

## Хитрости в OpenOffice.org Math

Я учусь в институте и уже не первый год как основной офисный пакет использую OpenOffice.org. Поскольку у меня техническая специальность, мне важно иметь под рукой инструменты для создания технических документов. Одним из них и является OpenOffice.org Math — редактор формул OpenOffice.org. Потратив пару дней на чтение кое-каких инструкций, я понял, что время, потраченное на освоение OOo Math, с лихвой окупится при наборе формул.

Но на одном удобстве метода далеко не уедешь. Пришлось научиться вводить не только простые, но и сложные формулы, наборы формул, выровненные особым образом, «многоэтажные» формулы и т.д. Предлагаю вашему вниманию несколько способов создания сложных формул.

Поскольку здесь речь пойдет именно о сложной разметке, предполагается, что читатель уже освоил основы работы с OpenOffice.org Math (например, прочитал 16-ю главу Руководства по Writer).

### Системы уравнений и неравенств

Запись систем рассмотрена в 16-й главе Руководства по Writer.

Пример 1

Формула	Код
$\left\{ \begin{array}{l} \text{уравнение1} \\ \text{уравнение2} \\ \text{уравнение3} \end{array} \right.$	<pre>left lbrace alignl { stack{ уравнение1# уравнение2# уравнение3} } right none</pre>

Разберем пример 1, где:

left lbrace — открывающаяся масштабируемая фигурная скобка;

right none — указывает, что закрывающей скобки не требуется;

alignl — устанавливает выравнивание строк по левому краю;

# -- переход на новую строку.

Следует учесть, что команды left и right нельзя использовать по отдельности. Кроме того, использование # как символа перехода на новую строку вне блока stack может привести к ошибке.

Скобки можно комбинировать, в частности, left [ right rbrace. Ниже приведено преобразование записи из совокупности неравенств в промежуток на числовой оси.

Пример 2

$\left[ \begin{array}{l} x < 0 \\ x > 5 \end{array} \right] \Rightarrow x \in (-\infty; 0) \cup (5; +\infty)$	<pre>left [ alignl stack{x&lt;0#x&gt;5} right rbrace drarrow x in (-infinity;0) union (5; +infinity)</pre>
---	--

## Системы уравнений и неравенств с условиями

Иногда к уравнениям требуется дописать условие. Например,  $y = 0$  при  $x > 3$ . Часто вместо слова «при» ставят запятую. Запишем пример системы уравнений с условиями.

Пример 3

$\left\{ \begin{array}{l} \text{уравнение1, условие1} \\ \text{уравнение2, условие2} \end{array} \right.$	<pre>left lbrace alignl stack{уравнение1, условие1# уравнение2, условие2} right none</pre>
---	--

Однако подобная запись не всегда выглядит эстетично.

Пример 4

$\left\{ \begin{array}{l} \text{уравнение1, условие1} \\ \text{длинное уравнение2, условие2} \end{array} \right.$	<pre>left lbrace alignl stack{уравнение1, условие1# длинное уравнение2, условие2} right none</pre>
---	--

Требуется реализовать такую запись, при которой условия находились бы друг под другом и были выровнены по левому краю. Для этого следует использовать команду `matrix`.

Пример 5

$\left\{ \begin{array}{ll} \text{уравнение1,} & \text{условие1} \\ \text{длинное уравнение2,} & \text{условие2} \end{array} \right.$	<pre>left lbrace alignl matrix{уравнение1,#условие1## длинное уравнение2,#условие2} right none</pre>
--	--

В коде из примера 5:

`left lbrace` -- открывающаяся масштабируемая фигурная скобка;

`right none` -- указывает, что закрывающей скобки не требуется;

`alignl` -- устанавливает выравнивание строк по левому краю.

`#` -- следующий элемент строки (т.е. переход на следующий столбец в текущей строке);

`##` -- переход на новую строку.

При такой записи можно опустить запятые после уравнений.

Теперь напишем кусочно-непрерывную функцию.

Пример 6

$y = \begin{cases} -x & x < -1 \\ 1 & -1 \leq x \leq 1 \\ x & x > 1 \end{cases}$	<pre>y=left lbrace alignl matrix { -x#x&lt;-1## 1#-1 leslant x leslant 1## x#x&gt;1 } right none</pre>
--	--

### Выравнивание в дробях

Если вы будете записывать системы уравнений, содержащие дроби, то наверняка столкнетесь с проблемой выравнивания числителя и знаменателя. Дело в том, что если указать alignl перед системой уравнений или перед конкретной строкой, то все элементы этой строки выровняются по левому краю, в том числе числитель и знаменатель дроби. То же будет происходить и в многострочных формулах, например при использовании newline и/или stack.

