间隔数值为 0.00；
6. 点击“确定”关闭对话框。

这一更改将会影响所有没有手动指定间隔大小的公式对象。您可以找到使用样式的更多信息，参见 Writer 导引第三章有关使用样式和模板的内容。

7.4 文字模式

一般情况下，占用空间较大的公式最好独立显示的一行中，与上下文的文字部分分开。当您在运行文字中使用公式元素时，它们一般会比上下文中文字占用更大的高度。

如果您一定要将公式排在运行文字中，您可以切换到公式编辑器，选择“格式→文字模式”。Math 将会尝试尽可能压缩公式高度以适应上下文的文字。在分式中，分子和分母的字号将会缩小，而积分与求和公式的上下限将会在在符号一侧显示。例如，以下公式独立占用一行：

$$\sum_{i=2}^5 i^2$$

在文字模式中： $\sum_{i=2}^5 i^2$

7.5 背景、边框和大小

作为对公式的格式化，公式被视为具有公式框架样式的“框样式对象”(frame style type)。背景色和边框可以直接通过使用样式，或选择“格式→图文框/对象”，或右键点击公式，从上下文菜单中选取“对象”来进行修改。在默认设置下，公式具有无色背景，无边框，大小不可更改。在 Writer 文档中，公式的大小取决于创建公式时的基本字体大小。

7.6 创建公式库

如果您直接使用 Math 的“文件→新建→公式”功能创建公式，您将得到单独的 odf 文件，每个文件包含一个独立的公式。您可以用这些 odf 文件创建常用公式的集合。嵌入公式库的公式也可以作为独立的公式保存。右键点击欲保存的公式，选择“副本另存为”。如需将这样的 Math 文档插入 Writer 文档中，选择“插入→对象→OLE 对象”，选择“从文件创建”并输入或浏览路径名和文件名，也可以利用操作系统自带的搜索功能。

提示：您不能通过鼠标拖放或“插入→文件”的方式向 Writer 中插入公式。

由于公式文档不具备图形格式属性，您不能将公式保存在图片库中。然而，您可以将公式保

存为自动图文集。在独立的一行中输入公式，点击选中公式，然后选择“编辑→自动图文集”。更多信息参见 Writer 导引第三章关于使用自动图文集的内容。

7.7 快速插入公式

如果您已知公式的标记语言代码，您可以通过以下方法快速插入公式：

- 在 Writer 中输入标记语言代码；
- 选中标记语言代码；
- 使用工具栏按钮、菜单项目或快捷键(参见 9.2~9.3 节)插入公式。

这种方法使您无需打开再关闭 Math 界面，可以为您节省时间。

8. Calc、Impress 和 Draw 文档中的公式

8.1 图形属性

公式对象在 Calc、Impress 和 Draw 中具有相似的属性，它们总是以无色背景、无边框的形式插入到上述类型文档中。在 Draw 和 Impress 中，公式对象被默认赋予图形属性，而在 Calc 中，公式不会被赋予任何属性，需要手动指定。在“样式和格式”(F11)窗口、上下文菜单或“格式→图片”选项中列出的属性是有限的，或者完全不可用。

8.1.1 线条、平面和阴影

您可以设置这些参数值，但它们对于公式完全无效。

8.1.2 文字属性

所有文字属性，比如字体和对齐方式，并不是针对公式中的文字，而是针对所有图形属性对象中存在的文字元素。对于一个选定的对象，可以通过 F2 键访问这些属性，更多信息参见 Draw 导引第 10 章。

8.1.3 位置

您可以使用鼠标拖放、方向键或“格式→对象→位置和大小”选项的方式将公式置于上述类型文档的任何位置。

8.1.4 大小

在上述类型文档中，公式并不是直接显示出来，而是替换为一个图形，这个图形的初始大小是被锁定的。如果您需要更改大小，选择“格式→对象→位置和大小”，在第一个页面的“保护”一栏中反选“大小”复选框，这将解锁上方“大小”一栏的选项。输入您所需的更改，当关闭对话框时，公式的大小会立即被再次锁定。

用以上方式更改公式大小并不影响公式本身，只是改变了与之对应的图形显示的大小，特别是公式中的基本字体大小不会随之改变。如果需要恢复由公式内容决定的基本大小，您可以在上下文菜单中选择“原始大小”。这个功能可以用于纠正由于插入由不同模块创建的公式

时所引起的大小错乱。

8.1.5 旋转、剪切和翻转

旋转、剪切和翻转对于公式是不可能的，对应对话框选项处于不可用状态。如果您需要进行这些操作，您需要将公式转换为 GDI (graphics device interface)图元文件(metafile)。一旦转换为 GDI 图元文件，原来的公式不再具有公式属性而成为一个图片。将公式复制到剪切板，选择“编辑→选择性粘贴”，或者在点击“标准”工具栏中“粘贴”图标旁边的三角形，在下拉菜单中选择“GDI 图元文件”选项。

8.2 将公式与文字组合

由于公式是 OLE 对象，您不能将其插入电子表格(spreadsheet，又称电子试算表)的单元格、图形或演示文稿的文字内容中。与 Writer 文档不同，您不能将公式与运行文本组合。不过，以下列出了一些其他解决方法：

8.2.1 公式内部的文字元素

您可以直接将文字写进公式中，例如：

This illustration of a parallelogram suggests the relationship
 $\overrightarrow{AB} = \overrightarrow{CD} \Leftrightarrow \overrightarrow{AC} = \overrightarrow{BD}$
But this does not constitute a proof.

