

ЯЗЫКИ РАЗМЕТКИ.

ЧАСТЬ 4: ВЁРСТКА МАТЕМАТИЧЕСКИХ ФОРМУЛ

Аннотация

В четвертой из серии статей о языках разметки автор рассматривает верстку математических формул на языке LaTeX. Являясь действующим математиком, автор обращает внимание в основном на те аспекты, которые он использовал при наборе собственных статей. В отличие от книги Львовского, автор рассматривает некоторые нестандартные возможности, которые улучшают качество текста, однако делает это в гораздо меньшем объеме, чем в фундаментальной книге Миттельбаха, Самарина и Гуссенса.

Ключевые слова: TeX, LaTeX, верстка математических формул.

И все-таки TeX был создан математиком для набора текста с математическими формулами! Изучением того, как это делать, мы сейчас и займемся. Так как набор математических формул в HTML крайне неудобен, а язык MathML так и не стал общепринятым стандартом (по крайней мере, не поддерживается почти ни одним обозревателем), то ставшие уже привычными экскурсы в HTML в этой статье отсутствуют.

МАТЕМАТИЧЕСКИЙ РЕЖИМ

Все математические формулы в LaTeX должны быть набраны между знаками « $\$$ » или « $\$\$$ » или внутри соответствующих окружений, о которых мы поговорим позже. При этом в « $\$\$$ » заключаются *выделенные формулы*, то есть формулы, вынесенные в отдельную строку. По умолчанию выделенные формулы центрируются, однако это поведение можно переопределить в стилевом файле, что и сделано в некоторых математических журналах.

Вообще, в процессе работы TeX может находиться в одном из шести режимов, называемых модами:

- горизонтальная мода – форматирование абзаца;
- внутренняя горизонтальная мода – форматирование строки внутри `\hbox`;
- внутренняя вертикальная мода – форматирование вертикального материала внутри `\vbox`;
- математическая мода – форматирование математических формул внутри « $\$$ »;
- выделенная математическая мода – форматирование математических формул внутри « $\$\$$ ».

Поведение TeX в математической и выделенной математической моде немного отличается: например, числитель и знаменатель дроби в выделенной формуле набирается основным шрифтом, а в обычной –



... горизонтальная мода ... вертикальная мода ...
математическая мода...

уменьшенным; пределы интегрирования в выделенной формуле располагаются над и под знаком интеграла, а в обычной – как нижний и верхний индексы и т. п. Опуская эти незначительные детали, будем обе эти моды называть математическим режимом. Основные отличия математического режима от горизонтальной моды состоят в следующем.

- Пробелы в математической моде игнорируются, TeX сам расставляет пробелы в соответствии со стандартами, описанными в стилевом файле.

- По умолчанию, формулы набираются шрифтом «математический курсив», который отличается от обычного курсива увеличенными промежутками между буквами, а также тем, что цифры взяты из прямого шрифта. Переключение шрифтов в математическом режиме происходит совершенно по-другому, чем в горизонтальной моде, и будет рассмотрено отдельно.

- Символы нижнего и верхнего индекса « $_$ » и « $^$ » и некоторые макросы, например названия греческих букв, запрещено употреблять вне математического режима. При неправильном употреблении вы получите сообщение об ошибке «Missing \$ inserted...», что означает, что, пожалев вас, TeX вставил пропущенный вами знак доллара в нужное (по его мнению) место текста и предлагает продолжить работу. Вскоре, правда, оказывается, что для нормальной работы ему надо еще что-то вставить, и количество ошибок растет, как снежный ком.

СТАНДАРТНЫЕ СРЕДСТВА

Простейшие формулы набирать очень просто. Для набора нижнего индекса используйте знак « $_$ », а верхнего – « $^$ », при этом то, что должно появиться в индексе, должно быть либо одним токеном (символом или макросом) либо заключено в фигурные скобки, например: x^2 , x_{γ} , $(xy)^{12}$, $x^{y^{zt}}$. На печати это будет выглядеть следующим образом:

$$x^2, x_{\gamma}, (xy)^{12}, x^{y^{zt}}.$$

Дроби набираются командой `\frac` с двумя аргументами: числитель и знаменатель. Например, $\frac{12}{a^2+bc}$ или $\frac{ab}{a^2+\frac{bc}{}}$. Конструкция `\over` Plain TeX, например $a^2+\frac{b}{c}$, у LaTeX в опале: он уже много лет предупреждает, что в следующей версии она поддерживаться не будет.