Пример 7

$\frac{1}{100}$	alignl 1 over {100}
-----------------	---------------------

Выход из ситуации — установка выравнивания по центру для числителя и знаменателя дроби.

Пример 8

$\frac{1}{100}$	alignl {alignc 1} over {alignc 100}
-----------------	-------------------------------------

Math определяет границы выравнивания по наиболее длинному элементу. Например, короткий числитель выравнивается по левому краю относительно левого края знаменателя. Аналогично короткий числитель выровняется по центру относительно знаменателя. В большинстве случаев можно заранее определить длины числителя и знаменателя, и тогда достаточно указать выравнивание либо числителя, либо знаменателя. Это сокращает записи.

Пример 9

$\frac{1}{100}$	alignl {alignc 1} over 100
$\frac{237}{8}$	alignl 237 over {alignc 8}

Рассмотрим многострочную формулу.

Пример 10

$\phi_2(2+j)=200+100j$ $\phi_2=\frac{200+100j}{2+j}=\frac{100(2+j)}{2+j}=100$	<pre>alignl dot %varphi_2(2+j)= 200+100 j newline alignl dot %varphi_2= {200+100 j}over{alignc 2+j}= {100(2+j)} over {alignc 2+j}=100</pre>
---	---

Подобным образом выравниваются числитель и знаменатель по центру в системах уравнений и неравенств.

## Расширенные матрицы

Теперь о записи расширенных матриц, не вдаваясь в математическую сторону вопроса. По сути требуется провести вертикальную линию между столбцами матрицы с помощью команды `mline` (пример 11).

Пример 11

$\left( \begin{array}{cc c} 1 & 2 & 5 \\ 3 & 4 & 6 \end{array} \right)$	<pre>left ( matrix{1#2##3#4} mline matrix{5##6} right )</pre>
---	---

Однако у `mline` есть серьезный недостаток, который мне лично пока не удалось компенсировать даже другими методами проведения вертикальной линии. Речь идет о некорректном выравнивании элементов матрицы друг относительно друга.

Пример 12

$\left( \begin{array}{cc c} 1 & 2 & \frac{5}{7} \\ 3 & 4 & 6 \end{array} \right)$	<pre>left ( matrix{1#2##3#4} mline matrix{5 over 7##6} right )</pre>
---	--

На самом деле цифра 6 должна быть на уровне цифр 3 и 4. Разработчики оповещены об этой недоработке.

## Хитрости

Обсудим приемы записи некоторых выражений в OOo Math. Примеры составлены на основании тех вопросов, на которые я отвечал себе и другим.

**Вопрос:** Как сделать двойной верхний/нижний индекс? Как создать оба индекса одновременно?

**Ответ:** Можно, в частности, так.

Пример 13

$A^B$	<code>A^{B^C}</code>
$X_{C_1}$	<code>X_{C_1}</code>
$x_1^2$	<code>x_1^2</code> или <code>x^2_1</code>

**Вопрос:** Как сделать «многоэтажные» дроби?

**Ответ:** С помощью операторных фигурных скобок.

Пример 14

$\frac{1 + \frac{2}{3}}{4 + \frac{5}{6}}$	<code>{1+2 over 3} over {4+5 over 6}</code>
---	---

**Вопрос:** Как напечатать знак равенства (неравенства, принадлежности и т. п.), не используя спецсимволы? (Math ругается на отсутствие операндов.)

**Ответ:** С помощью операторных фигурных скобок указать Math пустой операнд, записав на его месте {}.

Пример 15

=	<code>{}= {}</code>
≤	<code>{ }leslant{ }</code>
∈	<code>{ }in{ }</code>

С использованием подобной записи легко организовать перенос выражения на новую строку.

Пример 16

$y=5^2+3=$ $=28$	<code>y=5^2+3={ } newline</code> <code>{ }=28</code>
---------------------	---

В большинстве случаев необходимо выравнивание по левому краю. Поэтому изменим запись.

Пример 17

$y=5^2+3=$ $=28$	<code>alignl y=5^2+3={ } newline</code> <code>alignl { }=28</code>
------------------	---

Поскольку alignl выравнивает по левому краю относительно самой длинной строки в формуле, то alignl перед нею можно опустить. В рассматриваемом случае таковой строкой будет верхняя.