其标记语言代码为：

```
"This illustration of a parallelogram suggests the relationship" newline  
alignl {widevec {AB} = widevec {CD} dlrarrow widevec {AC} =  
widevec {BD}} newline  
"But this does not constitute a proof."
```

您需要使用 newline 命令换行。您还可以使用“格式→字体”选项使得公式中的字体与环绕文本相匹配，参见 6.2 节。

8.2.2 将公式与图形对象组合

与文本文档中无法实现此功能不同，公式可以简单地与其他对象组合。在下例中，文本框与公式组合为整体，而矩形则被用于强调公式中的一部分：

The diagram illustrates how text and formulas can be combined in a document. It shows three lines of text and formula, each enclosed in a green square frame. The first line is "Now, the quadratic component in introduced". The second line is the formula $x^2 - 6x + 3^2 - 3^2 - 7$, where the terms 3^2 and -3^2 are highlighted with yellow rectangular boxes. The third line is "and then elements are gathered". Below this, the same formula $x^2 - 6x + 3^2 - 3^2 - 7$ is shown, but the terms $x^2 - 6x + 3^2$ and $-3^2 - 7$ are each enclosed in a red rectangular box. The final line shows the formula $(x - 3)^2 - 4^2$.

Now, the quadratic component in introduced

$$= x^2 - 6x + 3^2 - 3^2 - 7$$

and then elements are gathered

$$= (x^2 - 6x + 3^2) - 3^2 - 7$$
$$= (x - 3)^2 - 4^2$$

8.2.3 Writer 文档中的 OLE 对象中的公式

为了使用文本文档中的全部辅助功能，您可以在 Writer 中创建公式和文本，选择一个适当的页面大小，大致与您的最终目标相当，不要输入多于一页的内容(页面选择可能会造成其他问题)，保存此文档。

在 Impress 或 Draw、Calc 中，选择“插入→对象→OLE 对象”，然后选择“从文件创建”，公式将会以其本身的 OLE 对象形式被嵌入文本，其结果是公式不能在 Impress 中进行修改。因此您需要保持“链接到文件”复选框被选中。如果您需要对公式进行修改，打开文本文档并在其中修改公式和文本。回到 Impress，选择“编辑→链接”，按下“更新”按钮，即可显示存储于当前文本文档中的内容，不要尝试直接在 Impress 中修改公式或文本。

8.3 图表中的公式

由于图表本身也是一种 OLE 对象，因此您不能在其中调用公式编辑器。而 LibreOffice 暂时并不包括独立的模块用以创建图表，您也不能用与上文所述类似的方法在 Writer 中向图表中插入公式。不过您可以在图表编辑器以外创建公式，将其复制到剪切板再回到当前编辑的图表，从剪切板中粘贴公式。这样，公式将会自动转换为图元文件。如果您需要修改公式，您需要重新编辑并粘贴公式。

9. 自定义

您可以通过以下几种方式自定义 Math 使用：

9.1 使公式编辑器成为浮动窗口

公式编辑器可能会占用 Writer 窗口相当大的一部分高度，如需将其转换为浮动窗口，可以采取以下步骤：

1. 将鼠标指针悬停在公式编辑器框架上方(图 21)；
2. 按住 Ctrl 键，双击鼠标。

结果如图 22 所示，您可以用同样方式使浮动窗口再次锁定在 Writer 窗口下方。

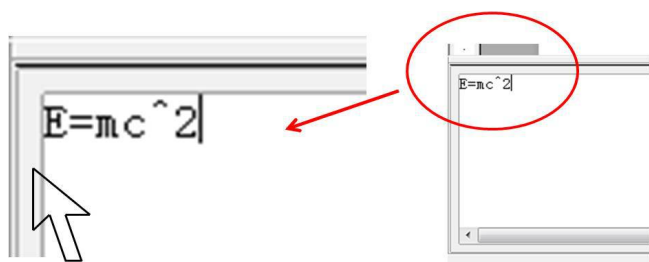


图 21 将鼠标悬停在公式编辑器边框

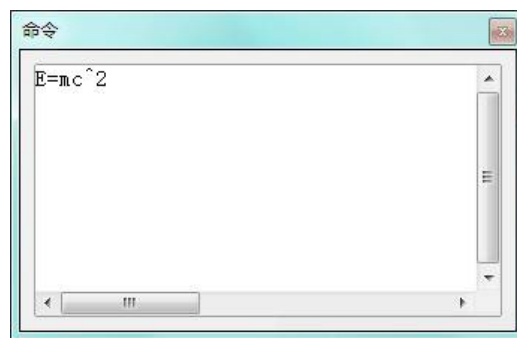


图 22 公式编辑器成为浮动窗口

9.2 添加工具栏按钮

插入公式的按钮默认位于 Writer 和 Impress 的“插入”工具栏，您也可以通过如下方式将其添加到其他工具栏：

- 选择“视图→工具栏→自定义”或右键点击任一工具栏最左端，选择“自定义工具栏”，在“工具栏”选项卡中选择需要修改的工具栏；
- 在“工具栏内容”列表中选择您想要插入该按钮的位置；
- 点击“添加”；
- 在“添加命令”对话框中，选择“插入”类别，然后选择“公式”命令(如图 23 所示)；
- 点击“附加”，您还可以继续添加其他按钮。当您完成后点击“关闭”；
- 在“自定义”对话框中，可以利用方向键改变按钮在列表中的位置，然后点击“确定”保存更改。

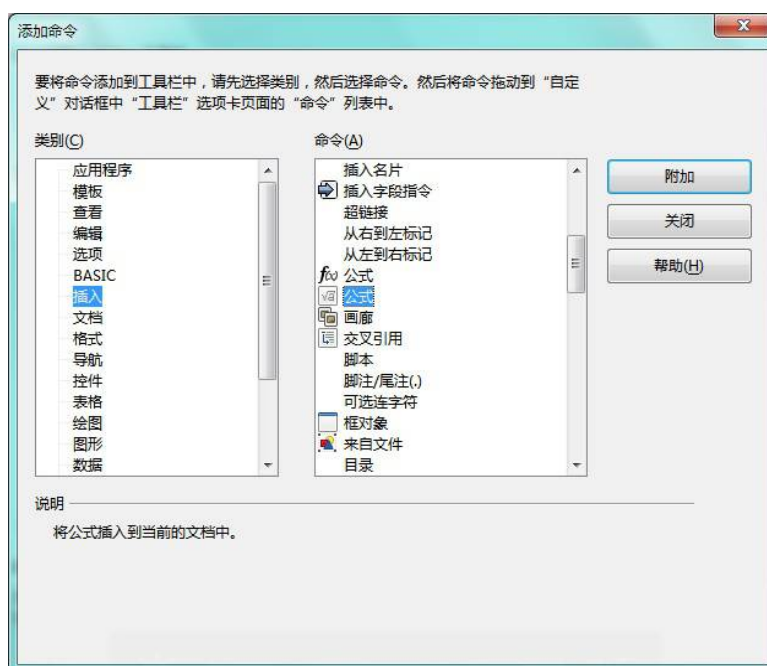


图 23 添加插入公式的按钮(注意，有两种“公式”按钮)