Корни производятся конструкцией `\sqrt[n]{xxx}`, где n – необязательный параметр – показатель степени, а xxx – подкоренное выражение. Альтернативный вариант Plain TeX – `\root n\of{xxx}` тоже работает.

Штрихи ставятся непосредственно с помощью символа « $'$ », при этом символ верхнего индекса использовать не надо. Аналогичный символ посередине строки называется `\prime`, так что конструкция $f'(x)$ эквивалентна $f^{\prime}(x)$.

Многоточия набираются с помощью макросов `\dots`, `\ldots` и `\cdots`. При этом `\ldots` ставит многоточие по базовой линии, `\cdots` – по центру, а `\dots` – по умолчанию, в зависимости от знака, следующего после `\dots`. В американском стандарте многоточие в сумме ставится по центру, в русском – по базовой линии, но это не должно вас заботить, это дело стилевого файла. Таким образом, макрос `\dots` – это еще один пример логической разметки. Аналогичная ситуация со знаками нестрогих неравенств – знаки `\leq` и `\geq` (less than or equals to, greater than or equals to) производят американские знаки \leq и \geq , `\leqslant` и `\geqslant` – принятые в России $\leq\!\!\!\!\!\leq$ и $\geq\!\!\!\!\!\geq$, а `\le` и `\ge` – знаки, используемые по умолчанию (при этом в стандартных стилях умолчания, конечно, американские). Для использования русских вариантов знаков неравенств необходимо подключение стилевого пакета `amssymb`.

Имена стандартных математических функций принято набирать прямым шрифтом с соответствующими отступами от предыдущей части формулы и от аргументов. Поэтому вместо $\sin x$ (что будет выглядеть просто безобразно, потому что еще и пробел между n и x исчезнет) нужно писать $\$ \sin x \$$ и т. п. Будьте осторожны: в стандартных нерусифицированных стилях не определе-

ны макросы `\tg`, `\ctg`, `\arctg` и `\arcctg`, потому что в англоязычных странах они называются `\tan` и т. д. Нет никаких проблем определить новую функцию самому. В LaTeX для этого предназначена команда `\DeclareMathOperator{macros}{name}`, где `\macros` – это имя определяемого макроса, а `name` – имя функции. Но в большинстве случаев то, что вам захочется, будет уже определено, например: `\lim`, `\min`, `\sup`, `\det` и другие.

Греческие буквы, которые традиционно любят использовать математики (собственно только потому, что не хватает латинских) обозначаются очевидными макросами: `\alpha`, `\beta`, `\gamma`, ..., `\omega`. Однако для некоторых букв используется альтернативное начертание, более привычное русскому глазу. Я использую две такие буквы: `\varphi` – φ , в отличие от `\phi`, которая больше похожа на знак пустого множества, и `\varepsilon` – ε , в то время как `\epsilon` больше смахивает на знак принадлежности. Есть еще `\vartheta`, но она мне менее привычна, чем обычная `\theta` – θ . Кстати, во всех международных математических журналах греческие буквы набираются курсивом (точнее, математическим курсивом), что бы вам ни говорили редакторы нематематических издательств.

Довольно часто математики используют буквы с акцентами, всякими там черточками, волнами или домиками. Для этого в TeX предназначены команды `\bar`, `\tilde`, `\hat` и их широкие аналоги `\overline`, `\widetilde` и `\widehat`. Для подчеркивания текста используется команда `\underline`. Все эти команды имеют один аргумент, а именно текст, над которым ставится акцент (или то, что подчеркивается). В Plain TeX вне математического режима подчеркивание не работает, так что для подчеркивания обычного текста приходится использовать конструкцию

$$\$\underline{\hbox{text}}\$,$$

где `text` – это то, что хочется подчеркнуть. К счастью в LaTeX эта неприятность отменена, так что доллары и `\hbox` можно



Для подчеркивания текста используется команда `\underline`.

не писать. Несмотря на отсутствие `\hbox`, переноситься на следующую строку подчеркнутый текст все равно не будет.