Пример 18

$y=5^2+3=$ $=28$	<code>y=5^2+3={ } newline</code> <code>alignl { }=28</code>
------------------	--

Визуально формулы выглядят одинаково.

**Вопрос:** Когда я ввожу Re, чтобы указать действительную часть комплексного числа, у меня подставляется спецсимвол, а мне этого не надо. Как записать выражение для действительной части числа и обозначение числа Рейнольдса?

**Ответ:** Поясню, о чем идет речь. У нас принято обозначать действительную часть комплексного числа  $z$  как  $Re(z)$ , т.е. Re является функцией и, следовательно, используется обычное начертание. Math подставляет символ  $\Re$  вместо Re. Один из способов решения основан на следующем. По умолчанию Math настроен так, что обычный текст оформляется обычным начертанием, а если это не так, то соответствующие параметры можно настроить из меню, «Формат» > «Шрифты». Поэтому просто укажем Math, что Re -- это текст (пример 19).

Пример 19

$Re(z)=x$	<code>"Re"(z)=x</code>
-----------	------------------------

Число Рейнольдса -- в общем случае переменная, а в конкретном -- константа. Поэтому оно должно быть оформлено курсивом: Re. Есть два способа решения этой проблемы: первый из них -- указать курсивное начертание.

Пример 20

<i>Re</i>	<code>ital "Re"</code>
-----------	------------------------

Второй сработает тогда, когда вы не обозначили какой-либо спецсимвол через %Re. Если нет соответствующего символа, Math записывает курсивом последовательность символов, стоящую после знака %.

Пример 21

<i>Re</i>	<code>%Re</code>
-----------	------------------



Полученная запись выглядит эстетичной и удобной с точки зрения дальнейшего решения проблемы.

Пример 26

$\begin{array}{cc} a & b \\ c & d \\ \hline 123 \end{array}$	<pre>matrix{a#b##c#d} newline overstrike matrix{phantom 1#phantom 2} newline 123</pre>
--	--

Рассмотрим запись системы уравнений без почленного выравнивания слагаемых, т.е. одно уравнение написано просто под другим.

Пример 27

$\begin{array}{l} + \left\{ \begin{array}{l} x+101y=15 \\ -x+2y=3 \end{array} \right. \\ \hline 103y=18 \end{array}$	<pre>"+" underline{ left lbrace alignl stack{ x+101 y=15# -x+2 y=3} right none} newline alignl 103 y=18</pre>
$\begin{array}{l} + \left\{ \begin{array}{l} x+101y=15 \\ -x+2y=3 \end{array} \right. \\ \hline 103y=18 \end{array}$	<pre>"+" left lbrace alignl stack{ x+101 y=15# -x+2 y=3} right none newline <b>alignr overstrike stack{ phantom {x+101 y=15q}}</b> newline alignl 103 y=18</pre>

В выделенной полужирным начертанием команде использованы невидимые символы для указания длины линии. Очевидно, что линию нужно выровнять по правому краю (alignr) и сделать по длине чуть больше самого длинного уравнения (в данном случае -- верхнего).

Выражение под командой phantom представляет собой самое длинное уравнение системы с добавленным символом q (на его месте может быть другой символ, несколько символов или символы могут отсутствовать вообще, если удлинение линии нецелесообразно). Указание под phantom самого длинного уравнения (а не другой комбинации символов) упрощает подбор линии нужной длины, поскольку отсчет как бы ведется от известной длины.

В обоих случаях имеется одна и та же проблема -- выравнивание итога (103y = 18). Маловероятно, что alignc или alignr случайно обеспечат нужное выравнивание, поэтому придется выравнивать с помощью пробелов (обратный апостроф или гравис ` и тильда ~) и невидимых символов. Например, чтобы выровнять итог по левому краю уравнений системы, можно воспользоваться невидимыми символами.



Пример 28

$+ \begin{cases} x+101y=15 \\ -x+2y=3 \end{cases}$ <hr style="width: 20%; margin: auto;"/> $103y=18$	<pre>"+" <b>left lbrace</b> alignl stack{ x+101 y=15# -x+2 y=3} <b>right none</b> newline alignr overstrike stack{phantom {x+101 y=15q}} newline alignl <b>phantom{"+" left lbrace right none}</b> 103 y=18</pre>
--	---

Сопоставьте выделенные полужирным начертанием участки кода -- и все поймете.

Рассмотрим более сложное почленное выравнивание в уравнении. Грубо говоря,  $x$  находится под  $x$ ,  $y$  -- под  $y$  и т.д. Здесь надо использовать две матрицы: одну -- для системы уравнений, другую -- для итоговой строки.