9.3 添加快捷键

您可以添加用于插入公式的快捷键：

- 选择“工具→自定义”，然后选择“键盘”选项卡；
- 选择欲创建的快捷键的级别(适用于所有 LibreOffice 模块或仅适用于当前模块)；
- 选择“范围”列表中的“插入”，然后选择“功能”列表中的“公式”(有两个“公式”选项，其中一个默认与 F2 相关联，这里应选取另一个)；
- 选择一个可用的快捷键，然后点击“修改”；
- 以同样方式添加同一级别的快捷键，然后点击“确定”保存更改。

如果您设定的快捷键打开了类似 Calc 的公式栏，那么您选择了错误的功能。

9.4 自定义图标列表

如果您需要使用不包含在 Math 默认设置中的符号，您可以手动将其添加至图标列表，以下例子示意如何向 Math 中添加常用于教学工作的符号。

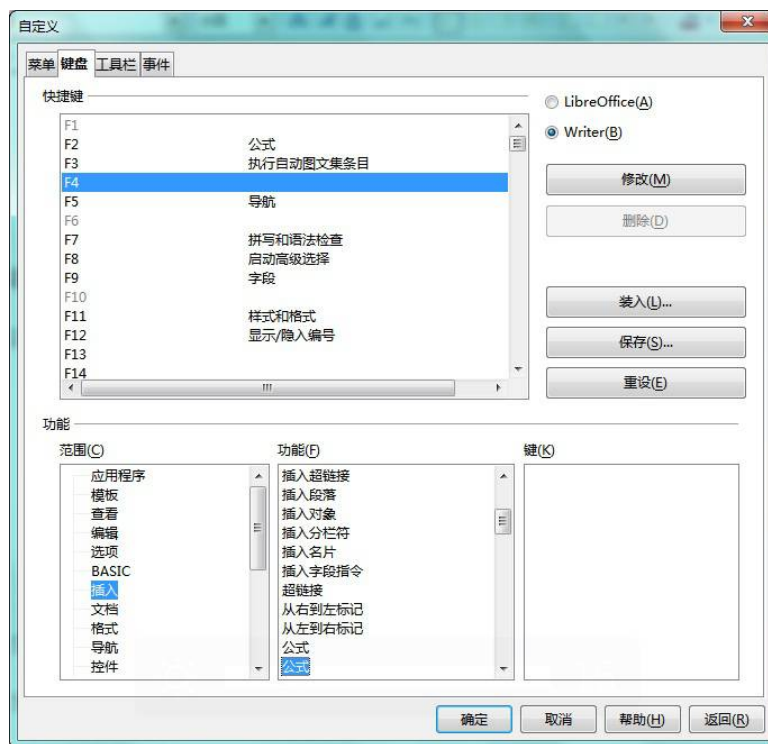


图 24 添加快捷键

1. 选择“工具→图标”或点击工具栏上的图标，打开“图标”对话框；
2. 点击“编辑”按钮，将会出现“编辑图标”对话框；
3. 所有图标被分置在“图标组”内。在窗口下方为您所需使用的图标选择一个可用的图标组名称，例如“特殊”。您也可以在对应的文本框内直接输入新的图标组名称；
4. 在“字体”列表框中选择一种包含您所需使用的符号的字体。当您选中一种字体后，它所包含的所有字符将会出现在上方字符列表中。您可以通过拖曳滚动条或选择适当的“子集”在字符列表中找到您所需要的符号。在本例中，选择 Times New Roman 字体以及“普通的标点符号”子集；
5. 选择您所需要的符号(本例中为 Ux2032)，它将放大显示在窗口右侧预览框内(图 26)。确认您所选择的图标组为“特殊”或您自定义的图标组；
6. 在“图标”文本框中输入该符号的名称，例如 prime (撇号)；
7. 如果该符号的名称未被使用，则“新增”按钮变为可用，点击“新增”；
8. 您还可以继续添加更多符号，本例中继续添加了 Ux2033 和 Ux2034；
9. 点击“确定”关闭对话框。

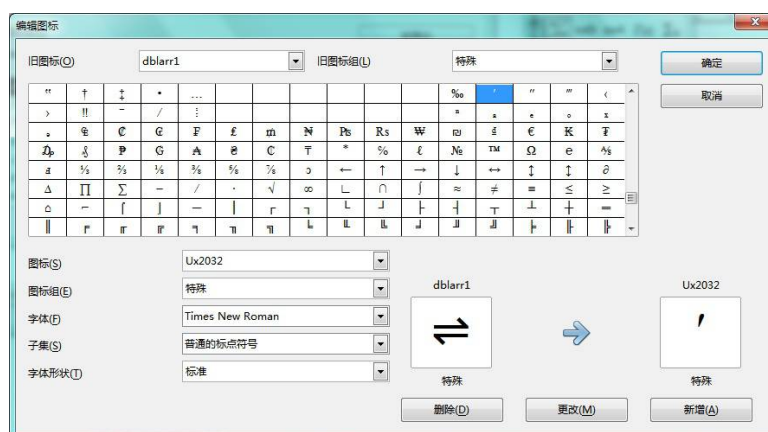


图 25 编辑图标对话框



图 26 编辑图标：选定欲添加的符号

现在，图标组中显示出了新增的符号(图 27)。您可以像使用 Math 自带符号那样使用它们，既可在该界面选取，也可直接在公式编辑器中使用标记语言`%prime`。