Специалистам по линейной алгебре следует знать окружения `matrix`, `pmatrix`, `bmatrix`, `vmatrix` и `Vmatrix`, с помощью которых набираются матрицы. При этом `matrix` печатает матрицу без скобок, `pmatrix` – в круглых скобках, `bmatrix` – в квадратных, `vmatrix` – в прямых (стандартное обозначение определителя матрицы), а `Vmatrix` – в двойных прямых скобках. Надо отметить, что правильно эти окружения работают только при подключении пакета `amsmath` (который автоматически подключается при использовании стилей AMS). Матрица, с точки зрения TeX, является частным случаем таблицы, поэтому внутренность матрицы набирается так же, как и содержимое таблицы, то есть колонки разделяются символом `&`, а строки – символами `\\`. При этом объединение столбцов, как и в таблицах, достигается с помощью команды `\multicolumn{n}{c}{...}`, где `n` – количество объединяемых столбцов, `c` – центрирование, а многоточие заменяет текст объединенной ячейки. В матрицах неопределенного размера довольно часто используют многоточие. Для этого, помимо уже известных `\dots`, `\ldots` или `\cdots`, можно использовать макросы `\vdots` для вертикального многоточия и `\ddots` – для диагонального. Для создания многоточия, захватывающего несколько столбцов, рекомендуется¹ использовать команду

$$\backslash\hdotsfor[spacefactor]{number},$$

¹ Честно говоря, я узнал об этой рекомендации только в момент написания этой статьи из книги «LaTeX II». До этого я употреблял конструкцию `\multicolumn{number}{c}{\dotfill}`, которая вполне жизнеспособна.

где **spacefactor** – необязательный параметр – коэффициент, определяющий расстояние между соседними точками, а **number** – количество столбцов.

ТАБЛИЦА СИМВОЛОВ

Наконец, приведу список символов, определенных в LaTeX, которыми я пользуюсь. Более полную таблицу можно найти в книге Львовского. Кроме того, чтобы как-то выделиться из толпы авторов, воспроизводящих эту таблицу, я напишу код на языке LaTeX, используя который вы сами сможете получить расширенную распечатку этой таблицы. Для этого в файле **ShowMacro.tex** напишите следующие макроопределения (см. листинг 1).

После подключения этого файла командой `\input` становится доступной команда `\ShowMacroses`, аргумент которой ограничен символом «;», а в этом аргументе через запятую написаны имена макросов (без об-

ратной кривой чертой), значения которых хочется увидеть в таблице. Например, файл **macroses.tex** (см. листинг 2) произведет на свет первые две строки следующей таблицы (см. рис. 1).

У некоторых бинарных операций есть **big**-аналоги, например `\bigoplus` или `\bigcap`. У многих стрелок есть варианты с префиксом `long` или начинающиеся с прописной буквы, например `\Longlefttrightharpoon` равносильна `\iff`.

Забегая вперед, не могу не сказать несколько слов о смысле того, что написано в файле **ShowMacro.tex**, хотя это и не относится к верстке математических формул. В определении команды **ShowMacroses** происходит инициализация макропоследовательностей `\Template` (строка-образец), `\rowI` и `\rowII` (верхняя и нижняя строки таблицы), затем вызывается команда `\MakeTableBody`, которая создает шапку и текст таблицы `\TableBody`, а потом выводит таблицу с помощью команды `\halign`.

Листинг 1

```
\makeatletter
\def\x{####}
\def\t{\tabskip=0pt plus 1fil}
\def\Halign{\halign to \textwidth }
\def\ShowMacroses#1,#2;{\par\smallskip
  {\baselineskip=0pt\offinterlineskip\tabskip=0pt
  \gdef\Template{\vrule height13pt depth4pt\x\hfill\t
    &\hfill\,\x\,\hfill\t}%
  \gdef\rowI{&\texttt{\backslash$#1}}%
  \gdef\rowII{&\expandafter\csname #1\endcsname$}%
  \MakeTableBody#2,,%
  \expandafter\Halign \expandafter{\TableBody }}
\par\medskip }
\def\MakeTableBody#1,{\def\tmp{#1}%
  \ifx\tmp\empty
    \g@addto@macro{\Template}{&\tabskip=0pt\hfill\x\vrule\cr}
    \xdef\TableBody{\Template}
    \g@addto@macro{\TableBody}{%
      \hline\rowI &\cr \hline \rowII &\cr \hline }
  \else
    \g@addto@macro{\Template}{&\vrule\t\x\hfill &\t\hfill\,\x\,\hfill}
    \g@addto@macro{\rowI}{&\texttt{\backslash$#1}}%
    \g@addto@macro{\rowII}{&\expandafter\csname #1\endcsname$}%
    \expandafter\MakeTableBody
  \fi }
\makeatother
```

