

Набор математических формул в $\text{\LaTeX} 2_{\epsilon}^*$

Описаны все средства $\text{\LaTeX} 2_{\epsilon}$ и $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ версии 2 для печати математики в документах на английском и русском языках.

Средства разметки документа и форматирования текста, необходимые для подготовки публикаций на русском и английском языках, описаны в первой части, которая называется «Справочник по командам $\text{\LaTeX} 2_{\epsilon}$ ».

Справочник подготовлен на $\text{\LaTeX} 2_{\epsilon}$ с использованием математических шрифтов из пакета `txfonts` и текстовых PostScript шрифтов с русскими буквами TimesNewRomanPSMT, TextbookPSCyr и ERKurierPSCyr из коллекции PSCyr¹.

Содержание

1 Переключение в математическую моду	4
1.1 Формулы внутри текста	4
1.1.1 Как избежать длинных строк в тексте	5
1.2 Выделенные формулы	6
1.2.1 Однострочные уравнения	6
1.2.2 Системы уравнений	7
1.2.3 Расщепление длинных формул	8
1.2.4 Блоки уравнений	9
1.2.5 Вертикальные пробелы в многострочных формулах	10
1.2.6 Смещение номера уравнения	10
1.2.7 Разрыв многострочных формул	10
1.3 О промежутках между символами	11
1.4 О размерах символов	11
2 Математические символы	12
2.1 Показатели степени, индексы и штрихи	12
2.2 Многоточия	12
2.3 Символы бинарных операций	13
2.4 Символы бинарных отношений	13
2.5 Греческие буквы	15
2.6 Знаки пунктуации	16
2.7 Акценты	17
2.8 Корни	17
2.9 Дроби	18
2.10 Операторы с пределами	19
2.10.1 Управление расположением пределов	19
2.10.2 Интегралы	20

*© 2002 Владимир Сюткин. Замечания приветствуются: syutkin@ns.kinetics.nsc.ru

¹Шрифты из коллекции PSCyr доступны по адресу <ftp://mch5.chem.msu.su/pub/russian/psfonts/>. Ряд русских PostScript шрифтов фирмы Параграф и TrueType шрифтов поддерживает коллекция пакетов FontsC (<http://www.inp.nsk.su/people/I.A.Kotelnikov/Manual98.html>).

2.10.3	Многострочные и сторонние индексы	20
2.11	Скобки и другие разделители	21
2.11.1	Скобки переменного размера	21
2.11.2	Разделители	21
2.11.3	Разделители без пары	22
2.11.4	Команды, задающие размер разделителя	22
2.12	Стрелки	23
2.13	Неклассифицированные символы	24
2.14	Надстрочные и подстрочные знаки	25
2.14.1	Шляпки и тильды	25
2.14.2	Линии	25
2.14.3	Фигурные скобки	25
2.14.4	Стрелки	26
2.14.5	Произвольные символы	26
2.15	Стрелки с индексами	26
2.16	Биномиальные коэффициенты AMS	26
2.17	Где ещё можно найти математические символы	27
3	Математические функции	27
3.1	Функции типа логарифма	27
3.2	Функции с пределами	28
3.3	Определение новых имен операций	28
3.4	Функции модуля	29
4	Конструкции для многострочных выражений	29
4.1	Матрицы	29
4.1.1	Расчерчивание матрицы	30
4.1.2	Окаймлённая матрица	31
4.1.3	Матрицы AMS	31
4.2	Двухрядные формулы типа дроби	31
4.3	Система условий со скобкой	32
5	Шрифты	32
5.1	Включение текста в формулы	32
5.1.1	Вставка текста между уравнениями	33
5.2	Математические алфавиты	33
5.2.1	Кириллические математические алфавиты	34
5.3	Декларация нового алфавита	34
5.4	Полужирная насыщенность символов	35
6	Настройка формул	35
6.1	Промежутки в математической моде	35
6.1.1	Пробелы произвольного размера	36
6.2	Дублирование знаков при переносе формулы	36
6.3	Неразрывный дефис	37
6.4	Невидимые символы	37
6.5	Видимые символы, незанимающие места	37
7	Теоремы, законы и др.	38
8	Дополнительная нумерация уравнений	39
8.1	Нумерация уравнений вручную	39

9	Разное	40
9.1	Коммутативные диаграммы	40
9.2	Формулы в рамке	40
9.3	Команды, пригодные для любой моды	40
9.4	Изменение размеров формулы	41
9.5	Подбор размера разделителя вручную	41
	Алфавитный указатель	42

Список таблиц

1	Многоточия	12
2	Символы бинарных операций	13
3	Символы бинарных операций AMS (пакет <code>amssymb</code>)	13
4	Символы бинарных отношений	14
5	Символы бинарных отношений AMS (пакет <code>amssymb</code>)	14
6	Отрицательные бинарные отношения AMS (пакет <code>amssymb</code>)	15
7	Строчные греческие буквы	15
8	Прописные греческие буквы	16
9	Греческие буквы AMS (пакет <code>amssymb</code>)	16
10	Строчные греческие буквы прямого начертания (пакеты <code>txfonts</code> и <code>pxfonts</code>)	16
11	Символы пунктуации	16
12	Акценты математической моды	17
13	Символы переменного размера	19
14	Разделители	21
15	Большие разделители	22
16	Разделители AMS (пакет <code>amssymb</code>)	22
17	Стрелки	23
18	Стрелки AMS (пакет <code>amssymb</code>)	24
19	Отрицательные стрелки AMS (пакет <code>amssymb</code>)	24
20	Дополнительные символы	24
21	Символы, доступные в математический и в текстовом моде.	24
22	Символы AMS математический и текстовом моде (пакет <code>amfonts</code> или <code>amssymb</code>).	24
23	Дополнительные символы AMS (пакет <code>amssymb</code>)	25
24	Буквы иврита AMS (пакет <code>amssymb</code>)	25
25	Функции типа логарифма	27
26	Тригонометрические и гиперболические функции для России	28
27	Функции, принятые в России	28
28	Математические функции с пределами	28
29	Математические функции с пределами AMS (пакет <code>amsopn</code>)	28
30	Математические алфавиты	33
31	Математические алфавиты AMS	34
32	Каллиграфический алфавит RSFS (пакет <code>mathrsfs</code>)	34
33	Кириллические математические алфавиты (пакеты <code>mathtext</code> и <code>babel</code>)	34
34	Промежутки в математической моде	36

1 Переключение в математическую моду

В исходном тексте математические выражения, а также верхние и нижние индексы, выделяются специальными командными скобками для переключения Т_ЕXа в математическую моду вёрстки. В математической моде Т_ЕX игнорирует *все* пробелы между символами в исходном тексте и расставляет промежутки сам. Формулы *не* должны содержать пустых строк. Кроме латинских букв *a . . z*, *A . . Z* и цифр *0 . . 9*, распознаются и печатаются следующие символы:

+ - = * / < > () [] | . , ; ? ! : \ ' " @

Любая буква считается именем переменной и печатается шрифтом «математический курсив». Символы `^` и `_` являются служебными: они используются для набора верхних и нижних индексов, соответственно. Математические символы, а также греческие буквы, которых нет на клавиатуре, печатаются специальными командами. Имена команд обычно совпадают с названиями символов. Все они собраны в разделе 2 «Математические символы».

После подключения пакета `mathtext` согласно общим правилам

```
\usepackage{mathtext}
```

в математической моде можно использовать русские буквы, набирая их прямо на клавиатуре. Детальное описание этой темы дано в разделе 5.2.1 «Кириллические математические алфавиты».

1.1 Формулы внутри текста

Формулы внутри текста надо размещать между командами `\(` и `\)` или в окружении `math` (между `\begin{math}` и `\end{math}`), или выделять с обеих сторон знаками доллара `$`²:

Катеты a и b треугольника связаны с гипотенузой c формулой $c^2 = a^2 + b^2$. Катеты a и b треугольника связаны с гипотенузой c формулой $(c^2 = a^2 + b^2)$.

Когда знаки препинания являются частью предложения, а не формулы, их надо ставить *вне* формул. В этом случае мы получим более подходящие пробелы и, кроме того, можем избежать проблем с вёрсткой абзаца. Пример:

Очевидно, что $a_i < b_i$ для $i = 1, 2, \dots, n$. Очевидно, что $(a_i < b_i)$ для $i=1, \dots, n$.

В этом примере после многоточия `\ldots` пришлось вставить дополнительный пробел командой `\,`. Иначе запятая стояла бы сразу после многоточия. Команды для коррекции пробелов «вручную» приведены в таблице 34.

Настройка По умолчанию промежутки между формулой и окружающим её текстом такие же, как между словами в тексте. Можно, однако, присвоить параметру `\mathsurround` значение ненулевой длины и тогда формулы будут окружены дополнительными пробелами справа и слева (этот пробел не печатается перед формулой, попавшей в начало строки, и после формулы, попавшей в конец строки). Например, если задать

```
\setlength{\mathsurround}{2pt}
```

то после этого каждая формула внутри текста будет окружена дополнительными пробелами по 2 пункта с обеих сторон.

При вёрстке абзаца Т_ЕX может разорвать формулу для переноса её части на другую строку. Переносы возможны после символов бинарных отношений типа знака равенства и символов бинарных операций типа знака сложения, причём последний знак в строке, вопреки российской традиции, не дублируется в начале следующей (один из вариантов решения этой проблемы описан на стр. 36). В остальных местах формулы не разрываются. Поэтому надо избегать длинных неразрываемых фрагментов в формулах, иначе могут возникнуть проблемы с вёрсткой абзаца.

Имеется специальная команда `*`, указывающая возможное место для переноса формулы с одной строки на другую по знаку умножения: $(x+y) * (z+t)$. Если формула поместится на строке, то

²Здесь и далее с правой стороны примера показан исходный текст.

получится обычное $(x + y)(z + t)$. Если же позиция знака умножения оказывается подходящей для переноса строки, получится $(x + y) \times$ в конце первой строки и $(z + t)$ в начале строки следующей.

В длинных формулах, не содержащих знаков «=», «+» и т.п., можно самому указать точку возможного переноса формулы командой `TeXa \allowbreak`:

Вектор состояния имеет вид $(x_1, \dots, x_m, y_1, \dots, y_n)$. Вектор состояния имеет вид `$(x_1, \dots, x_m, \allowbreak y_1, \dots, y_n)$`.

Заметим, что в этом примере запятые являются частью формулы.

Если формула с командой `\allowbreak` окажется внутри строки, то разрыва, естественно, не будет:

$x_1, \dots, x_m, y_1, \dots, y_n$ `$(x_1, \dots, x_m, \allowbreak y_1, \dots, y_n)$`

Если какую-нибудь часть формулы заключить в фигурные скобки, то её `TeX` *никогда* не разорвёт (даже после знака «=»). Такой способ борьбы с разрывом формул является грубым, поскольку промежутки, которые `TeX` вставляет между символами, становятся нерастяжимыми, что затрудняет вёрстку абзаца.

На заметку Штраф за перенос формулы после символов бинарной операции и бинарного отношения задают параметры `TeXa \binoppenalty` и `\relpenalty`, соответственно. По умолчанию значение `\binoppenalty` равно 700, а `\relpenalty`—500. Можно в преамбуле присвоить им бóльшие значения, например, `\relpenalty=900` (допустимы только целые числа), тогда вероятность разрыва формулы уменьшится: `TeX` постарается сделать переносы в тексте, а не в формулах. 10000 означает абсолютный запрет переноса. При этом пробелы между символами остаются растяжимыми длинами.

1.1.1 Как избежать длинных строк в тексте

Обсудим ситуацию, когда формула (или другой недопускающий переносов текст), оказавшись на правом краю строки, выходит на поля. У `TeXa` был выбор: либо перенести формулу на другую строку, сделав текущую строку разреженной (с большими промежутками между словами), либо не переносить формулу и создать более длинную строку. `TeX` выбрал второй вариант. Что можно сделать, чтобы избавиться от длинных строк. Во-первых, можно переписать абзац так, чтобы формула оказалась внутри строки. Если этого сделать нельзя, то можно ослабить критерий, по которому `TeX` предпочитает длинные строки, а не разреженные.

В обычном режиме вёрстки разреженность строки, которую `TeX` никогда не превышает, задаётся значением параметра `\tolerance`. По умолчанию оно равно 200. Можно увеличить это значение, тогда `TeX` сможет делать все строки более разреженными. Если длинная строка находится в середине или в конце большого абзаца, то вставляя небольшие дополнительные промежутки между словами в предшествующих строках и делая новые переносы, `TeX` может накопить текст для заполнения этой строки и, следовательно, перенести формулу на следующую строку. Увеличить значение параметра `\tolerance` можно для всего документа сразу, поместив в преамбуле строку наподобие

`\tolerance=500`

или внутри группы, содержащей абзац с длинной строкой. В последнем случае закрывающая фигурная скобка должна стоять после пустой строки за тем абзацем, на который вы хотите распространить действие декларации.