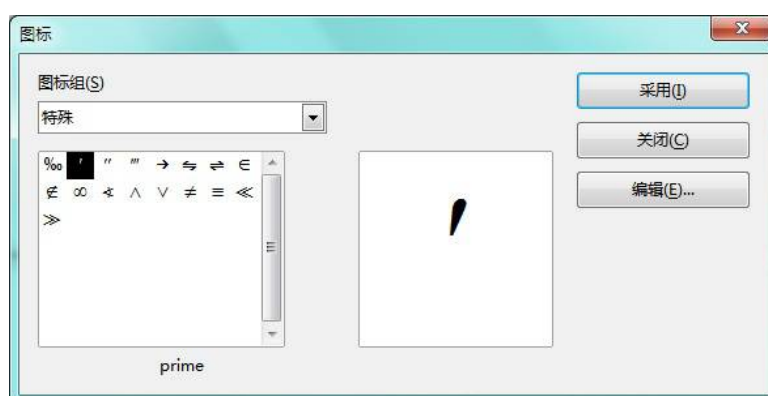


图 27 图标组中出现新增符号

一些免费字体包含大量专业数学符号，其中 STIX 是一种值得一提的字体，它是专门被设计用于书写数学或专业技术文档的字体。其他字体诸如 DejaVu、Lucida 也包含大量常用数学符号。

在 LibreOffice 默认配置中，只有实际使用到的用户自定义符号才会被嵌入文件中。有时您需要嵌入所有用户自定义符号，例如此篇文档需要与他人共同分享、编辑。选择“工具→选项→LibreOffice Math→设置”，取消“只嵌入用到的符号”复选框。此选项只有在当前使用 Math 编辑文档时有效。

9.5 自定义算符

任何需要识别为函数(使用正体显示)的名称需要加上标记语言命令 `func` 进行定义。任何需要识别为算符(加大字体显示，并且可带类似积分或求和公式的上下限)的名称需要使用 `oper` 定义。您可以使用单个字母、单词或自定义符号作为算符的名称，例如：

$\sum_{i \in I} A_i$ 可写作 `oper F from {i in I} A_i`。

您可以改变算符的行为，或将您的自定义符号转换为算符。例如，使用标记语言 `boper` 可以创建二元运算符，而 `uoper` 将创建一元运算符。

9.6 公式末尾的空白

符号```将会在公式中插入一小段空白，符号`~`将会插入一大段空白。在默认设置下，公式末尾的空白符号将被略去。然而，在运行文字中插入公式时，可能需要在公式末尾插入空白，此选项仅当您正在使用 Math 编辑文档时可用，而不是在向其他文档中插入公式时。

选择“文件→新建→公式”创建一个 Math 文档，选择“工具→选项→LibreOffice Math→设置”，取消“忽略行尾的~与`符号”复选框。

9.7 使用样式的默认布局

在 Writer 文档中，公式默认被按照公式框架样式来进行格式化。在“样式和格式”(F11)窗口中，点击上方第三个图标“框架样式”，然后右键点击“公式”并选择“修改”(图 28)，您可以通过此种方式直接修改文档中所有公式的样式，包括间隔和背景。不然您将需要手动修改每个公式的格式。

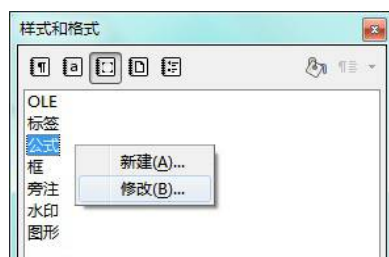


图 28 更改公式的框架样式

如果要将此种样式应用到所有新建 Writer 文档中，您需要将这种公式框架样式添加到模板之中。要实现此目的，您可以新建一个 Writer 文档，按照您所需的效果修改公式框架样式，选择“文件→模板→保存”来保存对模板的修改。为了将此模板定义为默认模板，选择“文件→模板→管理”，双击“我的模板”，然后右键点击您刚才创建的模板，选择“设为默认模板”。如果您需要恢复原来的默认模板，使用命令“重置默认模板”。此后新建的文档将基于当前的默认模板。

9.8 在化学中的应用

Math 被设计用于书写数学公式，但它也可用于书写化学方程式。在化学中，诸如 H_2O 这样的名称通常用大写正体字母书写。要使用 Math 书写化学式，您需要首先知道如何设置非斜体字体(参见 6.2 节)。

以下是一些化学式的范例：

类别	示例	标记语言
分子式	H_2SO_4	<code>nitalic {H_2 SO_4}</code>
同位素	$^{238}_{92}\text{U}$	<code>nitalic {U lsub 92 lsup 238}</code>
离子	SO_4^{2-}	<code>nitalic {SO_4^{2-}}</code>

`lsub` 和 `lsup` 分别是左侧下标和左侧上标的代码，在硫酸根离子的 2-右边的空括号是必需的，因为减号是二元运算符。

对于可逆化学反应，Math 中并不包括适当的双向箭头。如果您拥有包含了这些符号的字体，则可以将这些符号添加到图标组中(参见 9.4 节)。例如，DejaVu 字体包含了所需的双向箭头。如果您没有合适的符号字体可用，您还可以从现成的文档中找到所需符号并粘贴到您的文档中。例如对于上述可逆反应，您可以从以下链接获取所需符号(符号 x021C4)：
<http://dev.w3.org/html5/html-author/charref>

10. 导出和导入

10.1 导出为 MathML

除了可以将文档导出为 PDF (portable document format)，Math 还允许将公式导出为 MathML (mathematical markup language)格式。然而，当前的主流浏览器对这种标准的支持欠佳。如果在网页中显示遇到问题，那么问题发生在您使用的浏览器上。