Листинг 2

```
\documentclass[10pt]{amsart}
\usepackage{amssymb}
\input ShowMacro
\begin{document}
\ShowMacroses pm,mp,times,cdot,star,ast,circ,bullet,setminus,%
cap,cup,oplus,otimes,lnot,lor,land;
\end{document}
```

Определение `\MakeTableBody` работает рекурсивно, создавая для каждого вновь прочитанного названия макроса столбец таблицы, а при выходе составляет их вместе в `\TableBody`. Прimitив `\expandafter` говорит TeXy: подожди один шаг с раскрытием следующего токена, раскрой сначала то, что идет следом, а потом вернись обратно. Имя команды `\g@addto@macro` говорит само за себя: она добавляет к определению своего первого параметра все, что написано во втором. `\ifx` – это оператор сравнения однократного раскрытия макропоследовательностей: если они равны (в нашем случае, если строка закончилась), то выполняется текст до `\else`, в противном случае – от `\else` до `\fi`. Самая большая проблема при написании этого файла состояла в том, что, читая макроопределение, TeX автоматически подставляет вместо `#1, #2, ...` фактические параметры, а вместо `##` – `#` (одиночный сим-

вол `#` внутри макроопределения запрещен). Поэтому пришлось определить временный макрос `\x`, который раскрывается только при выполнении команды `\xdef`. В отличие от обычного `\def` или `\gdef`, команды `\edef` и `\xdef` сначала раскрывают то, что стоит в фигурных скобках до примитивов, а потом определяют макропоследовательность.

ГРУППЫ МАТЕМАТИЧЕСКИХ СИМВОЛОВ

Каждый математический символ, точнее часть математической формулы, принадлежит к одной из 7 групп:

- 1) обычный символ,
- 2) бинарная операция (`\mathbin`),
- 3) бинарное отношение (`\mathrel`),
- 4) математический оператор (`\mathop`),
- 5) скобка,

<code>\pm</code>	<code>\mp</code>	<code>\times</code>	<code>\cdot</code>	<code>\ast</code>	<code>\+</code>	<code>\circ</code>	<code>\bullet</code>	<code>\setminus</code>	<code>\cap</code>	<code>\cup</code>	<code>\oplus</code>	<code>\otimes</code>	<code>\minus</code>	<code>\vee</code>	<code>\wedge</code>
<code>\neq</code>	<code>\sim</code>	<code>\cong</code>	<code>\approx</code>	<code>\equiv</code>	<code>\ll</code>	<code>\gg</code>	<code>\parallel</code>	<code>\perp</code>	<code>\lhd</code>	<code>\unlhd</code>	<code>\rhd</code>	<code>\triangle</code>	<code>\nabla</code>		
<code>\in</code>	<code>\notin</code>	<code>\ni</code>	<code>\subset</code>	<code>\supset</code>	<code>\subseteq</code>	<code>\supseteq</code>	<code>\forall</code>	<code>\exists</code>	<code>\emptyset</code>	<code>\varnothing</code>	<code>\infty</code>				
<code>\langle</code>	<code>\rangle</code>	<code>\sum</code>	<code>\prod</code>	<code>\coprod</code>	<code>\int</code>	<code>\iint</code>	<code>\int</code>	<code>\partial</code>	<code>\angle</code>	<code>\aleph</code>	<code>\hbar</code>	<code>\Box</code>	<code>\S</code>		
<code>\to</code>	<code>\rightarrow</code>	<code>\Rightarrow</code>	<code>\leftarrow</code>	<code>\uparrow</code>	<code>\downarrow</code>	<code>\searrow</code>	<code>\nearrow</code>	<code>\swarrow</code>	<code>\nwarrow</code>						
<code>\longrightarrow</code>	<code>\Longrightarrow</code>	<code>\Leftrightarrow</code>	<code>\iff</code>	<code>\hookrightarrow</code>	<code>\mapsto</code>	<code>\rightarrowtail</code>									

Рис. 1

- б) знак препинания,
- 7) подформула.