Более грубый способ избавления от длинных строк даёт декларация `\sloppy`. Она устанавливает максимальное значение параметра `\tolerance` 10000 и разрешает `TeXu` делать практически сколь угодно разреженные строки. Отменяет её действие декларация `\fussy`. Область действия декларации `\sloppy` можно ограничить группой. В этом случае закрывающая фигурная скобка должна стоять после пустой строки за тем абзацем, на который вы хотите распространить действие декларации. Вместо декларации можно использовать командные скобки `sloppypar`. Декларация `\sloppy` имеет один существенный недостаток. Она, начиная с некоторой величины дополнительных промежутков между словами в разреженной строке, позволяет `TeXu` не увеличивать штраф за дальнейший рост разреженности. В результате, `TeX` может сконцентрировать всю разреженность в одной строке.

1.2 Выделенные формулы

Выделенные в отдельную строку формулы \TeX печатает с временным прерыванием текущего абзаца. Это означает, что если после формулы нет пустой строки или команды `\par`, то следующая за формулой строка печатается без абзацного отступа.

1.2.1 Однострочные уравнения

Формулы, которые должны печататься без порядкового номера на отдельной строке, располагают между командами `\[` и `\]` или в окружении `displaymath`:

Катеты a и b треугольника связаны с гипотенузой c формулой

Катеты a и b треугольника связаны с гипотенузой c формулой `\[c^2=a^2+b^2. \]`

$$c^2 = a^2 + b^2.$$

Знаки пунктуации после выделенной формулы ставятся в конце самой формулы! Иначе этот знак появится в начале новой строки.

Если математическое выражение в строке состоит не из одной формулы, а из нескольких, то пробелы между ними надо расставлять вручную. Обычно используется команда `\qquad`:

$$F_n = F_{n-1} + F_{n-2}, \quad n \geq 2. \quad \backslash[F_n=F_{n-1}+F_{n-2}, \qquad n \geq 2. \backslash]$$

В математических текстах формулы обычно нумеруются для того, чтобы на них можно было сослаться по ходу документа. \LaTeX позволяет организовать нумерацию таким образом, чтобы номера формул и ссылки на них создавались автоматически. Чтобы \LaTeX пронумеровал формулу, её надо поместить в окружение `equation`. Каждая такая формула на печати автоматически получает свой порядковый номер:

Катеты a и b треугольника связаны с гипотенузой c формулой

Катеты a и b треугольника связаны с гипотенузой c формулой
`\begin{equation}`
`c^2 = a^2 + b^2.`
`\end{equation}`

$$c^2 = a^2 + b^2. \quad (1)$$

Чтобы на присвоенный формуле номер можно было сослаться в тексте, надо формулу пометить: в любом месте между `\begin{equation}` и `\end{equation}` поставить команду `\label`, аргументом которой будет «имя» формулы, и после этого команда `\ref` будет печатать номер этой формулы. Поясним все сказанное примером:

$$e^x = 1 + x \quad (2) \quad \backslashbegin{equation}\label{eq:exp} \quad e^x=1+x \quad \backslashend{equation}$$

Уравнение (2) справедливо только при очень малых x .

Уравнение~(`\ref{eq:exp}`) справедливо только при очень малых x .

Отметим, что скобки вокруг номера формулы, созданного командой `\ref`, автоматически *не* ставятся. В пакете `amsmath` из коллекции $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX есть команда `\eqref`, которая сама ставит скобки. Так, `\eqref{eq:exp}` печатает (2).

В стандартных классах документа номер формулы по умолчанию ставится справа от формулы. Однако, при наличии опции `leqno` в команде `\documentclass`

```
\documentclass[leqno]{article}
```

номера формул будут размещаться слева от формулы. Пакет `amsmath` вводит опцию `reqno`: если какой-нибудь нестандартный класс устанавливает по умолчанию нумерацию с левой стороны уравнений, то опция `reqno` отменяет это правило.

То, как именно выглядит на печати номер формулы, зависит от класса документа: например, в классе `article` формулы имеют сплошную нумерацию, а в классе `book` нумерация формул начинается заново в каждой главе и номер, скажем, 5-ой по счёту пронумерованной формулы из главы 3, имеет вид (3.5).

На заметку Номера уравнений хранятся в счётчике `equation`. С помощью команд, предназначенных для работы со счётчиками, можно самостоятельно менять формат номера формул и последовательность нумерации. Один из вариантов изменения нумерации предлагает пакет `amsmath`. Если в преамбулу документа вписать

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

то в каждой секции уравнения будут нумероваться независимо друг от друга, и номер формулы, скажем в классе `article`, будет иметь вид, например, (2.7). Вместо `section` можно указать другие имена счётчиков команд секционирования. Для `subsection` номер формулы будет, очевидно, иметь вид, например, (2.1.4).

Пакет `amsmath` наряду с окружением `equation` вводит окружение `equation*`, в котором формула не получает номера. Добавляя или убирая звёздочку, можно быстро включать или выключать из нумерации то или иное уравнение.

В стандартных классах документа по умолчанию формулы располагаются по центру строки. Если в команде `\documentclass` указать опцию `fleqn`

```
\documentclass[fleqn]{article}
```

то формулы будут выравниваться по левому краю страницы.

Настройка При включённой опции `fleqn` расстояние от левого поля страницы до формулы задаётся параметром `\mathindent`. По умолчанию оно равно `2.5em`. Меняя значение параметра `\mathindent` командами `\setlength` или `\addtolength`, можно устанавливать любую величину отступа формулы от края страницы.

Для информации \TeX вставляет дополнительные вертикальные промежутки перед выделенной формулой и после неё. Если конец строки текста перед формулой находится на `2em` ближе к левому полю страницы, чем левый край формулы («короткая» формула), то величина промежутков задаётся параметрами `\abovedisplayshortskip` и `\belowdisplayshortskip`. Это растяжимые длины со значениями `0pt` и `7pt`, соответственно. В случае «длинной» формулы величина промежутков задаётся параметрами `\abovedisplayskip` и `\belowdisplayskip`. Это растяжимые длины со значениями `12pt`. Если включена опция `fleqn`, то дополнительные вертикальные промежутки задаются параметром `\topsep`.

1.2.2 Системы уравнений

Для набора систем уравнений в \LaTeX предусмотрены окружения `eqnarray` для нумерованных уравнений и `eqnarray*` для ненумерованных формул. Внутри окружения уравнения, которые должны размещаться на отдельных строках, отделяются друг от друга командами `\\`. Выражение в пределах одной строки должно состоять из *трёх* частей (возможно пустых), разделённых амперсантами `&`. Каждая часть помещается в свой столбец. В левом столбце формулы прижимаются к правому краю, в среднем — центрируются, а в правом столбце — прижимаются к левому краю. Другими словами, символы `&` задают точки выравнивания уравнений из разных строк по вертикали. Каждая строка в окружении `eqnarray` получает при печати свой номер. Подавить нумерацию любой строки можно командой `\nonumber`. Пример:

$I = U + PV$	(3)	<code>\begin{eqnarray}</code>
$\Psi = U - TS$	(4)	<code> I & = & U + PV \\</code>
$\Psi + PV = \Phi$		<code> \Psi & = & U - TS \\</code>
		<code> \Psi + PV & = & \Phi \nonumber</code>
		<code>\end{eqnarray}</code>

Настройка Величину пробела между строками в окружениях `eqnarray` и `eqnarray*` можно поменять, изменяя значение нерастяжимой длины `\jot` (по умолчанию равна `3pt`).

Окружения `equation` и `eqnarray` используют один и тот же счётчик `equation`, значение которого и печатается как номер формулы. Следовательно, все нумерованные уравнения в документе будут иметь единую нумерацию.

Для печати системы уравнений без выравнивания по вертикали пакет `amsmath` вводит окружение `gather`. Оно даёт нумерованные уравнения. Для ненумерованных формул надо использовать окру-

жение `gather*`. Точки переноса строк задаются командой `\\`. Все строки центрируются. Пример:

$$\begin{gather}
 A_x = -Hy \quad (5) \\
 A_y = A_z = 0 \quad (6)
 \end{gather}$$

Для набора систем уравнений с выравниванием по вертикали, пакет `amsmath` предлагает окружения `align`, `alignat` и `flalign` для нумерованных уравнений и `align*`, `alignat*` и `flalign*` для ненумерованных уравнений. Точки переноса строк задаются командой `\\`. Точки выравнивания уравнений из разных строк по вертикали задаются амперсантом `&`. В отличие от окружения `eqnarray`, уравнения не надо разбивать на три части:

$$\begin{align}
 \Psi = U - TS \quad (7) \\
 \Psi + PV = \Phi \quad (8)
 \end{align}$$

В окружениях `AMS-LATEX` вставляются правильные пробелы в точках выравнивания. Кроме того, в каждой строке можно размещать по несколько уравнений, формируя из них столбцы. Первый, третий и т.д. знаки `&` в строке задают точки выравнивания по вертикали внутри столбцов. Второй, четвёртый и т.д. знаки `&` служат как разделители столбцов.

В окружениях `align` и `align*` между столбцами уравнений, а также перед первым столбцом и после последнего столбца, автоматически вставляются равные пробелы. При расчёте величины промежутка наличие номера у строки не учитывается. Пример:

$$\begin{align}
 x = y \quad a = b + c \quad (9) \\
 x' = y' \quad a' = b \quad (10)
 \end{align}$$

В окружениях `alignat` и `alignat*` автоматически вставляются равные пробелы только перед первым столбцом и после последнего столбца. Промежутки между столбцами уравнений автоматически не вставляются. Их надо задавать самому командами, которые вставляют горизонтальные промежутки в математической моде. Окружения `alignat` и `alignat*` имеют обязательный аргумент, значение которого указывает количество столбцов. Число символов `&` в каждой строке не должно превышать необходимого для создания этого количества столбцов. Пример:

$$\begin{alignat}{2}
 x = y \quad a = b + c \quad (11) \\
 x' = y' \quad a' = b \quad (12)
 \end{alignat}$$

В окружениях `flalign` и `flalign*` автоматически вставляются равные промежутки только между столбцами уравнений. Поэтому первый и последний столбцы прижаты к полям страницы:

$$\begin{flalign}
 x = y \quad a = b + c \quad z > 0 \\
 x' = y' \quad a' = b \quad z' = 0
 \end{flalign}$$

В окружении `flalign` крайний столбец прижимается к номеру строки.

В окружениях `AMS-LATEX`, предназначенных для печати систем нумерованных уравнений, можно подавить нумерацию любой строки, используя команду `\notag`, аналогичную команде `LATEX` `\nonumber`.

1.2.3 Расщепление длинных формул

Формулы, выделенные в отдельную строку, в отличие от внутритекстовых, `TEX` *никогда* не переносит. Если такая формула не помещается в строке и выходит на поля страницы, то при трансляции `TEX` выдаёт сообщение о переполнении (`Overfull ...`). Разбивать длинные формулы на строки

приходится вручную. Для печати таких формул можно использовать окружения `eqnarray` или `eqnarray*`:

$$\int (F_i x_k - F_k x_i) dV = \oint (u_{il} x_k - u_{kl} x_i) df_l$$

```
\begin{eqnarray*}
& \int (F_i x_k - F_k x_i) \, dV = & & \\
& = \oint (u_{il} x_k - u_{kl} x_i) \, df_l & & \\
\end{eqnarray*}
```

Чтобы части формулы на разных строках не начинались бы точно одна под другой, приходится использовать команды, вставляющие пробелы, например, `\qquad`.

В случае окружения `eqnarray` формула получит номер. Разумеется, перед командой `\`, завершающей оборванную строку, надо поставить команду `\nonumber`, иначе обрубок формулы будет пронумерован. Для смещения вправо части формулы, расположенной на второй строке, можно воспользоваться точками выравнивания, задаваемыми символами `&`:

$$x + y = a + b + c + d + e + f + g. \quad (13)$$

```
\begin{eqnarray}
x+y & = & a + b + c + & \{\} \backslash\number \\
& & d + e + f + g. & \\
\end{eqnarray}
```

Парные скобки `{}` в исходном тексте в конце первой строки задают «пустую формулу». Поскольку знак «+» стоит между двумя формулами, \TeX делает пробелы надлежащего размера.