您可以用以下公式(包含了 Math 提供的大部分功能组合)测试浏览器的兼容性：

标记语言	公式
<pre>%izeta(z)=sum from {n=1} to infinity {1 over {n^z}} newline left(matrix {a # b ## c # d} right) newline int_0^x f(t) nitalic {d}t newline x=3 y=1</pre>	$\zeta(z)=\sum_{n=1}^{\infty}\frac{1}{n^z}$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $\int_0^x f(t)dt$ $x=3y=1$

如果您正在编辑独立的 Math 公式，选择“文件→另存为”，打开“另存为”对话框，在“保存类型”下拉菜单中选择 MathML 格式。对于嵌入文档中的公式对象，在右键上下文菜单中选择“副本另存为”选项。

10.2 微软文件格式

选择“工具→选项→装入/保存”，在 Microsoft Office 一栏中的选项设置决定了针对公式如何进行微软 Office 文件格式的导入和导出。

10.2.1 载入

如果[L]选项被选中，当文档被加载时，LibreOffice 尝试将微软格式的公式转换为原生格式。在文档中的公式是由 MathType 3.1 以下版本或微软 Office 公式编辑器创建的情况下，这种转换是可能的。其中微软 Office 自带的“公式编辑器”可看做简化版的旧版本 MathType，它由微软授权并捆绑在 Office 办公套件中。

由新版本 MathType 或新版本微软 Office 公式编辑器(OMML 公式编辑器，用于 Office 2010，以及在一定程度上用于 Office 2007，OMML 是 Office Math markup language 的缩写)的公式不能被转换。

如果一份由微软 Office 2010 创建的 Word 文档包含 OMML 公式对象，并被保存为 doc 格式，则 Office 会将该公式转换为图形，而 LibreOffice 只能访问这个图形。

如果您试图载入一个包含 OMML 公式的 docx 文档，不论[L]是否被选中，均无法对此文档中的公式进行转换。

在微软 Office 中，使用 MathType 或微软公式编辑器创建的公式被视为 OLE 对象。如果[L]未被选中，LibreOffice 会维持此设定，双击此对象将会尝试打开 MathType，而且可以通过选

择“插入→对象→OLE 对象”来插入新的公式。如果您的操作系统安装了 MathType (仅限 Windows 或 Mac OS X)，并且希望使用 MathType 创建公式，建议使用此种设置。

10.2.2 保存

如果[S]被选中，LibreOffice 会将公式保存为可被微软公式编辑器或 MathType 读取和修改的格式。当[S]未被选中时，公式会在保存为 doc 文档时被视为 OLE 对象，但仍与 LibreOffice 关联。如果使用 Word 打开文档并双击公式对象，将会尝试打开 LibreOffice Math 进行编辑。

如果您将文档保存为 docx 格式，不论[S]是否选中，公式都不会被转换。

10.2.3 在微软 Word 2010 中打开 OpenDocument 文本文档

当您使用微软 Word 2010 时，可以打开由 LibreOffice Writer 创建的 odt 文档。Office 2010 会报告错误，但同时打开一个“修复的”文档副本。在这个副本中，所有原始公式被转换为 OMML 格式。如果您再次将其保存为 odt 格式，Office 2010 会将公式转换为 MathML 格式，这些公式可被 LibreOffice 打开和编辑。注意，编辑公式时务必将欲组合在一起的字符用括号括起来，即使某些括号在用于 LibreOffice 公式显示时是多余的。这样做将会确保 LibreOffice 能够将公式正确地转换为 MathML。例如，使用标记语言 $\sum_{i=1}^n i^2$ 代替 `sum from i=1 to n i^2`。

在这个过程中，注释元素(annotation element)将会丢失，其结果是 LibreOffice 无法将原始的 Math 标记语言代码正确地显示在公式编辑器中，而是尝试从 MathML 内部的标记法中生成新的标记语言代码。

10.3 XHTML

XHTML (extensible hypertext markup language，可扩展超文本标记语言)格式仅可用于导出，因此您将会在“文件→导出”中找到此选项，而不是在“文件→另存为”中。如果在您的软件设置中没有找到此选项，您可以重新安装必要的过滤器(运行 LibreOffice 安装程序，选择“更改”，然后选择“可选组件”中的 XSLT 样本过滤器，XSLT 是 extensible stylesheet language transformations 的缩写，意为“可扩展样式表转换语言”)。在此种情况下，将采用图形替换公式嵌入在文件中，而作为 MathML 注释元素的原始 Math 标记语言代码将被舍弃。

文档能否正确显示取决于浏览器。Firefox 可以正确地显示文档和公式，而 Opera 和 IE 在显示公式(嵌入的图形)时都存在错误。

旧版本 HTML 文档过滤器(用于 OpenOffice.org Writer 程序)完全无法保存公式，而只能使用一幅显示质量很差的 gif 图片作为替代。

10.4 Flat XML

这种格式在 LibreOffice 中既可读取又可保存。如果在“文件→打开”或“文件→另存为”中找不到相应选项，可以重新安装必需的过滤器。在这种文件格式中，文档的全部内容包括模板将会保存在一个 XML 文件中。换言之，它不再像保存为开放文档格式那样是一个压缩的文件夹。所有公式将被视为 MathML 元素嵌入，这一点与导出为 MathML 相似。然而，任何公式个体都没有被保存，真正被保存的只是包含所有公式的整个文档。这种格式适合于直接进入外部控制系统版本。

11. 扩展包

如果您经常使用 Math，您可能希望安装以下扩展包之一以进一步简化公式编辑步骤。这些程序并不互相竞争，而是在可能的情况下相互辅助。然而，Dmaths 和 CmathOOo 无法同时安装，它们可以互相补充。因此，您可以在不同的系统登录用户下分别试用再决定取舍。

11.1 Dmaths：不仅可以用于快速输入公式的扩展包

Dmaths 是一个用于 LibreOffice Writer 的宏软件包，这个数学软件包可以作为扩展程序安装，它可以从以下网址下载：<http://extensions.libreoffice.org>。