От того, к какой группе относится тот или иной символ (последовательность символов), зависит, какие пробелы TeX будет расставлять вокруг него и как он будет расставлять индексы. При этом вы всегда можете поменять умолчание, поставив перед группой символов название группы, например, написав `$a\mathbin{\text{trm}{div}}$`, вы заставите TeX набрать `div` прямым шрифтом и оставить небольшие пробелы до и после этого слова, как будто оно является бинарной операцией. В древнем AmS-TeX для определения математических функций употреблялось слово `\operatorname`, например, для определения русского обозначения тангенса нужно было написать

```
\def\tg{\operatorname{tg}}.
```

Само же слово `\operatorname` было определено следующим образом:

```
\def\operatorname#1{\mathop{\hbox{\rm #1}}\nolimits}
```

Наиболее часто употребляются слова `\mathop` и `\mathrel`. Для чего? Читайте следующий раздел.

РАСПОЛОЖЕНИЕ ИНДЕКСОВ

По умолчанию после всех символов, кроме математических операторов в выделенной математической моде, индексы ставятся чуть правее самого символа. Иногда хочется поставить индекс непосредственно над или под символом, как это делается для математических операторов в выделенной математической моде. Для математических операторов это сделать совсем легко: надо только написать команду `\limits` между оператором и индексами, например, `$\lim\limits_{x\to\infty}$`. В то же время, чтобы ставить индексы справа от сим-

вола, надо написать `\nolimits`. Однако, написав `\limits` после символа, не являющегося математическим оператором, вы получите сообщение об ошибке. Как же написать что-то над знаком равенства или эквивалентности? В LaTeX для этого предусмотрена команда `\stackrel`, первым аргументом которой является верхний индекс, а вторым – знак отношения. Например, `$a\stackrel{\nu}{\sim} b$` выйдет на печати следующим образом:

$$a \overset{\nu}{\sim} b .$$

Что же делать, однако, если вы хотите поставить что-то под знаком равенства или, не дай бог, вторую надпись под знаком предела? Ничего страшного, хотя и несколько громоздко, пишете:

```
$\mathrel{\mathop{=} \limits_{\text{по 2.3}}}$ \ и \
$\mathop{\mathop{\lim\limits_{x\to0}}
\limits_{y\to0}}f(x,y)$,
```

чтобы получить

$$x^2 - y^2 \underset{\text{по 2.3}}{=} (x - y)(x + y) \text{ и } \lim_{\substack{x \rightarrow 0 \\ y \rightarrow 0}} f(x, y)$$

(если это именно то, что вы хотите). Обратите внимание, что команда `\text` доступна только при подключении пакета `amsmath`.

СКОБКИ

Различные виды (левых) скобок обозначаются токенами `(`, `[`, `|`, `\{`, `\langle` (если вы хотите использовать вертикальную черту как знак бинарного отношения, пишете `\mid`, иначе пробелы будут расставлены неправильно). Каждую скобку можно напечатать в одном из пяти предопределенных размеров: обычный, `\big`, `\Big`, `\bigg` и `\Bigg`. Кроме того, можно попросить TeX выбрать подходящий размер скобок, окружающих некоторую формулу. Для этого перед открывающей скобкой нужно написать

Листинг 3

```
Пусть $y=f\big(g(x)\big)$, \
$z=\left(1+\frac{\beta}{1+\frac{\gamma}{\delta}}\right)^2$ и
$\left\{\begin{array}{l} u=y+z \\ v=y-z \end{array}\right.$.
```

`\left`, а перед закрывающей – `\right`. Приведем примеры стандартного использования изменения размера скобок. Если написать (листинг 3), то на печати получим следующее (см. рис. 2).

$$\text{Пусть } y = f(g(x)), \quad z = \left(1 + \frac{g}{1+\frac{g}{b}}\right)^2 \text{ и } \begin{cases} u = y + z \\ u = 2y - z \end{cases}.$$

Рис. 2

Обратите внимание, что команды `\left` и `\right` должны обязательно идти парами, даже если вы не собираетесь ставить закрывающую или открывающую скобку, потому что TeX должен знать, где начинается и заканчивается формула, под которую подбирается размер скобок. Специально для такой ситуации предусмотрена «пустая скобка», которая обозначается точкой: `\left.` или `\right.`.