Пакет `amsmath` вводит специальное окружение `multline` для расщепления уравнений на несколько частей. Оно даёт нумерованную формулу. Для нумерованных формул надо использовать окружение `multline*`. Точки переноса строк задаются командой `\`. Первая часть уравнения прижимается к левому полю, а последняя — к правому. Остальные части центрируются (исключая случай, когда в `\documentclass` задана опция `fleqn`). Пример:

$$A = a_{11}\xi_1^2 + 2a_{12}\xi_1\xi_2 + \dots + 2a_{1n}\xi_1\xi_n + \sum_{i,j=2}^n a_{ij}\xi_i\xi_j \quad (14)$$

```
\begin{multline}
A = a_{11}\xi_1^2 + & \{\} \\
& \{\} + 2a_{12}\xi_1\xi_2 + \dots + \\
& 2a_{1n}\xi_1\xi_n + & \{\} \\
& \{\} + \sum_{i,j=2}^n a_{ij}\xi_i\xi_j & \\
\end{multline}
```

Расстояние первой и последней строки до полей можно задавать, меняя значение параметра `\multlinegap`. Любую из средних строк можно сместить влево или вправо, сделав её аргументом команд `\shoveleft` и `\shoveright`, соответственно (`\` остаётся вне).

Для расщепления длинных выражений с выравниванием по вертикали, пакет `amsmath` предлагает окружение `split`. Точки переноса строк задаются командой `\`. Точки выравнивания частей формулы из разных строк задаются амперсантом `&`. Окружение `split` используется только внутри других окружений для набора выделенных формул, поскольку оно само не переключает \TeX в математическую моду. Пример:

$$\Psi = \cos kz + i \sin kz + \frac{f(\theta)}{r} (\cos kr + i \sin kr) \quad (15)$$

```
\begin{equation}\begin{split}
\Psi = & \cos kz + i \sin kz + & \{\} \\
& & \{\} + \frac{f(\theta)}{r} \\
& & (\cos kr + i \sin kr) & \\
\end{split}\end{equation}
```

По умолчанию действует опция `centertags` и номер расщеплённого уравнения центрируется по вертикали относительно общей высоты уравнения. Если пакет `amsmath` загрузить с опцией `tbtags` («Top-or-bottom tags»), то номер уравнения ставится на один уровень с последней строкой при нумерации с правой стороны или с первой строкой при нумерации слева.

1.2.4 Блоки уравнений

В окружениях `aligned`, `gathered` и `alignedat` из пакета `amsmath` формулы форматируются по тем же правилам, что и в окружениях `align`, `gather` и `alignat`, соответственно. Но, в отличие от последних, они занимают по горизонтали не всю строку, а ровно столько, сколько необходимо

для уравнений. Это позволяет размещать на одной строке несколько блоков уравнений, в каждом из которых формулы форматируются независимо друг от друга. Выравнивание блоков по вертикали относительно осевой линии строки задаётся необязательным аргументом с допустимыми значениями `t`, `c` (действует по умолчанию) или `b`. Окружения `aligned`, `gathered` и `alignedat` используются только внутри других окружений для набора выделенных формул. Пример:

$$\left. \begin{aligned} B' &= -\partial \times E \\ E' &= \partial \times B - 4\pi j \end{aligned} \right\} \text{Maxwell's eqs}$$

```

\begin{equation*}
\left. \begin{aligned}
B' &= -\partial \times E \\
E' &= \partial \times B - 4\pi j
\end{aligned} \right\} \text{Maxwell's eqs}
\end{equation*}

```

1.2.5 Вертикальные пробелы в многострочных формулах

Во всех командных скобках пакета `amsmath` расстояние между строками можно изменить, используя команду `\l` с необязательным аргументом, значением которого является величина дополнительного вертикального пробела. Так, команда `\l[6pt]` добавляет после строки, которую она завершает, пробел в 6 pt.

1.2.6 Смещение номера уравнения

Окружения пакета `amsmath` не допускают печать номера уравнения на самом уравнении, когда оно занимает всю строку. В этом случае номер уравнения размещается на отдельной строке над или под этим уравнением:

$$y = a + b + c + d + e + f + g + h + i + j \tag{16}$$

```

\begin{align}
y &= a+b+c+d+e+f+g+h+i+j \\
\end{align}

```

При неудачном расположении номера какого-нибудь уравнения, этот номер можно сдвинуть вверх или вниз, разместив в строке с этим уравнением команду `\raisetag`. Так `\raisetag{6pt}` поднимает номер на шесть пунктов.

1.2.7 Разрыв многострочных формул

В отличие от `eqnarray`, командные скобки из пакета `amsmath` не допускают переноса части многострочной формулы или части системы уравнений на следующую страницу. Если текст содержит большие системы уравнений, то это может привести к частично незаполненным страницам (или к растянутым по вертикали страницам, если действует декларация `\flushbottom`).

Команда `\allowdisplaybreaks` отменяет этот запрет сразу для всех окружений. Её место в преамбуле документа. Команда имеет необязательный аргумент с допустимыми значениями от 1 до 4: чем больше значение, тем слабее запрет. [1] рекомендует при возможности избегать разрыва; [4] — полностью снимает запрет (как и без аргумента).

Когда с помощью команды `\allowdisplaybreaks` позволено разрывать многострочные уравнения, можно, как обычно, запретить разрыв после какой-нибудь строки, используя вместо `\l` команду `\l*`.

Команда `\displaybreak` позволяет разорвать формулу там, где надо. Её место непосредственно перед `\l`, которые завершают строку на месте разрыва. Команда имеет необязательный аргумент с допустимыми значениями от 0 до 4: чем больше значение, тем вероятнее разрыв. `\displaybreak[0]` допускает разрыв, но не заставляет его делать. `\displaybreak[4]` принуждает к разрыву (как и без аргумента).

Команды разрыва многострочных формул не действуют в окружениях `split`, `aligned`, `gathered` и `alignedat`, которые форматируют блоки уравнений.

1.3 О промежутках между символами

TeX игнорирует все пробелы, вставленные между символами в исходном тексте, и расставляет промежутки сам. Размер промежутка зависит от того, к какому типу относятся символы. Пример:

$$y = 1.2 + x \quad y=(+x) + 1,2 \quad \backslash (y=1.2+x \quad \backslash \text{quad} \quad y\{=\} (+x)+1,2 \quad \backslash)$$

В первой формуле знаки «=» и «+» рассматриваются как символы сравнения и арифметической операции, соответственно. Их TeX окружает надлежащими пробелами. Во второй формуле этих пробелов нет, поскольку знак «=» стоит в фигурных скобках и рассматривается не как символ бинарного отношения, а как подформула; знак «+» перед буквой x не считается символом бинарной операции; запятая, в отличие от точки, считается символом пунктуации, после которых ставится небольшой пробел.

TeX различает восемь типов символов. Имеются специальные команды, которые объявляют свой аргумент — любую формулу — «символом» соответствующего типа. `\mathord` задаёт обычный символ ($A \ O \ \Phi \ \infty$), `\mathopen` и `\mathclose` — открывающий (`\langle` `|`) и закрывающий разделители (`|` `\rangle`), `\mathpunct` — знак пунктуации (`,` `;` `!`). `\mathbin`, `\mathrel` и `\mathop` определяют бинарные операции ($+ \times \pm$), бинарные отношения ($= \gg \subset$) и математические операции ($\sum \int \log$), соответственно. Последняя команда `\mathalpha` предназначена для объявления символов алфавитными, чтобы на них действовали команды, приведённые в разделе «Математические алфавиты».

Рассмотрим на примере, как с помощью команды `\mathop` можно «расставить» правильные пробелы. В формуле

$$\text{Sp}\hat{\sigma}_i = 0 \quad \backslash (\text{\mathrm{Sp}} \hat{\sigma}_i = 0 \quad \backslash)$$

явно не хватает небольшого пробела между `Sp` и $\hat{\sigma}_i$. Конечно, его можно вставить вручную командами из таблицы 34. Но лучше доверить эту работу TeXу: он уж точно сделает правильные пробелы. Поэтому объявим обозначение шпура матрицы математической операцией (что соответствует действительности):

$$\text{Sp}\hat{\sigma}_i = 0 \quad \backslash (\text{\mathop{\mathrm{Sp}}}\hat{\sigma}_i=0 \quad \backslash)$$

Видно, что теперь TeX сделал необходимый пробел. $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\text{TeX}$ автоматизировал определение новых математических операций. Процедура описана на стр. 28.

1.4 О размерах символов

В математической моде используются символы трех размеров: текстового, индексов и индексов к индексам. Если размер шрифта документа равен 10 pt, то размер индексов будет 7 pt, а индексов к индексам любого уровня — 5 pt. Размер символов, которыми печатается та или иная часть формулы, задаётся её стилем. TeX сам выбирает, какой стиль использовать в числителе и знаменателе дробей, в индексах и т.д. Можно, однако, изменить стиль формулы следующими декларациями:

`\displaystyle` задаёт стиль для формул на отдельной строке. В этом стиле используется текстовый размер символов.

`\textstyle` задаёт стиль для формул в тексте и в окружение `array`. В этом стиле также используется текстовый размер символов.

`\scriptstyle` задаёт стиль для индексов первого уровня. В этом стиле используется размер индексов.

`\scriptscriptstyle` задаёт стиль для индексов высших уровней. В этом стиле используется размер индексов к индексам.

От стиля зависит не только размер символов, но и положение показателя степени. Сравните

$$x^2 \quad x^2 \quad \$ x^2 \quad \backslash \text{quad} \quad \backslash \text{displaystyle} \quad x^2 \quad \$$$

Если не учитывать эту деталь, то правила переключения стиля можно сформулировать так.

Выделенная в отдельную строку формула начинает набираться в стиле `displaystyle`, внутритекстовая — в стиле `textstyle`; далее, если встретится команда типа `\frac`, то для набора числителя и знаменателя \TeX переключится на следующий по порядку стиль. Так, во внутритекстовой формуле числитель и знаменатель дроби набирается в стиле индексов:

$$p + \frac{a}{b} \quad (\backslash p + \backslash frac\{a\}\{b\} \backslash)$$

$$p + \frac{a}{b} \quad (\backslash [p + \backslash frac\{a\}\{b\} \backslash)$$

Если в момент действия двух первых стилей встретится индекс, то он набирается стилем для индексов; если индекс встретится в момент действия стиля для индексов или индексов к индексам, то набираться он будет в стиле `scriptscriptstyle`.

$$p + \frac{a^{x+y^2}}{b^2} \quad (\backslash p + \backslash frac\{a^{\{x+y^2\}}\}\{b^2\} \backslash)$$

$$p + \frac{a^{x+y^2}}{b^2} \quad (\backslash [p + \backslash frac\{a^{\{x+y^2\}}\}\{b^2\} \backslash)$$

Если вы хотите изменить стиль формулы, укажите его в явном виде, используя перечисленные в этом разделе декларации. Типичный пример, когда это необходимо, приведён на стр. 18.

2 Математические символы

2.1 Показатели степени, индексы и штрихи

Верхние и нижние индексы набираются с помощью знаков `^` и `_`, соответственно. Если индекс или показатель степени — выражение, состоящее более чем из одного символа, то его надо взять в фигурные скобки. Если у одной буквы есть как верхние, так и нижние индексы, то указывать их можно в произвольном порядке. Если же требуется, чтобы верхние и нижние индексы располагались не один под другим, а на разных расстояниях от выражения, к которому они относятся, то нужно оформить часть индексов как индексы к пустой подформуле. Пример:

Тензоры R_{jkl}^i и R_j^{ik} . Тензоры `\$R^i_{jkl}\$` и `\$R_j^{\{ik\}}_l\$`.

Формулы типа $\text{Fe}_2^{+2}\text{Cr}_2\text{O}_4$ смотрятся плохо, поскольку нижние индексы находятся на разной высоте. Один из вариантов решения этой проблемы приведён в разделе 6.4.

Штрихи в формулах набираются прямым апострофом `'` или командой `\prime`:

$$(fg)'' = f''g + 2f'g' + fg''. \quad (\backslash (fg)'' = f''g + 2f'g' + fg''. \backslash)$$

2.2 Многоточия

Многоточия разного вида набираются командами из таблицы 1.