安装 Dmaths 后，关闭并重新打开 LibreOffice，在 Writer 模块中将会新增以下功能：一个新的菜单项、一个具有 36 个按钮的可自定义工具栏、一个具有 26 个按钮用于 Gdmath 的工具栏、一个具有 10 个按钮用于 AHmath3D 的工具栏、两个用于控制长工具栏开关并且访问一些新功能的短工具栏。

Dmaths 提供一些用于创建数学文档的工具：

- 单击从文本输入创建 Math 公式(橙色)；
- 使用输入对话框创建公式(紫色)；
- 标记文本(蓝色)；
- 修改已有公式的属性(绿色)；
- 绘制图形、网格、统计表格和几何图形(青色)；
- 绘制和更改几何图形对象(灰色 Gdmath 工具栏，在 Draw 中也可用)；
- 绘制 3D 对象(AHmath3D 工具栏，带有连线模型)。

Dmaths 菜单提供访问以下内容：更多的添加件、配置，以及一个具体的 Dmaths 帮助导引。

Dmaths 最早由法国数学家 Didier Dorange 开发，但现在已有德文、英文、西班牙文版本。官方网站为：www.dmaths.org。

11.2 利用 CmathOOo 或 CmathOOoCAS 进行符号运算

CmathOOo 允许您使用与学生计算器相似的语法结构直接在文字处理软件中输入数学公式。CmathOOoCAS 允许您直接在文字处理软件中进行符号计算(借助 Xcas)，其结果的格式化与编排布局由 CmathOOo 完成。这些扩展程序可以在以下网址下载：

<http://extensions.services.openoffice.org/fr/project/CmathOOo>

<http://extensions.services.openoffice.org/fr/project/CmathOOoCAS>

12. 技术细节

12.1 用于办公应用的 OASIS 开放文档格式

开放文档格式(open document format, ODF)是由 OASIS (Organization for the Advancement of Structured Information Standards, 结构化信息标准推进组织)开发的开放文档标准。ODF 规范

并不为 Math 公式创建一套全新的标准，而是接受了由 W3C (World Wide Web Consortium, 万维网联盟) 开发的标准数学标记语言(MathML)，以期在网页上显示公式。对于 W3C 提供的两种标记语言类型：表示性标记语言(presentation markup)和内容性标记语言(content markup)，ODF 采用前者，它描述公式的表现形式而不是公式的具体数学含义。

MathML 不同于用于公式编辑器的标记语言。例如，对于公式 $\int_{f(t)}^{n(t)} dt$ ，其中的 from...to... 部分不会被翻译，而是直接转换为类似于 $\int_{i=1}^3$ 的 MathML 代码。公式编辑器中的原始公式内容使用 StarMath 5.0 的编码格式。为防止丢失信息，它被保存为一个注释元素。其他应用程序不需要评估或保存这种元素，也不需要自行创建这种元素。

在 LibreOffice 中，一个公式总是与一幅替代图形相关联。在 odf、odt 等文件封装格式中，这些替代图形存储于单独的文件夹内，并且与它们所在文档保持内在关联。然而，如果您使用“开放文档文本” Flat XML 文件格式或导出为 XHTML，这些图形必须被嵌入。出于这种目的，它们将以 BASE64 格式进行编码。

12.2 用 Basic 语言处理公式

这部分内容不是关于 Basic 语言编程的介绍，而是简单描述公式对象的一些特殊属性。您可以从以下地址得到关于类型和服务的更为具体的信息：

<http://api.libreoffice.org/common/ref/com/sun/star/module-ix.html>

如果您在索引中查找此处提供的标记，您将快速找到相关描述。

12.2.1 Writer 文档中的公式

在 Writer 文档中，包括公式在内的所有 OLE 对象都属于以下的类别集合：

SwXTextEmbeddedObjects

```
oCurrentController = ThisComponent.getCurrentController()  
oTextDocument = oCurrentController.Model  
oEmbeddedObjects = oTextDocument.EmbeddedObjects
```

这一集合从 0 开始连续编号，您可以用以下方式查找任一给定对象：

```
nEndIndex = oEmbeddedObjects.Count - 1  
for nIndex = 0 to nEndIndex  
    oMathObject = oEmbeddedObjects.GetByIndex(nIndex)
```

要检测这个对象是否为公式，可以测试它是否支持公式对象所支持的服务：

```
oModel = oModelObject.Model  
if  
oModel.supportsService("com.sun.star.formula.FormulaProperties")  
then
```

或者，可以检测该对象的 CLSID 是否属于 Math 公式(任何 OLE 对象都有 CLSID)：

```
if oMathObject.CLSID = "078B7ABA-54FC-457F-8551-6147e776a997" then
```

使用这种模型，您可以直接修改公式属性，例如基本字体大小：

```
oModel.BaseFontHeight = 12
```

为使更改可见，需要重新绘制公式。属性 `ExtendedControlOverEmbeddedObject` 提供了一些特别应用于 OLE 对象的方法：

```
oXCOEO = oMathObject.ExtendedControlOverEmbeddedObject  
oXCOEO.update()
```

作为范例，以下宏程序将更改文本文档中已有的任何公式的任何字体格式：

```
Sub ChangeFormatFormula  
    oCurrentController = ThisComponent.getCurrentController()  
    oTextDocument = oCurrentController.Model  
    oEmbeddedObjects = oTextDocument.EmbeddedObjects  
    nEndIndex = oEmbeddedObjects.Count - 1  
    for nIndex = 0 to nEndIndex  
        oMathObject = oEmbeddedObjects.getByIndex(nIndex)  
        oModel = oMathObject.Model  
        if (not isNull(oModel)) then  
            if (not isEmpty(oModel)) then  
                if  
oModel.supportsService("com.sun.star.formula.FormulaProperties")  
then  
                ' or if oMathObject.CLSID = "078B7ABA-54FC-8551-  
6147e776a997" then  
                    oModel.BaseFontHeight = 11  
                    policeCommune = "Liberation Serif"  
                ' Variables  
                    oModel.FontNameVariables = policeCommune  
                    oModel.FontVariablesIsItalic = True  
                    oModel.FontVariablesIsBold = False  
                ' Functions  
                    oModel.FontNameFunctions = policeCommune  
                    oModel.FontFunctionsIsItalic = False  
                    oModel.FontFunctionsIsBold = False  
                ' Numbers  
                    oModelNameNumbers = policeCommune  
                    oModelNumbersIsItalic = False  
                    oModelNumbersIsBold = False  
                ' Text  
                    oModelNameText = policeCommune  
                    oModelTextIsItalic = False  
                    oModelTextIsBold = False  
                ' Update  
                    oXCOEO = oMathObject.ExtendedControlOverEmbeddedObject  
                    oXCOEO.update()  
                end if ' if formula  
            end if ' if not empty  
        end if ' if not null  
    next nIndex  
    ThisComponent.reformat()  
End Sub
```