Используя пакет `amsmath`, вы можете ставить скобки не только справа или слева, но и над и под частью формулы. Это работает аналогично подчеркиванию с помощью команд `\underbrace` и `\overbrace`. Например, `\underbrace{a\cdots\cdots a}` выдаст $\underbrace{a \cdots \cdots a}$, а `\overbrace{ABC}` будет напечатано как \overbrace{ABC} . Скобки, правда, здесь предусмотрены только фигурные, зато вместо скобок можно поставить стрелки с помощью команд `\overrightarrow`, `\overleftarrow`, `\underrightarrow` и `\underleftarrow`. Заметим, что индексы к конструкции `\underbrace{\dots}` или `\overbrace{\dots}` будут поставлены над или под выражением, а не справа, как обычно. Например,

```
\underbrace{a+\dots+a}_{\text{\$n\$ слагаемых}}\$
```

на печати превратится в $\underbrace{a + \dots + a}_{n \text{ слагаемых}}$.

ВЫРАВНИВАНИЕ И НУМЕРАЦИЯ ФОРМУЛ

Выравнивание внутри математической формулы может быть организовано с помощью окружения `array`, которое работает аналогично окружению `tabular`, подробно рассмотренному в предыдущей статье цикла. Частными случаями этого окружения являются окружения типа `matrix`, рассмотренные выше, а также окружение `cases`,

обычно употребляемое при наборе формул с разбором случаев, например:

```
\f(x)=\begin{cases}1, & x \in D \\ 0, & \text{\textit{иначе}}\end{cases}
```

преобразуется в

$$f(x) = \begin{cases} 1, & x \in D \\ 0, & \text{иначе} \end{cases}$$

и определяет характеристическую функцию (индикатор) множества D .

Для выравнивания и автоматической нумерации формул в LaTeX предусмотрены следующие окружения:

- `equation` – нумерация без выравнивания;
- `gather` – по одной центрированной формуле в каждой строке с нумерацией каждой строки;
- `align` – несколько формул в каждой строке с нумерацией каждой строки;
- `aligned` – то же, что и `align`, но для подформулы (и, естественно, без нумерации);
- `multline` – расположение формул лесенкой (используется для расположения одной формулы на нескольких строках);
- `\split` – разбивка формулы на строки с выравниванием (работает только внутри окружений `equation` и `equation*`).



...multline — расположение формул лесенкой...

Все приведенные окружения, кроме `equation`, определены в пакете `amsmath`. Без подключения этого пакета в LaTeX предусмотрены другие окружения, но они работают несколько хуже (не везде пробелы соответствуют типографским правилам), поэтому мы их не упоминаем. Кроме окружений `aligned` и `split`, все остальные автоматически переводят TeX в выделенную математическую моду, поэтому доллары вокруг них ставить не надо (и нельзя). Окружения, предусматривающие нумерацию, имеют варианты со звездочкой, которая подавляет нумерацию. Для того чтобы отменить нумерацию в конкретной строке, непосредственно перед символами `\` нужно написать команду `\notag`. Можно пометить формулу своей меткой, для чего перед `\` написать `\tag{метка}`. Впрочем, лучше один раз увидеть, как это работает, чем сто раз услышать. Пишем (листинг 4). Получаем рис. 3.

(1)	$E = mc^2$	
(2)	$7 \cdot 9 = 63$	$63 : 9 = 7$
(2')	$5 \cdot 12 = 60$	$60 : 5 = 12$
(3)	$\begin{cases} x^2 + y^2 = a \\ x + y = b \end{cases}$	
(4)	$1 + 2 + 3 + 4 + \dots \\ + 46 + 47 + 48 + \dots \\ + 98 + 99 + 100 = 5050$	
	$\begin{aligned} x^4 + y^4 &= (x^4 + 2x^2y^2 + y^4) - 2x^2y^2 \\ &= (x^2 + y^2)^2 - (\sqrt{2}xy)^2 = \\ &= (x^2 + y^2 + \sqrt{2}xy)(x^2 + y^2 - \sqrt{2}xy) \end{aligned}$	
(5)	$X + Y = a_1$	
(6)	$X^2 + Y^2 = a_2$	
	\vdots	
(7)	$X^n + Y^n = a_n$	