Таблица 1: Многоточия

$$\dots \quad \cdots \quad \ddots \quad \dots \quad \ldots \quad \vdots \quad \vdots$$

Согласно американской традиции, пропущенные слагаемые или множители заменяются многоточием `\cdots`. В России обычно многоточие ставят внизу строки и в этом случае. Пакет `amsmath` вводит команду `\dots`, которая сама выбирает положение многоточия в зависимости от того, что за ней следует: после знаков бинарных операций многоточие ставится на осевой линии формулы. Пример:

$$X_1 + X_2 + \dots + X_n \quad (\backslash (X_1 + X_2 + \dots + X_n \backslash)$$

Поскольку нет общей договорённости о расположении многоточий, пакет `amsmath` ввёл ряд команд для набора многоточий в конце формул в зависимости от контекста:

`\dotsc` надо ставить после запятой (A_1, A_2, \dots);

`\dotspb` — после символов бинарных операций и отношений ($A_1 + A_2 + \dots$);

`\dotasm` — после сомножителей ($A_1 A_2 \dots$);

и, наконец,

`\dotssi` — после знака интеграла ($\int_{A_1} \int_{A_2} \dots$).

Команда `\dotso` рекомендуется для остальных случаев. Теперь, в зависимости от требований издательства, можно переопределением команд легко заменить одно многоточие на другое.

2.3 Символы бинарных операций

Знаки сложения, умножения и т.п. являются символами бинарных операций. \TeX окружает такие символы «средними пробелами» (теми, которые задаются вручную командой `\:` из таблицы 34). Если есть основания считать, что эти знаки используются не для обозначения операций, а для других целей, то пробела не будет. Полный список символов бинарных операций вместе с командами для их набора приведён в таблице 2.

Таблица 2: Символы бинарных операций

$+$	<code>\+</code>	$-$	<code>\-</code>	\oplus	<code>\oplus</code>	\times	<code>\times</code>
\amalg	<code>\amalg</code>	\cup	<code>\cup</code>	\otimes	<code>\otimes</code>	\triangleleft	<code>\triangleleft</code>
$*$	<code>\ast</code>	\dagger	<code>\dagger</code>	\oslash	<code>\oslash</code>	\triangleleft	<code>\triangleleft</code>
\bigcirc	<code>\bigcirc</code>	\ddagger	<code>\ddagger</code>	\otimes	<code>\otimes</code>	\triangleright	<code>\triangleright</code>
∇	<code>\bigtriangledown</code>	\diamond	<code>\diamond</code>	\pm	<code>\pm</code>	\leq	<code>\unlhd^a</code>
Δ	<code>\bigtriangleup</code>	\div	<code>\div</code>	\triangleright	<code>\rhd^a</code>	\geq	<code>\unrhd^a</code>
\bullet	<code>\bullet</code>	\triangleleft	<code>\lhd^a</code>	\setminus	<code>\setminus</code>	\uplus	<code>\uplus</code>
\cap	<code>\cap</code>	\mp	<code>\mp</code>	\sqcap	<code>\sqcap</code>	\vee	<code>\vee</code>
\cdot	<code>\cdot</code>	\odot	<code>\odot</code>	\sqcup	<code>\sqcup</code>	\wedge	<code>\wedge</code>
\circ	<code>\circ</code>	\ominus	<code>\ominus</code>	\star	<code>\star</code>	\wr	<code>\wr</code>

^aТребуется подключение пакета `latexsym`.

Пакет `amssymb` из коллекции $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ вводит ряд экзотических символов, которые приведены в таблице 3.

Таблица 3: Символы бинарных операций AMS (пакет `amssymb`)

$\bar{\wedge}$	<code>\barwedge</code>	\circledcirc	<code>\circledcirc</code>	\intercal	<code>\intercal</code>
\boxdot	<code>\boxdot</code>	\circledast	<code>\circledast</code>	\leftthreetimes	<code>\leftthreetimes</code>
\boxminus	<code>\boxminus</code>	\Cup	<code>\Cup, \doublecup</code>	\ltimes	<code>\ltimes</code>
\boxplus	<code>\boxplus</code>	\curlyvee	<code>\curlyvee</code>	\rightthreetimes	<code>\rightthreetimes</code>
\boxtimes	<code>\boxtimes</code>	\curlywedge	<code>\curlywedge</code>	\rtimes	<code>\rtimes</code>
\Cap	<code>\Cap, \doublecap</code>	\divideontimes	<code>\divideontimes</code>	\smallsetminus	<code>\smallsetminus</code>
\centerdot	<code>\centerdot</code>	\dotplus	<code>\dotplus</code>	\veebar	<code>\veebar</code>
\circledast	<code>\circledast</code>	$\overline{\wedge}$	<code>\doublebarwedge</code>	$\&$	<code>\And^a</code>

^aТребуется подключение пакета `amsmath`.

2.4 Символы бинарных отношений

Знаки $=$, $<$, $>$ и т.п. являются символами бинарных отношений. \TeX окружает такие символы «толстыми пробелами» (теми, которые задаются вручную командой `\;` из таблицы 34). Полный список символов бинарных отношений вместе с командами для их набора приведён в таблице 4.

Таблица 4: Символы бинарных отношений

\lt	<code><</code>	\gt	<code>></code>	$=$	<code>=</code>	$:$	<code>:</code>
\approx	<code>\approx</code>	\in	<code>\in</code>	\prec	<code>\prec</code>	\subset	<code>\subset</code>
\asymp	<code>\asymp</code>	\Join	<code>\Join^a</code>	\preceq	<code>\preceq</code>	\subseteq	<code>\subseteq</code>
\bowtie	<code>\bowtie</code>	\leq	<code>\leq, \le</code>	\propto	<code>\propto</code>	\succ	<code>\succ</code>
\cong	<code>\cong</code>	\ll	<code>\ll</code>	\sim	<code>\sim</code>	\succeq	<code>\succeq</code>
\dashv	<code>\dashv</code>	\mid	<code>\mid</code>	\simeq	<code>\simeq</code>	\supset	<code>\supset</code>
\doteq	<code>\doteq</code>	\models	<code>\models</code>	\smile	<code>\smile</code>	\supseteq	<code>\supseteq</code>
\equiv	<code>\equiv</code>	\neq	<code>\neq, \ne</code>	\sqsubset	<code>\sqsubset^a</code>	\vdash	<code>\vdash</code>
\frown	<code>\frown</code>	\ni	<code>\ni, \owns</code>	\sqsubseteq	<code>\sqsubseteq</code>	\notin	<code>\notin</code>
\geq	<code>\geq, \ge</code>	\parallel	<code>\parallel</code>	\sqsupset	<code>\sqsupset^a</code>		
\gg	<code>\gg</code>	\perp	<code>\perp</code>	\sqsupseteq	<code>\sqsupseteq</code>		

^aТребуется подключение пакета `latexsym`.

Знаки \leq и \geq отсутствуют в `LATEX`; но они доступны при использовании пакета `amssymb`. Все символы бинарных отношений `AMS-LATEX` приведены в таблице 5.

Таблица 5: Символы бинарных отношений AMS (пакет `amssymb`)

\approx	<code>\approxeq</code>	\gt	<code>\gtrdot</code>	\smile	<code>\smallsmile</code>
\backsimeq	<code>\backepsilon</code>	\gtrless	<code>\gtrless</code>	\sqsubset	<code>\sqsubset</code>
\backsim	<code>\backsim</code>	\gtrqless	<code>\gtrqless</code>	\sqsupset	<code>\sqsupset</code>
\backsimeq	<code>\backsimeq</code>	\gtrless	<code>\gtrless</code>	\Subset	<code>\Subset</code>
\because	<code>\because</code>	\gtrsim	<code>\gtrsim</code>	\subseteq	<code>\subseteq</code>
\between	<code>\between</code>	\leqq	<code>\leqq</code>	\succapprox	<code>\succapprox</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\leqslant	<code>\leqslant</code>	\succcurlyeq	<code>\succcurlyeq</code>
\blacktriangleright	<code>\blacktriangleright</code>	\lessapprox	<code>\lessapprox</code>	\succsim	<code>\succsim</code>
\bumpeq	<code>\bumpeq</code>	\lessdot	<code>\lessdot</code>	\Supset	<code>\Supset</code>
\bumpeq	<code>\bumpeq</code>	\lesseqgtr	<code>\lesseqgtr</code>	\supseteq	<code>\supseteq</code>
\circeq	<code>\circeq</code>	\lesseqgtr	<code>\lesseqgtr</code>	\therefore	<code>\therefore</code>
\curlyeqprec	<code>\curlyeqprec</code>	\lessgtr	<code>\lessgtr</code>	\thickapprox	<code>\thickapprox</code>
\curlyeqsucc	<code>\curlyeqsucc</code>	\lesssim	<code>\lesssim</code>	\thicksim	<code>\thicksim</code>
\doteqdot	<code>\doteqdot, \Doteq</code>	\lll	<code>\lll, \lllless</code>	\trianglelefteq	<code>\trianglelefteq</code>
\eqcirc	<code>\eqcirc</code>	\pitchfork	<code>\pitchfork</code>	\triangleq	<code>\triangleq</code>
\eqslantgtr	<code>\eqslantgtr</code>	\precapprox	<code>\precapprox</code>	\trianglerighteq	<code>\trianglerighteq</code>
\eqslantless	<code>\eqslantless</code>	\preccurlyeq	<code>\preccurlyeq</code>	\varpropto	<code>\varpropto</code>
\fallingdotseq	<code>\fallingdotseq</code>	\precsim	<code>\precsim</code>	\vartriangleleft	<code>\vartriangleleft</code>
\geqq	<code>\geqq</code>	\risingdotseq	<code>\risingdotseq</code>	\vartriangleright	<code>\vartriangleright</code>
\geqslant	<code>\geqslant</code>	\shortmid	<code>\shortmid</code>	\Vdash	<code>\Vdash</code>
\ggg	<code>\ggg, \gggtr</code>	\shortparallel	<code>\shortparallel</code>	\vDash	<code>\vDash</code>
\gtrapprox	<code>\gtrapprox</code>	\smallfrown	<code>\smallfrown</code>	\Vvdash	<code>\Vvdash</code>

Чтобы получить отрицание отношения или сравнения (изображение перечёркнутого символа), надо перед командой, генерирующей этот символ, поставить команду `\not`. Пример:

$$A \not\subset B \qquad \backslash(A\not\subset B \backslash)$$

Если положение черты вас не устраивает, то его можно поправить командами, задающими про-

белы. Многие символы отрицательных бинарных отношений определены в пакете `amssymb`. Они приведены в таблице 6.

Таблица 6: Отрицательные бинарные отношения AMS (пакет `amssymb`)

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\lnleq	\lngeq	\nsubseteqeq
\lnleqslant	\lngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\lnleqq	\lngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\precneqq	\succneqq	\nvDash
\precnsim	\succnsim	\nVDash
\precnapprox	\succnapprox	\nVdash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subsetneqq	\supsetneqq	\ntrianglerighteq

2.5 Греческие буквы

Греческие буквы набираются с помощью команд. Имя команды, печатающей строчную греческую букву, совпадает с английским названием этой буквы (например, буква α задаётся командой `\alpha`). Исключение составляет буква o (омикрон): по начертанию она совпадает с курсивной латинской o , так что специальной команды для нее не предусмотрено и для ее набора достаточно просто написать o в формуле. Некоторые греческие буквы имеют по два варианта начертаний. Полный список команд для набора строчных греческих букв приведён в таблице 7. Греческая буква «каппа» в виде κ печатается командой `\varkappa` из пакета `amssymb` (см. таблицу 9).

Таблица 7: Строчные греческие буквы

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	υ	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

Имя команды, задающей прописную греческую букву, пишется с прописной буквы (например, буква Ψ задаётся командой `\Psi`). Ряд прописных греческих букв («альфа», например) совпадают по начертанию с латинскими и для них специальных команд нет. Надо просто набрать соответствующую

Заметим, что двоеточие как знак пунктуации набирается командой `\colon`. Символ « : », который набирается на клавиатуре, в математической моде является символом бинарного отношения. Сравните пробелы в обоих случаях:

$$f: X \rightarrow Y \quad f : X \rightarrow Y \quad \backslash(f\colon X\to Y \quad \backslash\quad f:X \to Y \quad \backslash)$$

2.7 Акценты

Для постановки разных значков над *одиночными* буквами в формулах предусмотрены команды, перечисленные в таблице 12, в которой эти значки поставлены над буквой *a*. Команды, производящие акценты из трех и четырех точек, определены в пакете `amsmath`. Они также приведены в таблице 12. Кроме того, пакет `amsmath` вводит команды, одноимённые с командами `LaTeX`, но имена которых начинаются с заглавной буквы. Они рекомендуются для двойных акцентов, хотя в современных версиях `LaTeX` разница уже не улавливается:

$$\hat{S} \quad \hat{S} \quad \backslash(\backslash\text{Hat}\{\backslash\text{Vec}\{S\}\} \quad \backslash\quad \backslash\text{hat}\{\backslash\text{vec}\{S\}\}\backslash)$$

Таблица 12: Акценты математической моды

\acute{a}	<code>\acute{a}</code>	\check{a}	<code>\check{a}</code>	\grave{a}	<code>\grave{a}</code>	\vec{a}	<code>\vec{a}</code>
\bar{a}	<code>\bar{a}</code>	\dot{a}	<code>\dot{a}</code>	\hat{a}	<code>\hat{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\breve{a}	<code>\breve{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\tilde{a}	<code>\tilde{a}</code>	$\overset{\cdot\cdot\cdot}{a}$	<code>\overset{\cdot\cdot\cdot}{a}</code>

^aТребуется подключение пакета `amsmath`.