以上程序代码中，蓝色部分为可以自行设定的参数(字体名称和字号)，绿色部分为注释。以上程序设定了分别用于显示变量、函数、数字和文字的字体属性。

您可以从以下网址查阅所有可供修改的属性：
<http://api.libreoffice.org/common/ref/com/sun/star/formula/FormulaProperties.html>

12.2.2 Draw、Impress、Calc 文档中的公式

在以上类型的文档中，OLE 对象相似地作为图形处理。当您访问了一个特定对象之后，首先测试它是否为 OLE 对象，然后测试它是否为公式。您的宏程序核心代码如下：

```
if oShape.supportsService("com.sun.star.drawing.OLE2Shape") then
  if oShape.CLSID = "078B7ABA-54FC-457F-8551-6147e776a997" then
    oModelFormula = oShape.Model
    oModelFormula.BaseFontHeight = 12
```

在这种情况下，不需要对公式进行具体更新。

附录

A.1 一元/二元运算符

算符	命令
正号	+
负号	-
正负号	+ - OR plusminus
负正号	- + OR minusplus
加法	a + b
减法	a - b
点乘(内积)	a cdot b
叉乘(外积)	a times b
乘法(星号)	a * b
除法(分式)	a over b
除法(符号)	a div b
除法(斜杠)	a / b
串接	a circ b
除法(宽斜杠)	a wideslash b
宽反斜杠	a widebslash b
逻辑非	neg a

逻辑与	<code>a and b</code> OR <code>a & b</code>
逻辑或	<code>a or b</code> OR <code>a b</code>
反斜杠	<code>a bslash b</code>
直和	<code>a oplus b</code> , <code>a ominus b</code>
张量积	<code>a otimes b</code> , <code>a odot b</code> , <code>a odivide b</code>
自定义一元运算符	<code>uoper monOp b</code>
自定义二元运算符	<code>a boper monOp b</code>

A.2 关系运算符

运算符	命令
等于	<code>a = b</code>
不等于	<code>a <> b</code> OR <code>a neq b</code>
小于	<code>a < b</code> OR <code>a lt b</code>
小于或等于	<code>a <=b</code> OR <code>a leslant b</code>
远小于	<code>a ll b</code> OR <code>a << b</code>
大于	<code>a > b</code> OR <code>a gt b</code>
大于或等于	<code>a >=b</code> OR <code>a geslant b</code>
远大于	<code>a gg b</code> or <code>a >> b</code>
约等于	<code>a approx b</code>
相似	<code>a sim b</code>
相似或等于	<code>a simeq b</code>
全等	<code>a equiv b</code>
正比	<code>a prop b</code>
平行	<code>a parallel b</code>
正交	<code>a ortho b</code>
整除	<code>a divides b</code>
不整除	<code>a ndivides b</code>
趋于	<code>a toward b</code>
左箭头	<code>a dlarrow b</code>
双向箭头	<code>a dlrarrow b</code>
右箭头	<code>a drarrow b</code>
优先	<code>a prec b</code>
继承	<code>a succ b</code>
优先或等于	<code>a preccurlyeq b</code>

继承或等于	$a \succcurlyeq b$
优先或相似	$a \succsim b$
继承或相似	$a \succsim b$
不优先	$a \not\succ b$
不继承	$a \not\succ b$
定义	$a \text{ def } b$
像	$a \text{ transl } b$
原像	$a \text{ transr } b$

A.3 集合运算符

运算符	命令
属于	$a \in B$
不属于	$a \notin B$
包含	$A \text{ owns } b \text{ OR } A \ni b$
交集	$A \text{ intersection } B$
并集	$A \text{ union } B$
差集	$A \text{ setminus } B$
商集	$A \text{ slash } B$
子集	$A \text{ subset } B$
子集或相等	$A \text{ subseteq } B$
超集	$A \text{ supset } B$
超集或相等	$A \text{ supseteq } B$
非子集	$A \text{ nsubset } B$
非子集或相等	$A \text{ nsubseteq } B$
非超集	$A \text{ nsupset } B$
非超集或相等	$A \text{ nsupseteq } B$
空集	emptyset
基数	\aleph
自然数集	setN
整数集	setZ
有理数集	setQ
实数集	setR
复数集	setC

A.4 函数

运算符	命令
绝对值	abs {a}
阶乘	fact {a}
根式	sqrt {a}
n 次根式	nroot {a}
乘幂	a^{b}
指数	func e^{a}
自然对数	ln {a}
指数函数	exp {a}
对数	log {a}
正弦	sin {a}
余弦	cos {a}
正切	tan {a}
余切	cot {a}
双曲正弦	sinh {a}
双曲余弦	cosh {a}
双曲正切	tanh {a}
双曲余切	coth {a}
反正弦	arcsin {a}
反余弦	arccos {a}
反正切	arctan {a}
反余切	arccot {a}
反双曲正弦	arsinh {a}
反双曲余弦	arcosh {a}
反双曲正切	artanh {a}
反双曲余切	arcoth {a}

A.5 积分与求和/积运算符(可带上下限)