Рис. 3

Листинг 4

```
\documentclass[10pt]{amsart}
\usepackage[russian]{babel}
\usepackage[left=2cm,right=12cm,top=1cm,bottom=1.5cm]{geometry}
\usepackage{amssymb}

\begin{document}
\begin{equation}E=mc^2\end{equation}
\hrule
\begin{align}7\cdot 9&=63 & 63:9&=7\\
5\cdot 12&=60 & 60:5&=12\tag{\$2'\$}\end{align}
\hrule
\begin{equation}
\left\{\begin{aligned}x^2+y^2&=a \\ x+y&=b\end{aligned}\right.
\end{equation}
\hrule
\begin{multline*}
1+2+3+4+\cdots \\ +46+47+48+\cdots \\ +98+99+100=5050
\end{multline*}
\hrule
\begin{equation*}\begin{split}
x^4+y^4&=(x^4+2x^2y^2+y^4)-2x^2y^2 \\
&=(x^2+y^2)^2-(\sqrt{2}xy)^2= \\
&=(x^2+y^2+\sqrt{2}xy)(x^2+y^2-\sqrt{2}xy)
\end{split}
\end{equation*}
\hrule
\begin{gather}X+Y=a_1 \\ X^2+Y^2=a_2 \\ \vdots \\ X^n+Y^n=a_n\end{gather}
\end{document}
```


Я бы, правда, предпочел все формулы выравнивать по знаку равенства, а вместо `aligned` воспользовался бы `array`, поставив x и y в разных формулах друг под другом, но это вопрос вкуса. Обратите внимание также на то, что вместо долларов эта формула заключена в окружение `equation`, для того чтобы получить номер.

ПЕРЕКЛЮЧЕНИЕ ШРИФТОВ

Переключение шрифтов в формулах LaTeX2 ϵ , в отличие от Plain TeX и от предыдущей версии LaTeX2.09, устроено совершенно по-другому. Все шрифтовые команды имеют один аргумент – текст, который будет напечатан этим шрифтом, при этом большая часть начинается с приставки `\math`. Это сделано потому, что чаще всего в математических формулах надо набрать выделенным шрифтом один символ, и потому, что размер шрифта TeX выбирает сам в зависимости от местоположения символа. Вместе с шрифтами, определенными в пакете `amsfonts`, доступны следующие шрифтовые команды:

- `\mathrm`, `\mathbf`, `\mathit`, `\mathsf` с очевидными значениями (последняя выбирает шрифт без засечек – Sans Serif), при этом, по умолчанию, как вы знаете, используется `\mathit`;

- `\mathbb` – Blackboard Bold – изначально являлся заменой жирного шрифта на доске, в современной математике используется для обозначения множеств натуральных, рациональных, вещественных и комплексных чисел, а также некоторых других базовых в той или иной области множеств;

- `\mathrrak` – готический шрифт;
- `\mathcal` – рукописный шрифт (только для прописных букв);

- `\EuScript` – рукописный шрифт с «завитушками» (доступен после подключения пакета `euscript` и только для прописных букв);

- `\boldsymbol` – делает жирным любой символ, который умеет (`\mathbf` действует только на латинские буквы);



...`\boldsymbol` – делает жирным любой символ, который умеет...

- `\boldmath` – делает жирным все, что умеет; в отличие от других команд, дается перед формулой (вне математического режима) и действует до завершения группы, то есть до закрывающей фигурной скобки;

- `\pmb` (Poor Man's Bold, то есть «жирный шрифт для бедных») – действует на один символ, причем делает жирными даже те символы, для которых нет жирных вариантов, путем печати нескольких одинаковых символов рядом (рекомендуется по возможности не использовать);

- `\text` – восстанавливает те шрифтовые атрибуты, которые были определены в текстовом режиме перед формулой, не игнорирует пробелы.

Вот строка символов, набранных всеми приведенными шрифтовыми командами через запятую:

```
mathrm, X ∈ mathbf, mathit, mathsf,
ℝ, mathfrak ℄, MATHECAL, EUSCRIPT,
α · α, ∑n=1∞ an, ∑n=1∞ an,
f(x) = 0 почти везде.
```

Как видите, `\mathbf` не повлиял на начертание знака принадлежности (впрочем, в реализации TeX, установленной на моем компьютере, `\boldsymbol\in` тоже не работает), `\boldmath` не сделала знак сум-

мы жирным, а вот команде `\pmb` это удалось.