Заметим, что акценты принято ставить не над буквами с точкой *i* и *j*, а над символами *i* и *j* из таблицы 20.

2.8 Корни

Квадратный корень набирается с помощью команды `\sqrt`, обязательным аргументом которой является подкоренное выражение; корень произвольной степени набирается с помощью той же команды `\sqrt` с необязательным аргументом — показателем корня. Пример:

$$\sqrt[3]{x^3} = x \quad \sqrt{x^2} = |x| \quad \backslash[\backslash\text{sqrt}[3]{x^3}=x \quad \backslash\quad \backslash\text{sqrt}\{x^2\}=|x|\backslash]$$

Команда `\sqrt` автоматически выбирает размер знака радикала таким образом, чтобы он точно соответствовал высоте подкоренного выражения. Иногда такой выбор приводит к не очень удачному результату:

$$\sqrt{y} + \sqrt{d} \quad \backslash(\backslash\text{sqrt}\{y\} + \backslash\text{sqrt}\{d\} \backslash)$$

Дело тут, конечно, в том, что буквы *y* и *d* имеют разную высоту и глубину. Чтобы сделать знаки радикала одинаковыми, надо добавить в подкоренные выражения по символу одинакового размера. Этот символ, естественно, не должен печататься и не должен занимать места по горизонтали. Такой невидимый символ создаёт команда `\mathstrut`:

$$\sqrt{y} + \sqrt{d} \quad \backslash\backslash\text{sqrt}\{\backslash\text{mathstrut } y\} + \backslash\text{sqrt}\{\backslash\text{mathstrut } d\}\backslash$$

`\mathstrut` — это невидимый символ, который занимает по высоте столько же места, сколько открывающая скобка (, и не имеет ширины. Он является частным случаем символов «фантомов», которые обсуждаются на стр. 37.

Пакет `amsmath` вводит команды `\leftroot` и `\uproot`, которые корректируют положение показателя корня. Значения аргументов команд, задающие смещение показателя корня по горизонтали и вертикали, подбираются вручную. Пример:

$$\sqrt[\beta]{k}, \quad \sqrt[\beta]{k} \quad \backslash(\backslash\text{sqrt}[\backslash\beta]{k}, \quad \backslash\quad \backslash\text{sqrt}[\backslash\leftroot\{1\}\uproot\{3\}\backslash\beta]{k} \backslash)$$

2.9 Дроби

Дроби, обозначаемые косой чертой (так рекомендуется обозначать дроби во внутритекстовых формулах), набираются непосредственно:

Неравенство $x + 1/x \geq 2$ выполнено для всех $x > 0$. Неравенство `\(x+1/x \ge 2 \)` выполнено для всех `\(x>0 \)`.

Если вы используете в формуле десятичные дроби, в которых дробная часть отделена от целой с помощью запятой (согласно российским правилам), то эту запятую следует взять в фигурные скобки. В противном случае после нее будет оставлен небольшой пробел, поскольку запятая считается знаком препинания.

$\pi \approx 3,14$ `\(\pi \approx 3{,}14 \)`

Дроби, в которых числитель расположен над знаменателем, набираются с помощью команды `\frac`. Эта команда имеет два обязательных аргумента: первый — числитель, второй — знаменатель. Если числитель и/или знаменатель дроби записывается одной буквой или цифрой, то можно их и не брать в фигурные скобки. Пример:

$\frac{1}{2} + \frac{x}{2} = \frac{1+x}{2}$ `\[\frac{1}{2} + \frac{x}{2} = \frac{1+x}{2} \]`

TeX автоматически выбирает шрифт для числителя и знаменателя дробей, созданных командой `\frac`. В случае цепных дробей этот выбор явно неудачен:

$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$ `\[\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}} \]`

В подобных случаях надо указывать размер символов самому:

$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$ `\[\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}} \]`
`\displaystyle` перед `\frac` необходима для того, чтобы вторая дробь набиралась в стиле `displaystyle`, невзирая на то, что она стоит в знаменателе.

Команда `\displaystyle` перед `\frac` необходима для того, чтобы вторая дробь набиралась в стиле `displaystyle`, невзирая на то, что она стоит в знаменателе.

Пакет `amsmath` вводит команду `\cfrac`, которая всегда начинает набирать дробь в стиле `displaystyle`:

$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}$ `\[\cfrac{1}{\sqrt{2} + \cfrac{1}{\sqrt{2} + \dots}} \]`

Кроме того, команда `\cfrac` имеет перед обязательными аргументами один необязательный аргумент со значением `l` или `r`. Задавая соответствующую опцию, можно сместить числитель к левому или правому краю разделительной черты.

Пакет `amsmath` вводит также команды `\dffrac` (сокращение команды `\displaystyle\frac`) и `\tffrac` (сокращение команды `\textstyle\frac`), которые позволяют задавать нестандартный размер символов в дробях во внутритекстовых формулах

$\frac{1}{k} \log_2 f(x)$ `\(\frac{1}{k} \log_2 f(x) \)`
 $\frac{1}{k} \log_2 f(x)$ `\(\dffrac{1}{k} \log_2 f(x) \)`

и в выделенных формулах:

$\sqrt{\frac{1}{k} \log_2 f(x)}$ `\[\sqrt{\frac{1}{k} \log_2 f(x)} \]`
 $\sqrt{\frac{1}{k} \log_2 f(x)}$ `\[\sqrt{\tffrac{1}{k} \log_2 f(x)} \]`

2.10 Операторы с пределами

Обсудим, как можно получить формулу

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (17)$$

с записями над и под знаками операций. В случае оператора \sum они называются «пределы суммирования», поэтому записи над и под знаком операций принято называть пределами (по-английски *limits*). В исходном тексте пределы обозначаются точно так же, как индексы; следовательно, формулу (17) можно получить так:

```
\[
  \lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}
\]
```

В этом примере существенно, что формула выделена в отдельную строку; во внутритекстовой формуле пределы печатаются на тех же местах, что и индексы:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad \left(\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \right)$$

Заметим, что сам размер оператора \sum также изменился. В таблице 13 приведены операторы, имеющие разный размер во внутритекстовых формулах и в формулах, выделенных в отдельную строку. У этих операторов индексы ведут себя как пределы у оператора \sum .

Таблица 13: Символы переменного размера

\cap	\bigcap	\otimes	\bigotimes	\wedge	\bigwedge	\prod	\prod	\prod	\prod
\cup	\bigcup	\sqcup	\bigsqcup	\coprod	\coprod	\sum	\sum	\sum	\sum
\odot	\bigodot	\oplus	\bigoplus	\int	\int	\oint	\oint	\oint	\oint
\oplus	\bigoplus	\vee	\bigvee	\oint	\oint	\oint	\oint	\oint	\oint

Индексы ведут себя как пределы не только у операторов переменного размера, но и у приведённых на стр. 28 операторов типа \lim , которые печатают имя операции прямым шрифтом без изменения размера букв.

2.10.1 Управление расположением пределов

Если надо, чтобы пределы у какого-либо оператора стояли не над и под знаком оператора, а сбоку, то после команды для знака оператора надо поставить команду \nolimits , а уже после неё — обозначения для пределов:

$$\prod_{i=1}^n i = n! \quad \left[\prod_{i=1}^n i = n! \right]$$

Пакет `amsmath` имеет опцию `sumlimits|nosumlimits`, которая контролирует размещение пределов у операторов переменного размера, кроме знака интеграла. `sumlimits` действует по умолчанию. Опция `nosumlimits` равносильна команде `\nolimits` во всех формулах.

Действие, обратное действию команды `\nolimits`, делает команда `\limits`. Пример её использования приведён в следующем разделе.

2.10.2 Интегралы

Для набора знаков интеграла используются команды `\int` для обычного интеграла и `\oint` для контурного интеграла. Пределы интегрирования по умолчанию печатаются сбоку как индексы не только в формулах внутри текста, но и в выделенных формулах:

$$\int_0^1 f(x) dx \quad \oint_C \mu dl \quad \begin{array}{l} \backslash[\\ \backslash\int_0^1 f(x)\backslash, dx \quad \backslash\oint_C \mu\backslash, dl \\ \backslash] \end{array}$$

Чтобы пределы интегрирования стояли над и под знаком интеграла, надо сразу после `\int` (или `\oint`) поставить команду `\limits`, а уже после нее — обозначения для пределов интегрирования:

$$\int_0^\infty y(x) dx \quad \oint_C dl \quad \begin{array}{l} \backslash[\backslash\int\limits_0^\infty y(x)\backslash, dx \\ \quad \quad \quad \backslash\quad \\ \backslash\oint\limits_C dl \backslash] \end{array}$$

Пакет `amsmath` имеет опцию `intlimits|nointlimits`, которая контролирует размещение пределов у знаков интегрирования. `nointlimits` действует по умолчанию. Опция `intlimits` равносильна команде `\limits` во всех формулах.

Пакет `amsmath` вводит команды `\iint`, `\iiint` и `\iiint` для печати кратных интегралов и команду `\idotsint` для печати двух знаков интеграла с многоточием между ними:

$$\iint dx dy \quad \iiint dx dy dz \quad \iiint dV \quad \begin{array}{l} \backslash\iint dx dy \quad \backslash\iiint dx dy dz \quad \backslash\iiint dV \\ \backslash[\\ \quad \quad \quad \backslash\idotsint\limits_V d\vec{x} \\ \backslash] \end{array}$$

Разнообразные знаки интегралов, в том числе кратных, можно найти в пакете `txfonts`, например:

$$\oint \quad \int \quad \begin{array}{l} \backslash[\backslash\varointclockwise \quad \backslash\quad \backslash\int \backslash] \end{array}$$

2.10.3 Многострочные и сторонние индексы

Пакет `amsmath` вводит команду `\substack` для набора многострочных индексов у символов переменного размера. Индексы разбиваются на строки, как обычно, командой `\\`. Синтаксис команды ясен из следующего примера:

$$\sum_{\substack{i \in \Lambda \\ 0 < j < n}} P(i, j) \quad \begin{array}{l} \backslash[\\ \backslash\sum_{\substack{i \in \Lambda \\ 0 < j < n}} P(i, j) \\ \backslash] \end{array}$$

Возможности команды `\substack` расширяют командные скобки `subarray`. Они имеют обязательный аргумент, который указывает, как должны выравниваться строки индексов. Допустимые значения `l`, `c` и `r` соответствуют выравниванию по левому краю, по центру и правому краю, соответственно. Пример:

$$\sum_{\substack{i \in \Lambda \\ 0 < j < n}} P(i, j) \quad (18) \quad \begin{array}{l} \backslash\begin{equation} \\ \backslash\sum_{\substack{\begin{subarray}{l} i \in \Lambda \\ 0 < j < n \end{subarray}}} P(i, j) \\ \backslash\end{subarray}} \\ \backslash\end{equation} \end{array}$$

Пакет `amsmath` вводит также команду `\sideset` для печати индексов по углам символов переменного размера. Синтаксис команды демонстрируют следующие примеры:

$$\prod_k^2 \prod_3^4 \quad \sum'_{0 \leq i \leq m} E_i \beta x \quad \begin{array}{l} \backslash[\\ \backslash\sideset{}_{_1^2}{}_{_3^4} \backslash\prod_k \quad \backslash\quad \backslash\quad \\ \backslash\sideset{}{\prime} \backslash\sum_{0 \leq i \leq m} E_i \backslash\beta x \\ \backslash] \end{array}$$

Видно, что команда `\sideset` позволяет легко поставить штрих у знака суммы. В рамках \LaTeX для того, чтобы штрих не размещался на месте верхнего предела, приходится создавать новый оператор «сумма со штрихом», используя команду `\mathop`:

$$\sum'_{x \in \Gamma} f(x) \quad \sum'_{x \in \Gamma} f(x) \quad \begin{array}{l} \backslash [\\ \backslash \text{sum}'_{x \in \Gamma} f(x) \quad \backslash \text{qqquad} \\ \backslash \text{mathop}\{\{\backslash \text{sum}}'\}_{x \in \Gamma} f(x) \\ \backslash] \end{array}$$

2.11 Скобки и другие разделители

Круглые и квадратные скобки набираются просто так, для фигурных скобок используются команды `\{` и `\}`, для угловых также есть специальные команды `\langle` (левая угловая скобка `<`) и `\rangle` (правая угловая скобка `>`).