运算符	命令
极限	lim {a}
下限	liminf {a}
上限	limsup {a}
求和	sum {a}
求积	prod {a}

余积	<code>coprod {a}</code>
积分	<code>int {a}</code>
二重积分	<code>iint {a}</code>
三重积分	<code>iiint {a}</code>
曲线积分	<code>lint {a}</code>
二重曲线积分	<code>llint {a}</code>
三重曲线积分	<code>lllint {a}</code>
求和符号下限	<code>sum from {x} b</code>
求积符号上限	<code>prod to {x} r</code>
积分上下限	<code>int from {r_0} to {r_t} a</code>
自定义算符	<code>oper Op from 0 to 1 a</code>

A.6 属性

运算符	命令
acute 号	<code>acute a</code>
grave 号	<code>grave a</code>
反音调符号	<code>check a</code>
breve 号	<code>breve a</code>
圆圈	<code>circle a</code>
点	<code>dot a</code>
双点	<code>ddot a</code>
三点	<code>dddots a</code>
上横线	<code>bar a</code>
向量箭头	<code>vec a</code>
波浪线	<code>tilde a</code>
音调符号	<code>hat a</code>
宽向量箭头	<code>widevec abc</code>
宽波浪线	<code>widetilde abc</code>
宽音调符号	<code>widehat abc</code>
上划线	<code>overline abc</code>
下划线	<code>underline abc</code>
删除线	<code>overstrike abc</code>
虚字符	<code>phantom a</code>
粗体	<code>bold a</code>

非粗体	<code>nbold a</code>
斜体	<code>ital "a" OR italic "a"</code>
非斜体	<code>nitalic a</code>
字号(绝对值)	<code>size 16 qv</code>
字号(相对值, 加减)	<code>size +12 qv</code>
字号(相对值, 比例)	<code>size *1.5 qv</code>
无衬线字体	<code>font sans qv</code>
衬线字体	<code>font serif qv</code>
等宽字体	<code>font fixed qv</code>
青色	<code>color cyan qv</code>
黄色	<code>color yellow qv</code>
白色	<code>color white qv</code>
绿色	<code>color green qv</code>
蓝色	<code>color blue qv</code>
红色	<code>color red qv</code>
黑色	<code>color black qv</code>
品红色	<code>color magenta qv</code>
为单个字段设置颜色	<code>color green X qv</code>
为多个字段设置颜色	<code>color green {X qv}</code>

A.7 括号

运算符	命令
分组括号(用于结构控制)	<code>{a}</code>
圆括号	<code>(a)</code>
方括号	<code>[b]</code>
双方括号	<code>ldbracket c rdbracket</code>
花括号	<code>lbrace w rbrace</code>
尖括号	<code>langle d rangle</code>
算符括号	<code>langle a mline b rangle</code>
下半方括号	<code>lceil a rceil</code>
上半方括号	<code>lfloor a rfloor</code>
单线	<code>lline a rline</code>
双线	<code>ldline a rdline</code>
圆括号(可变大小)	<code>left (stack {a # b # z} right)</code>

方括号(可变大小)	<code>left [a over b right]</code>
双方括号(可变大小)	<code>left ldbracket a over b rdbracket</code>
花括号(可变大小)	<code>left lbrace a over b right rbrace</code>
尖括号(可变大小)	<code>left langle a over b right rangle</code>
算符括号(可变大小)	<code>left langle a over b mline c right rangle</code>
上半方括号(可变大小)	<code>left lceil a over b right rceil</code>
下半方括号(可变大小)	<code>left lfloor a over b right rfloor</code>
单线(可变大小)	<code>left lline a over b right rline</code>
双线(可变大小)	<code>left ldline a over b right rdline</code>
不成对括号	<code>left langle a over b right rfloor</code>
孤立括号	<code>left lbrace stack {a=2 # b=3} right none</code>
上括号(可变大小)	<code>{the brace is above} overbrace a</code>
下括号(可变大小)	<code>{the brace is below} underbrace {f}</code>

A.8 格式

运算符	命令
右上标	<code>a^{b}</code>
右下标	<code>a_{b}</code>
左上标	<code>a \sup {b}</code>
左下标	<code>a \sub {b}</code>
中上标	<code>a \csup {b}</code>
中下标	<code>a \csub {b}</code>
换行	<code>newline</code>
短间隔	<code>`</code>
长间隔	<code>~</code>
无间隔	<code>nospace {x+y}</code>
普通间隔	<code>x+y</code>
上下堆叠(2个元素)	<code>binom {a} {b}</code>
上下堆叠(多个元素)	<code>stack {a # b # z}</code>
矩阵	<code>matrix {a # b ## c # d}</code>
左对齐	<code>alignl</code>
居中	<code>alignc</code>
右对齐	<code>alignr</code>
按等号对齐(使用矩阵)	<code>matrix {a # "=" # b ## {} # "=" # c+1}</code>

按等号对齐(使用虚字符)	<code>stack {a=b # phantom {a}=c+1}</code>
--------------	--

A.9 其他

运算符	命令
无穷	<code>infinity</code>
偏微分	<code>partial</code>
劈形	<code>nabla</code>
存在	<code>exists</code>
不存在	<code>notexists</code>
对所有	<code>forall</code>
\hbar	<code>hbar</code>
λ	<code>lambdabar</code>
实部	<code>re</code>
虚部	<code>im</code>
魏尔斯特拉斯 P	<code>wp</code>
反写的 ε	<code>backepsilon</code>
左箭头	<code>leftarrow</code>
右箭头	<code>rightarrow</code>
上箭头	<code>uparrow</code>
下箭头	<code>downarrow</code>
底部省略号	<code>dotslow</code>
中部省略号	<code>dotsaxis</code>
垂直省略号	<code>dotsvert</code>
向上对角省略号	<code>dotsup</code> OR <code>dotsdiag</code>
向下对角省略号	<code>dotsdown</code>

A.10 希腊字母

参见 3.4 节。

A.11 特殊符号(默认)

参见 9.4 节。

A.12 保留字

(参见附录 A.1~A.11)