Для того чтобы изменить размеры шрифтов в математических формулах по всему документу, предназначена команда `\DeclareMathSizes`, которая имеет 4 аргумента:

- 1) размер текущего текстового шрифта;
- 2) размер основного шрифта для формул;
- 3) размер шрифта для индексов;
- 4) размер шрифта для повторных индексов.

При этом все размеры указываются в пунктах (`pt` не пишется). Естественно, если вы собираетесь использовать несколько размеров основного текстового шрифта, например более крупный в заголовках и более мелкий в сносках, то вы должны дать несколько команд `\DeclareMathSizes`. Естественно, место этим командам – в стиле-вом файле, в крайнем случае, в преамбуле документа. Например, некоторые редакции, имеющие не самую новую множительную технику, требуют увеличить размер повторных индексов по сравнению со стандартом TeX, потому что иначе они просто сливаются в точку. Для этого достаточно дать команды

```
\DeclareMathSizes{10}{10}{8}{6}
% для основного шрифта 10pt
\DeclareMathSizes{8}{8}{7}{6}
% для сносок
```

Конечно, изменение соотношений размеров индексов нежелательно, так как непривычно для читателя, но это лучше, чем индексы, которые невозможно разобрать. В стандарте LaTeX аналогичные команды выглядят следующим образом:

```
\DeclareMathSizes{10}{10}{7}{5}
% для основного шрифта 10pt
\DeclareMathSizes{8}{8}{6}{5}
% для сносок
```

ЧТО ЕЩЕ БЫВАЕТ?

Можно написать еще столько же и не исчерпать тонкости набора математических формул в TeX. С другой стороны, большей

части читателей будет вполне достаточно и того, что уже написано, кроме того, мои силы и возможности журнала не беспредельны. Отсылая интересующихся к книгам «Набор и верстка в системе LaTeX» С.М. Львовского и «Путеводитель по пакету LaTeX и его расширению LaTeX2e» (краткое название – «LaTeX II») М. Гуссенса, Ф. Миттельбаха и А. Самарина, упомяну, какими еще возможностями я пользуюсь.

Несложные коммутативные диаграммы удобно набирать с помощью окружения `CD`, определенного в пакете `amscd`. Еще больше возможностей – диагональные, криволинейные и пересекающиеся стрелки – предоставляет пакет `xypic`, но за удобство надо платить: он работает довольно медленно. С несложными картинками типа эскизов графиков функций вполне справляется окружение `picture`. Все кривые рекомендуется рисовать отрезками парабол, которые задаются командой `\qbezier` (кривые Безье). Одно над другим можно набирать с помощью команды `\atop` (грубо говоря, получается дробь без дробной черты), не обязательно каждый раз писать таблицу.

Для расстановки индексов по всем четырём углам символа удобно использовать `\sideset`, а вот поставить индекс непосредственно над или под символом (даже если это не математический оператор) позволяют команды `\overset` и `\underset`.

Иногда бывает необходимо обмануть TeX по поводу размера подформулы. Для этого используются команды `\smash`, `\lefteqn` и `\phantom` с вариантами `\hphantom` и `\vphantom`. Эти команды либо печатают формулы, не отводя для них места по ширине, высоте или глубине, либо отводят место для формулы, ничего не печатая. Дополнительные (отрицательные или положительные) пробелы задаются командами `\! \`, `\: \`; `\quad` и `\qquad`.

Как положено оформлять теоремы (и другие формулировки) с помощью пакета `amsthm`, будет изложено в следующей статье цикла, посвященной автоматической нумерации и (гипер)ссылкам.

Литература

1. Львовский С.М. Набор и верстка в системе LATEX. М: МЦНМО, 2006. 448 с.
2. Гуссенс М., Миттельбах Ф., Самарин А. Путеводитель по пакету LaTeX и его расширению LaTeX2ε. Пер. с англ. М.: «Мир», 1999. 606 с.

Abstract

In the fourth article of the series on markup languages the author considers making-up mathematical formulas using the LaTeX language. Being an active mathematician, the author pays attention to those aspects which he uses typing his articles. Unlike the book by Lvovski the author considers some nonstandard resources, but touches them not as deep as in the book of Mittelbach, Samarin and Gussens.

*Степанов Алексей Владимирович,
доцент кафедры Высшей
математики № 2 СПбГЭТУ
«ЛЭТИ».*



Наши авторы, 2008.
Our authors, 2008.