2.11.1 Скобки переменного размера

Если заключённый в скобки фрагмент формулы занимает много места по вертикали (за счет дробей, степеней и тому подобного), то и сами скобки должны быть больше размером, чем обычные. В \TeX на этот случай предусмотрен механизм автоматического выбора размера скобок. Пользуются им так.

В формуле

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n$$

скобки обычного размера вокруг $1 + \frac{1}{n}$ смотрелись бы плохо; поэтому при её наборе надо поставить команду `\left` перед открывающей скобкой (`(` и команду `\right` перед закрывающей скобкой (`)`):

```
\[ e=\lim_{n\to\infty} \left( 1+\frac{1}{n} \right) ^n \]
```

Если перед одной скобкой стоит `\left`, а перед другой скобкой стоит `\right`, то на печати размер этих скобок будет соответствовать высоте фрагмента формулы, заключенного между `\left` и `\right`.

2.11.2 Разделители

Конструкция с `\left` и `\right` применима не только к круглым скобкам. В таблице 14 перечислены скобки и другие символы, которые с помощью `\left` и `\right` автоматически принимают нужный размер. Их называют разделители (по-английски *delimiters*).

Таблица 14: Разделители

<code>(</code>	<code>(</code>	<code>)</code>	<code>)</code>	<code>\uparrow</code>	<code>\Uparrow</code>
<code>[</code>	<code>[</code>	<code>\lbrack</code>	<code>]</code>	<code>\downarrow</code>	<code>\Downarrow</code>
<code>{</code>	<code>\{</code>	<code>\lbrace</code>	<code>\}</code>	<code>\updownarrow</code>	<code>\Updownarrow</code>
<code>\lfloor</code>	<code>\lfloor</code>	<code>\rfloor</code>	<code>\rfloor</code>	<code>\lceil</code>	<code>\rceil</code>
<code>\langle</code>	<code>\langle</code>	<code>\rangle</code>	<code>\rangle</code>	<code>/</code>	<code>\backslash</code>
<code> </code>	<code> </code>	<code>\vert</code>	<code>\ </code>	<code>\ </code>	<code>\Vert</code>

Формулы типа $|x|$ и $\|A\|$ в исходном тексте становятся более ясными, если вместо `|` и `\|` использовать разные команды для левых (`\lvert` и `\lVert`) и правых (`\rvert` и `\rVert`) разделителей. Они определены в пакете `amsmath`. Пример:

```
|x| \quad \|A\| \quad \left( \lvert x\rvert \right) \quad \left( \lVert A\rVert \right)
```

Отметим, что команды `\left` и `\right` должны присутствовать в формуле парами, иначе `TeX` выдаст сообщение об ошибке. Вместе с тем вовсе не требуется, чтобы разделители при командах `\left` и `\right` были расположены сколько-нибудь осмысленно: можно написать что-нибудь вроде `\left(...\right]` или даже `\left)...\right(`.

В таблице 15 собраны большие разделители, а в таблице 16 разделители из пакета `amssymb`. Большие разделители *должны* использоваться в конструкции с `\left` и `\right` или с командами серий `\Big...`, `\bigg...` и `\Bigg...`. Они обсуждаются в разделе 2.11.4.

Таблица 15: Большие разделители

```
( \lgroup ) \rgroup { \lmoustache } \rmoustache
| \arrowvert || \Arrowvert | \bracevert
```

Таблица 16: Разделители AMS (пакет `amssymb`)

```
⌈ \ulcorner ⌋ \urcorner ⌌ \llcorner ⌍ \lrcorner
```

2.11.3 Разделители без пары

Вместо разделителя после команды `\left` или `\right` можно поставить точку. На месте этой точки *ничего* не напечатается, а другой разделитель будет необходимого размера. Пример:

$$\int_a^b \frac{1}{2}(1+x)^{-3/2} dx = -\frac{1}{\sqrt{1+x}} \Big|_a^b$$

```
\[ \int\limits_a^b \frac{1}{2}(1+x)^{-3/2} dx = -\frac{1}{\sqrt{1+x}} \Big|_a^b \]
```

Другой важный пример — вёрстка большой фигурной скобки у систем уравнений и систем с условиями — приведён на стр. 32.

2.11.4 Команды, задающие размер разделителя

Размер разделителей, выбранный `TeX`ом с помощью команд `\left` и `\right`, часто бывает завышенным, поскольку подгоняется под полную высоту формулы, включая индексы. В таких случаях лучше самому задавать размеры разделителей, используя команды

```
\bigl, \Bigl, \biggl и \Biggl
```

для левых разделителей, команды

```
\bigr, \Bigr, \biggr и \Biggr
```

для правых разделителей, команды

```
\bigm, \Bigm, \biggm и \Biggm
```

для разделителей, стоящих посередине, когда нужно вставить небольшой пробел по обе стороны от разделительного символа, и команды

```
\big, \Big, \bigg и \Bigg
```


Таблица 18: Стрелки AMS (пакет amssymb)

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftleftarrows	<code>\leftleftarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>	\Uparrow	<code>\upuparrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>	\Downarrow	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Uparrowleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\Uparrowright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\Downarrowleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\Downarrowright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

Таблица 19: Отрицательные стрелки AMS (пакет amssymb)

\nleftarrow	<code>\nleftarrow</code>	\nrightarrow	<code>\nrightarrow</code>	\nleftrightarrow	<code>\nleftrightarrow</code>
\nLleftarrow	<code>\nLleftarrow</code>	\nRrightarrow	<code>\nRrightarrow</code>	\nLleftrightarrow	<code>\nLleftrightarrow</code>

2.13 Неклассифицированные символы

В этом разделе собраны символы, которые не относятся ни к одной из ранее обсуждавшихся категорий. В таблице 20 приведены неклассифицированные символы \LaTeX .

Таблица 20: Дополнительные символы

\aleph	<code>\aleph</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>	\Re	<code>\Re</code>
\angle	<code>\angle</code>	\exists	<code>\exists</code>	\jmath	<code>\jmath^a</code>	\sharp	<code>\sharp</code>
\perp	<code>\perp</code>	\flat	<code>\flat</code>	\mathring{U}	<code>\mathring{U}^b</code>	\spadesuit	<code>\spadesuit</code>
\Box	<code>\Box^b</code>	\forall	<code>\forall</code>	∇	<code>\nabla</code>	\surd	<code>\surd</code>
\clubsuit	<code>\clubsuit</code>	\hbar	<code>\hbar</code>	\natural	<code>\natural</code>	\top	<code>\top</code>
\Diamond	<code>\Diamond^b</code>	\heartsuit	<code>\heartsuit</code>	\neg	<code>\neg, \not</code>	\triangle	<code>\triangle</code>
\diamondsuit	<code>\diamondsuit</code>	\Im	<code>\Im</code>	∂	<code>\partial</code>	\wp	<code>\wp</code>
ℓ	<code>\ell</code>	\imath	<code>\imath^a</code>	\prime	<code>\prime</code>		

^aНужны для постановки дополнительных значков над буквами i и j .

^bТребуется подключение пакета `latexsym`.

В таблицах 21 и 22 приведены символы, которые можно использовать как в математической, так и в текстовой моде.

Таблица 21: Символы, доступные в математический и в текстовом моде.

\copyright	<code>\copyright</code>	\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>	\S	<code>\S</code>
--------------	-------------------------	-----------	----------------------	------------	-----------------------	------	-----------------	-----------	----------------------	------	-----------------

Таблица 22: Символы AMS математический и текстовом моды (пакет amsfonts или amssymb).

\checkmark	<code>\checkmark</code>	\textcircled{R}	<code>\textcircled{R}</code>	\maltese	<code>\maltese</code>
--------------	-------------------------	-------------------	------------------------------	------------	-----------------------

В таблице 23 собраны неклассифицированные символы $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.

Таблица 23: Дополнительные символы AMS (пакет amssymb)

\sphericalangle	<code>\angle</code>	\complement	<code>\complement</code>	\sphericalangle	<code>\measuredangle</code>
\backprime	<code>\backprime</code>	\diagdown	<code>\diagdown</code>	\mho	<code>\mho</code>
\Bbbk	<code>\Bbbk</code>	\diagup	<code>\diagup</code>	\nexists	<code>\nexists</code>
\bigstar	<code>\bigstar</code>	\eth	<code>\eth</code>	\sphericalangle	<code>\sphericalangle</code>
\blacklozenge	<code>\blacklozenge</code>	\Finv	<code>\Finv</code>	\square	<code>\square</code>
\blacksquare	<code>\blacksquare</code>	\Game	<code>\Game</code>	\triangledown	<code>\triangledown</code>
\blacktriangle	<code>\blacktriangle</code>	\hbar	<code>\hbar</code>	\varnothing	<code>\varnothing</code>
\blacktriangledown	<code>\blacktriangledown</code>	\hslash	<code>\hslash</code>	\vartriangle	<code>\vartriangle</code>
\circledS	<code>\circledS</code>	\lozenge	<code>\lozenge</code>		

К букве `\aleph` (\aleph) $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ добавил ещё три буквы иврита. Они приведены в таблице 24.

Таблица 24: Буквы иврита AMS (пакет amssymb)

\beth	<code>\beth</code>	\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>
---------	--------------------	-----------	----------------------	----------	---------------------

2.14 Надстрочные и подстрочные знаки

2.14.1 Шляпки и тильды

Команды `\widetilde` и `\widehat` создают «шляпки» и «тильды», охватывающие до трёх символов:

$$\widetilde{xyz} \quad \widetilde{XYZ} \quad \widehat{f * g} = \hat{f} \cdot \hat{g} \quad \begin{array}{l} \backslash(\quad \backslashwidetilde{xyz} \quad \backslashquad \quad \backslashwidetilde{XYZ} \\ \backslashquad \quad \backslashwidehat{f * g} = \backslashhat f \backslashcdot \backslashhat g \quad \backslash \end{array}$$

2.14.2 Линии

Команды `\overline` и `\underline` создают *горизонтальные* линии сразу над или под своим аргументом:

$$\underline{m+n} \quad \overline{z+k} \quad \begin{array}{l} \$\underline{m+n} \quad \backslashqqquad \\ \overline{\backslashstrut z} + \overline{\backslashstrut k} \$ \end{array}$$

Здесь использована команда `\strut` (невидимая линейка нулевой ширины). Благодаря ей во второй формуле чёточки располагаются на одной высоте: линейка простирается чуть выше максимальной высоты букв текущего шрифта и чуть ниже уровня, до которого могут опускаться буквы текущего шрифта.

2.14.3 Фигурные скобки

Команды `\overbrace` и `\underbrace` создают *длинные горизонтальные* фигурные скобки сразу над и под своим аргументом, соответственно:

$$\underbrace{a+b+\dots+z+1+\dots+10}_{26} \quad \begin{array}{l} \backslash[\\ \overbrace{\underbrace{a+b+ \\ \backslashcdots+z}_{26}+1+\backslashcdots+10}^{36}} \\ \backslash] \end{array}$$

Надпись над скобкой, которую создаёт команда `\overbrace`, оформляется как верхний индекс. Надпись под скобкой, которую создаёт команда `\underbrace`, оформляется как нижний индекс.

2.14.4 Стрелки

Команды `\overleftarrow` и `\overrightarrow` ставят над формулой стрелки, направленные влево и вправо, соответственно:

$$\overleftarrow{ABC + abc} \quad \text{\$}\overleftarrow{ABC+\overrightarrow{abc}}\text{\$}$$

Команды `\underrightarrow` и `\underleftarrow` из пакета `amsmath` рисуют правую и левую стрелки под формулой, например, $\underleftarrow{\psi_\delta(t)E_t h}$ и $\underrightarrow{\psi_\delta(t)E_t h}$.

Команды `\overleftrightharrow` и `\underleftrightharrow` из того же пакета рисуют двухсторонние стрелки над и под формулой, соответственно: $\overleftrightharrow{\psi_\delta(t)E_t h}$, $\underleftrightharrow{\psi_\delta(t)E_t h}$.