Пример 29

$+ \begin{cases} x +101y =15 \\ -x +2y =3 \end{cases}$ <hr style="width: 20%; margin: auto;"/> $103y =18$	<pre>"+" <b>left lbrace</b> alignr matrix{ x#+101 y#alignl {}=15## -x#+2 y#alignl {}=3} <b>right none</b> newline alignr overstrike phantom {matrix{ x#+101 y#alignl {}=15qq}} newline alignl phantom{"+" <b>left lbrace</b> right none} {alignr matrix { <b>phantom -x#~103 y#alignl {` `}=18}}</b></pre>
---	--

Может возникнуть вопрос: а зачем последняя строка записана с помощью матрицы? Можно было написать обычную строчку и выровнять ее вручную. Да, так можно сделать, но в описанном случае выравнивать элементы матрицы проще: доставлять пробелы и невидимые символы в отдельных элементах матрицы удобнее, поскольку тогда нужно привязываться только к границам соответствующей ячейки матрицы, а не ко всей строке.

Для выравнивания «по переменной» используется alignr, для расположения друг под другом знаков равенства -- alignl. Таким образом обеспечивается независимость записи системы уравнений от длины этих уравнений или коэффициентов при переменных.

Недостаток этого способа в том, что, несмотря на идеальное выравнивание самих уравнений, приходится подбирать длину горизонтальной линии (как и в предыдущем случае), а также отступы для итоговой строки (в указанном примере это сделано с помощью коротких пробелов).

Последний недостаток можно устранить в том случае, если допустимо оставить немного места

после итоговой строки. Понятно, что размер матрицы автоматически подбирается по самым длинным ее элементам. Суть предлагаемого мною метода заключается в том, чтобы задать равные размеры матриц системы уравнений и итоговой строки, а именно -- указать размер итоговой строки по наиболее длинной строке системы уравнений. Для этого удобно использовать команду phantom!

Пример 30

$+ \begin{cases} x + 101,238y = 15 \\ -x + 1,762y = 3 \end{cases}$ <hr style="width: 20%; margin-left: auto; margin-right: auto;"/> $103y = 18$	<pre> "+" left lbrace alignr matrix{ x#+101,238 y#alignl {}=15## -x#+2 y#alignl {}=3} right none newline alignr overstrike phantom {matrix{ x#+101,238 y#alignl {}=15qq}} newline alignl phantom{"+" left lbrace right none} {alignr matrix { phantom -x# ~ 103 y#alignl {}=18## phantom x# phantom { + 101,238 y}# phantom{alignl {}=15}}}</pre>
---	---

Мы просто «зафантомили» самую длинную строку и расположили ее под итоговой строкой, точнее, добавили к матрице итоговой строки. Именно для нее и требуется дополнительное место.

Должен признать, что одну задачу мне так и не удалось решить. Что делать, когда итоговая строка длиннее всех уравнений системы? Конечно, такая ситуация маловероятна, но все же. Видимо, суть решения задачи аналогична: добавить над системой уравнений скрытую строку, равную по длине итоговой строке, причем вписать скрытую строку в ту же матрицу (или stack), в которой записана система уравнений. Сама запись несложна, но фигурную скобку слева мне не удается поставить корректно, т.е. так, чтобы она обрамляла только уравнения, но не скрытую строку. При этом отделить скрытую строку от матрицы системы уравнений нельзя, поскольку тогда теряется эффект подбора ширины (за который и нужно бороться).

Из этой ситуации есть три выхода. Первый -- подобрать нужную ширину с помощью пробелов и скрытых символов, что не универсально и зачастую неудобно. Вторым -- не ставить слева знак + и фигурную скобку, но тогда нарушается весь смысл записи и она превращается просто в запись матрицы из нескольких строк, первая из которых скрыта. Третий выход подразумевает дорисовывание фигурной скобки внешними средствами, например, с помощью Math как редактора формул в Writer (как обычно и бывает) можно дорисовать фигурную скобку, выбрав соответствующий значок на панели инструментов «Рисунки».

\* \* \*

Можно найти еще много случаев, когда разметка Math сложна и малопонятна. Конечно, нельзя охватить сразу их все, но Math тем и хорош, что если понять логику и принципы его работы, то с его помощью удастся легко набирать практически любые математические формулы.

Связаться с автором можно по e-mail: [dma2002@hotmail.ru](mailto:dma2002@hotmail.ru).