На заметку Стрелка `\overrightarrow` используется обычно для набора вектора. Вектор лучше смотрится, если выражение, над которым стоит стрелка, сдвинуть влево описанной на стр. 36 командой `\mkern`, оставив саму стрелку на месте. Сравните:

$$\overrightarrow{AB} \dots\dots\dots \overrightarrow{AB}$$

$$\overrightarrow{\mkern -3mu AB\mkern 3mu} \dots\dots\dots \overrightarrow{AB}$$

2.14.5 Произвольные символы

Чтобы расположить одну часть формулы вровень с остальным текстом, а другую — над ней, используется команда `\stackrel`. У этой команды два аргумента: первый — то, что будет над строкой, второй — то, что останется в строке:

$$x^2 \stackrel{\text{def}}{=} x \cdot x \quad \text{\textbackslash}(x^2\stackrel{\text{def}}{=}x\cdot x)\text{\textbackslash}$$

Пакет `amsmath` вводит команды `\overset` и `\underset` с двумя аргументами. Первый аргумент печатается шрифтом индексного стиля над и под формулой, стоящей во втором аргументе, соответственно.

$$\overset{*}{X} \quad X \quad \underset{*}{X} \quad \overset{a}{\underset{b}{X}} \quad \text{\textbackslash}(\overset{*}{X} \quad \underset{*}{X} \quad \overset{a}{\underset{b}{X}})\text{\textbackslash}$$

2.15 Стрелки с индексами

Команды `\xleftarrow` и `\xrightarrow` из пакета `amsmath` производят растяжимые стрелки с индексами. Первый (необязательный) аргумент размещается под стрелкой, а второй (обязательный) — над ней:

$$0 \xleftarrow[\zeta]{\alpha} F \times \Delta[n-1] \xrightarrow{\partial_0 \alpha(b)} E^{\partial_0 b}$$

$$\text{\textbackslash}[0 \xleftarrow[\zeta]{\alpha} F \times \Delta[n-1] \xrightarrow{\partial_0 \alpha(b)} E^{\partial_0 b}]\text{\textbackslash}$$

2.16 Биномиальные коэффициенты AMS

Пакет `amsmath` вводит для биномиальных выражений аж три команды `\binom`, `\dbinom` и `\tbinom`. Последние две команды печатают бином всегда в стиле `displaystyle` и `textstyle`, соответственно:

$$\binom{k}{n} \quad \binom{k}{n} \quad \text{\textbackslash}[\ \binom{k}{n} \quad \tbinom{k}{n}]\text{\textbackslash}$$

$$\binom{k}{n} \quad \binom{k}{n} \quad \text{\textbackslash}(\ \binom{k}{n} \quad \dbinom{k}{n})\text{\textbackslash}$$

Таблица 26: Тригонометрические и гиперболические функции для России

\arctg	\ch	\ctg	\sh	\th^a
\arcctg	\cosec	\cth	\tg	

^aКоманда `\th` определена также в кодировке T1 для печати в текстовой моде символа β .

Таблица 27: Функции, принятые в России

P	<code>\Prob</code>	н.о.д.	<code>\nod^a</code>	н.о.к.	<code>\nok^a</code>	Пр	<code>\Proj^a</code>
D	<code>\Variance</code>	НОД	<code>\NOD^a</code>	НОК	<code>\NOK^a</code>		

^aТребуется подключение пакета `mathtext`.

3.2 Функции с пределами

Функции типа `\lim` печатаются прямым шрифтом; между именем функции и её аргументом автоматически вставляется маленький пробел. Но в отличие от функций типа `\log` из таблицы 25, у них расстановка индексов подчиняется тем же правилам, которые действуют для операторов с пределами и которые описаны на стр. 19:

$$\max_{1 \leq n \leq m} \log_2 P_n \quad \backslash [\ \max_{\{1 \leq n \leq m\}} \log_2 P_n \ \backslash]$$

Команды, которые печатают имена функции с пределами, приведены в таблице 28.

Таблица 28: Математические функции с пределами

<code>\limsup</code>	<code>\limsup</code>	<code>\lim</code>	<code>\lim</code>	<code>\max</code>	<code>\max</code>	<code>\det</code>	<code>\det</code>	<code>\sup</code>	<code>\sup</code>
<code>\liminf</code>	<code>\liminf</code>	<code>\inf</code>	<code>\inf</code>	<code>\min</code>	<code>\min</code>	<code>\gcd</code>	<code>\gcd</code>	Pr	<code>\Pr</code>

Пакет `amsgn` (автоматически загружается при подключении пакета `amsmath`) вводит ряд экзотических функций класса `\lim`. Они приведены в таблице 29.

Таблица 29: Математические функции с пределами AMS (пакет `amsgn`)

\varliminf	<code>\varliminf</code>	$\overline{\lim}$	<code>\varlimsup</code>	<code>\projlim</code>	<code>\projlim</code>
\varinjlim	<code>\varinjlim</code>	\varprojlim	<code>\varprojlim</code>		

3.3 Определение новых имен операций

Математические функции, такие как `\log` и `\lim`, традиционно печатаются прямым шрифтом. Поскольку появляются всё новые и новые названия, пакет `amsgn` ввёл общий механизм для определения таких математических операций. Чтобы определить функцию типа `\log`, например `\meas`, надо просто её декларировать:

```
\DeclareMathOperator{\meas}{meas}
```

Второй аргумент у `\DeclareMathOperator` — это текст, который будет напечатан на месте команды. Пример с новой командой:

$$\text{meas}_1 \{u \in R_+^1 : f^*(u) > \alpha\} \quad (19) \quad \begin{array}{l} \backslash \text{begin}\{\text{align}\} \\ \backslash \text{meas}_1 \{u \in R_+^1 \text{colon } f^*(u) > \alpha \backslash \} \\ \backslash \text{end}\{\text{align}\} \end{array}$$

Если у нового оператора индексы должны размещаться как у оператора `lim`, то надо воспользоваться декларацией `\DeclareMathOperator*`, например,

$$\|f\|_\infty = \operatorname{ess\,sup}_{x \in R^n} |f(x)| \quad (20)$$

```

\DeclareMathOperator*{\esssup}{ess\,sup}
\begin{align}
\|f\|_\infty &= \operatorname{ess\,sup}_{x \in R^n} |f(x)| \\
\end{align}

```

3.4 Функции модуля

Для функции модуля есть две команды: `\bmod` для бинарного оператора

$$a \bmod b \quad \backslash (a \bmod b \backslash)$$

и `\pmod` для выражений вроде

$$x \equiv a \pmod{b} \quad \backslash (x \equiv a \pmod{b} \backslash)$$

Пакет `amsmath` несколько уменьшает пробелы у команды `\pmod` во внутритекстовых формулах и вводит ещё две команды: `\mod` и `\pod`. `\mod` опускает круглые скобки, а `\pod` опускает «mod» и оставляет круглые скобки.

$$x \equiv y \bmod c; \quad x \equiv y \pmod{d} \quad x \equiv y \bmod c; \quad x \equiv y \pmod{d}$$

4 Конструкции для многострочных выражений

4.1 Матрицы

Чтобы набрать матрицу, надо воспользоваться окружением `array` в математической моде (само окружение не переключает `TeX` в математическую моду!). Разберём такой пример:

$$\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array}$$

```

\begin{array}{cccc}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{array}

```

Всякая матрица состоит из строк и столбцов; в исходном тексте строки матрицы разделяются командой `\\` (последнюю строку заканчивать командой `\\` не надо), а элементы внутри одной строки, относящиеся к разным столбцам, отделяются друг от друга символами `&`. Сколько и каких столбцов должно быть в матрице, задаётся в обязательном аргументе, который следует сразу за `\begin{array}`. В нашем примере он представляет собой четыре буквы `cccc`. Это значит, что в матрице 4 столбца (по букве на столбец), и что содержимое каждого из этих столбцов должно быть расположено по центру столбца (поскольку каждая из букв — буква `c`). Кроме `c`, в аргументе окружения могут стоять буквы `l` или `r`, означающие, что содержимое этих столбцов будет выравнено по левому краю или по правому краю, соответственно.

Скобки или иные разделители вокруг матрицы ставятся с помощью конструкции `\left` и `\right`, описанной на стр. 21. Например, чтобы заключить матрицу в круглые скобки, надо перед `\begin{array}` написать `\left(`, а после `\end{array}` — `\right)`. Если загрузить пакет `delarray` из коллекции `tools`, то разделители вокруг матрицы можно задавать прямо в окружении `array` вокруг его аргумента; в этом случае даже матрицы типа

$$J = \left\| \begin{array}{cc} \left(\begin{array}{cc} \lambda & 0 \\ 0 & \lambda \end{array} \right) & 0 \\ 0 & \left(\begin{array}{cc} \mu & 0 \\ 0 & \mu \end{array} \right) \end{array} \right\|$$

набираются достаточно просто

```
\[ J = \begin{array}{|cc|}
\begin{array}{cc} \lambda & 0 \\ 0 & \lambda \end{array} & 0 \\
0 & \begin{array}{cc} \mu & 0 \\ 0 & \mu \end{array} \end{array} \]
```

Отметим, что в окружении `array` набор формулы начинается в стиле `textstyle`, поэтому размер символов в числителе и знаменателе дроби сразу печатается в стиле индексов, как у внутритекстовой формулы.

Окружение `array` можно использовать не только для вёрстки матриц: фактически оно создает таблицы, состоящие из строк и столбцов, например:

M	\leftarrow	$\begin{array}{c} O \\ \\ XY \\ \\ \infty \end{array}$	\rightarrow	N	<code>\[</code>	<code>\begin{array}{ccccc}</code>	<code>& &\mathcal{O}& &\& &\vline& &\&</code>	<code>\mathcal{M}</code>	<code>& \gets & XY &</code>	<code>\to & \mathcal{N}</code>	<code>\& &\vline& &\& &\infty& &\&</code>	<code>\end{array}</code>	<code>\]</code>
-----	--------------	--	---------------	-----	-----------------	-----------------------------------	---	--------------------------	---	------------------------------------	---	--------------------------	-----------------

Как видите, если какая-то графа в нашей таблице должна быть пуста, то между (или перед, если эта графа — первая в своей строке) соответствующими знаками `&` нужно просто ничего не писать (или оставить сколько угодно пробелов). Если после того, что вы написали в строке, до конца строки идут только пустые графы, то можно не дописывать до конца значки `&`, а сразу написать `\&`.

Окружение `array` перед обязательным аргументом может иметь необязательный аргумент с допустимыми значениями `t`, `c` (действует по умолчанию) и `b`. Он задаёт положение во вертикали осевой линии «матрицы». По умолчанию она проходит через середину. При `[t]` осевая линия самой матрицы совпадает с осевой линией её верхней строки, а при `[b]` — осевая линия самой матрицы совпадает с осевой линией её нижней строки. Сама матрица позиционируется в уравнении так, чтобы её осевая линия совпала с осевой линией уравнения (на уровне знака минус). Пример:

$A =$	$-$	a	$,$	$-$	$\frac{a}{b}$	$,$	$-$	b	<code>\[</code>	<code>A =</code>	<code>-\begin{array}[t]{c} a \\ b \end{array},</code>	<code>-\begin{array}{c} a \\ b \end{array},</code>	<code>-\begin{array}[b]{c} a \\ b \end{array}</code>	<code>\]</code>
-------	-----	-----	-----	-----	---------------	-----	-----	-----	-----------------	------------------	---	--	--	-----------------

Настройка Промежутки между элементами матрицы можно регулировать. Параметр `\arraycolsep` равен половине величины горизонтального пробела между двумя колонками (по умолчанию равен 5 pt). Значением команды `\arraystretch` является вещественное число, равное по умолчанию 1. Величина вертикального пробела между рядами получается путем умножения этого числа на величину пробела, принятую по умолчанию. Если декларацией `\renewcommand` заменить значение `\arraystretch` на 1.25, то расстояние между строками на печати будет больше в 1.25 раза.

4.1.1 Расчерчивание матрицы

Помимо `l`, `c` и `r`, в аргументе окружения `array` можно использовать символ `|`, который проводит вертикальную линию на всю высоту матрицы. Перед первой строкой, а также непосредственно вслед за `\&` и последней строкой можно поставить команду `\hline`, которая проведёт горизонтальную линию на всю ширину матрицы. Горизонтальную линию через часть столбцов можно провести командой `\cline`. Вертикальную линию через одну строку можно провести, заменяя ячейку командой `\multicolumn` и используя в её аргументе символ `|`. Пример:

$C =$	$\left($	$\begin{array}{ccc}$	1	$&$	0	$&$	0	$\&$	$\cline{2-3}$	0	$&$	$\multicolumn{1}{ c}{0}$	$&$	i	$\&$	0	$&$	$\multicolumn{1}{ c}{-i}$	$&$	0	\end{array}	$\right)$	<code>\[</code>	<code>C = \left(\begin{array}{ccc}</code>	<code>1 & 0 & 0 \\ \cline{2-3}</code>	<code>0 & \multicolumn{1}{ c}{0} & i \\ 0 & \multicolumn{1}{ c}{-i} & 0</code>	<code>\end{array} \right) \]</code>
-------	----------	----------------------	-----	-----	-----	-----	-----	------	---------------	-----	-----	--------------------------	-----	-----	------	-----	-----	---------------------------	-----	-----	---------------	-----------	-----------------	--	---	--	-------------------------------------

Настройка Толщину линий можно выбрать, изменяя значение параметра `\arrayrulewidth`.

Пакет `amstext` (загружается автоматически при загрузке пакета `amsmath`, но может использоваться самостоятельно) вводит команду `\text` для вставки небольших фрагментов текста в формулу. Текст, как и в случае команды `\mbox`, печатается шрифтом, который был текущим перед формулой, и, кроме того, сам может содержать формулы. Преимущество команды `\text` в том, что устанавливается, как при использовании математических алфавитов, правильный размер шрифта:

$$F_i(x) \text{ при } i = 1, \dots, I_{\max} \quad \backslash [F_i(x) \text{ при } i=1, \dots, I_{\max}]$$

5.1.1 Вставка текста между уравнениями

Для вставки текста в одну-две строки между уравнениями, которые должны быть выравнены по вертикали, предназначена команда `\intertext` из пакета `amsmath`. Команду можно ставить только сразу после команд `\` или `\`*. Пример:

$$\begin{array}{l}
 y_0 = 1 \\
 y_1 = \xi \\
 \text{и далее для } n = 2, 3, \dots \\
 y_n = y_{n-1} + \beta y_{n-2}
 \end{array}
 \quad
 \begin{array}{l}
 \backslash \text{begin}\{align*\} \\
 y_0 \ \&= \ 1 \ \backslash \backslash \\
 y_1 \ \&= \ \xi \ \backslash \backslash \\
 \backslash \text{intertext}\{\text{и далее для } n=2, 3, \dots\} \\
 y_n \ \&= \ y_{n-1} + \beta y_{n-2} \\
 \backslash \text{end}\{align*\}
 \end{array}$$

5.2 Математические алфавиты

Сменить математический шрифт, используемый в формулах по умолчанию, на математический же шрифт с другой формой символов можно специальными командами (их называют математическими алфавитами). Они действуют только на алфавитные символы: буквы, цифры, акценты и прописные греческие буквы. Математические символы остаются без изменения. Все математические алфавиты вместе с примерами формы символов приведены в таблице 30.

Таблица 30: Математические алфавиты

<code>\mathcal</code>	<i>ABCDEF GHIJK LMNOPQRST UVWXYZ</i>	
<code>\mathnormal</code>	<i>ABC, abc, 123, â, ã, ΨΩ</i>	<code>\mathrm</code> ABC, abc, 123, â, ã, ΨΩ
<code>\mathbf</code>	ABC, abc, 123, â, ã, ΨΩ	<code>\mathsf</code> ABC, abc, 123, â, ã, ΨΩ
<code>\mathtt</code>	ABC, abc, 123, â, ã, ΨΩ	<code>\mathit</code> ABC, abc, 123, â, ã, ΨΩ

^aСодержит только прописные латинские буквы.

На заметку Алфавит `\mathtt` не ставит акцент `\dot`. `\mathnormal` корректно расставляет акценты только после подключения пакета `amsmath`; но в этом случае в других алфавитах вместо `\vec` ставится `\tilde`, например, `\mathbf{\vec a}` печатает **ã**.

Коллекция пакетов `AMS-LATEX` вводит три новых математических алфавита для печати строчных и прописных латинских букв готическим шрифтом (`\mathfrak`), контуров прописных латинских букв (`\mathbb`) и каллиграфических прописных латинских букв шрифтом Euler (`\mathscr`). Если пакет `eucal` загрузить без опцией `mathscr`, то он подменит в алфавите `\mathcal` каллиграфический шрифт `LATEX` на каллиграфический шрифт Euler. Такой подмены не будет, если действует опция `mathscr`. В этом случае каллиграфический шрифт Euler доступен через алфавит `\mathscr`. Алфавиты `AMS-LATEX` приведены в таблице 31. Там же указаны пакеты, которые надо загрузить для работы с ними.

Наряду с пакетом `eucal`, алфавит `\mathscr` для печати прописных латинских букв с «рукописным» начертанием определен в пакете `mathrsfs`. Форма символом демонстрируется в таблице 32.

Таблица 31: Математические алфавиты AMS

```

\mathscra  ABCDEFGHIJKLMNOPQRSTUVWXYZ
\mathfrakb,c  ABCDEFGHIJKLMNOPQRSTUVWXYZ
              abcdefghijklmnopqrstuvwxyz
\mathbbc    ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

^aПакет eucal с опцией mathscr ^bПакет eufrak ^c Пакет amssymb или amssymb

Таблица 32: Каллиграфический алфавит RSFS (пакет mathrsfs)

```

\mathscr  ABCDEFGHIJKL MNO PQR STUVWXYZ

```

5.2.1 Кириллические математические алфавиты

Стиль russian пакета babel установил кириллические математические алфавиты. Кириллические буквы в математической моде можно использовать после загрузки пакета mathtext *перед* загрузкой пакета babel. По умолчанию буквы берутся из шрифта прямого начертания (в отличие от латинских букв):

```

Я = sin ю + z          \ ( Я = \sin ю + z \ )

```

Сменить шрифт, используемый по умолчанию, на другой, в том числе с курсивным начертанием, можно командами, перечисленными в таблице 33.

Таблица 33: Кириллические математические алфавиты (пакеты mathtext и babel)

```

\cyrmathrm  \cyrmathbf  \cyrmathsf  \cyrmathit  \cyrmathtt

```

Практически всегда можно обойтись без кириллических математических алфавитов, поскольку русские буквы не принято использовать в качестве имён переменных, а включение в формулы текста в виде индексов можно выполнить командой \text из пакета amstext:

```

2ой      Rэфф          \ ( 2\text{ой}}\qquad R_{\text{эфф}} \ )

```

5.3 Декларация нового алфавита

Декларацию нового математического алфавита рассмотрим на примере. Коллекция пакетов FontsC поддерживает для L^AT_EXа ряд PostScript шрифтов фирмы Параграф. Среди них имеется шрифт Studio Script Cyrillic с рукописным начертанием символов. Этот шрифт можно использовать для печати формул, которые выглядят как написанные от руки. Шрифт Studio Script не декларирован как математический алфавит. Но это можно сделать самому, поместив в преамбуле входного файла строку

```

\DeclareMathAlphabet{\mathtud}{OT1}{tud}{m}{it}

```

которая вводит алфавит \mathtud для печати в математической моде строчных и прописных латинских букв и цифр шрифтом со значениями атрибутов: кодировка OT1, гарнитура tud, насыщенность m и начертания it. Это и есть шрифт Studio Script. Пример:

```

Arccos x2          \ ( \mathtud{ \mathop{Arccos} x^2 } \ )

```

Поскольку команда \mathtud объявлена как математический алфавит, она не действует ни на скобки, ни на знаки арифметических операций, хотя все они имеются в шрифте. Если эти знаки нужны, то надо изменить определение команды \mathtud. Сначала в преамбуле входного файла шрифт Studio Script объявляем как символьный, скажем, mtud, а затем уже определяем команду \mathtud:

```
\DeclareSymbolFont{mtud}{OT1}{tud}{m}{it}
\DeclareSymbolFontAlphabet{\mathtud}{mtud}
```

Теперь можно ввести команды для набора скобок и знаков арифметических операций, например, «=» и «+»:

```
\DeclareMathSymbol{\Slp}{\mathopen}{mtud}{'()}
\DeclareMathSymbol{\Srp}{\mathclose}{mtud}{'')}
\DeclareMathSymbol{\Seq}{\mathrel}{mtud}{"3D}
\DeclareMathSymbol{\Splus}{\mathbin}{mtud}{"2B}
```

При определении математического символа указывается его класс (см. раздел 1.3) и код в шрифте. После этого можно печатать формулы типа:

$$y(x) = \sin x^2 + 3$$

```
\[ \mathtud{ y\Slp x\Srp \Seq
\mathop{sin} x^2 \Splus 3} \]
```

Более сложные формулы, содержащие знаки типа $\sqrt{\quad}$, смотрятся плохо: уж больно аккуратно эти знаки написаны.

5.4 Полу жирная насыщенность символов

Насыщенность математических символов задаётся *математической версией*, которую можно поменять декларацией `\mathversion`. Декларация имеет один обязательный аргумент, который может принимать одно из двух значений — `normal` (значение по умолчанию) и `bold`. Используется декларация *перед* формулой. Вот пример её действия:

$$M_2 \times \sqrt{1 + \sin \pi t}$$

```
\( M_2\times\sqrt{1 + \sin\pi t} \)
```

$$\mathbf{M}_2 \times \sqrt{1 + \sin \pi t}$$

```
\( \mathbf{M}_2\times\sqrt{1 + \sin\pi t} \)
```

Видно, что \LaTeX поменял насыщенность не только алфавитных символов, но и операторов. Если «полу жирных» шрифтов для символов переменного размера нет, то они не меняют насыщенность даже в `bold` версии.

Команда `\boldsymbol` из пакета `ambsy` (автоматически загружается при загрузке пакета `amsmath`) также устанавливает `bold` версию. Но, в отличие от `\mathversion`, она работает в математической моде и, следовательно, позволяет набирать полу жирным шрифтом не всю, а только часть формулы:

$$M_2 \sin \omega t$$

```
\( M_2 \sin\omega t \)
```

$$\mathbf{M}_2 \sin \omega t$$

```
\( \boldsymbol{M}_2 \sin\omega t \)
```

В пакете `ambsy` определена команда `\pmb` («*proof man's bold*»), которую можно использовать для получения полу жирной насыщенности символов даже тогда, когда нет соответствующего шрифта. Возникающие при этом проблемы с расстановкой пределов у операторов переменного размера решает команда `\mathop`:

$$\sum_{i=0}^{\infty} A_i$$

```
\mathversion{bold} \[
\sum_{i=0}^{\infty} A_i \qqquad
\mathop{\pmb{\sum}}_{i=0}^{\infty} A_i
\]
```

6 Настройка формул

6.1 Промежутки в математической моде

Бывают случаи, когда формулы с промежутками между символами, выбранными \TeX ом, выглядят неудачно. В этом случае их можно изменить вручную, вставляя между символами или подформулами положительный или отрицательный промежуток с помощью команд, собранных в таблице 34.

Таблица 34: Промежутки в математической моде

	Команда	Промежуток	Величина ^a
	<code>\qqquad</code>	┘ ┐	2 em
	<code>\quad</code>	┘ ┐	1 em
	<code>\;</code> <code>\thickspace^b</code>	┘┐	5 mu ^c
	<code>\:</code> <code>\medspace</code>	┘┐	4 mu
	<code>\,</code> <code>\thinspace</code>	┘┐	3 mu
	<code>\!</code> <code>\negthinspace</code>	┘┘	-3 mu
	<code>\!</code> <code>\negmedspace</code>	┘┘	-4 mu
	<code>\!</code> <code>\negthickspace</code>	┘┘	-5 mu

^aИмеет plus- и minus-компоненты (кроме «тонких пробелов»)

^bКоманды из этого столбца определены в пакете `amsmath`

^cМатематическая единица длины $1\text{mu} = 1/18\text{em}$

Сравните формулы, сверстанные без и со вставленными при наборе пробелами:

$$\int \int f dx dy \quad \sqrt{3}x \quad n/\ln x \quad \$ \int \int f dx dy \sqrt{3}x \quad n/\ln x \$$$

$$\int \int f dx dy \quad \sqrt{3}x \quad n/\ln x \quad \$ \int \! \! \! \int f \! \! \! \, dx dy \sqrt{3} \! \! \! \, x \quad n/\! \! \! \ln x \$$$

6.1.1 Пробелы произвольного размера

Команда `\mkern` позволяет вставлять положительные и отрицательные горизонтальные пробелы любой величины. Величина пробела задаётся в единицах mu . По аналогии с `\mkern`, пакет `amsmath` вводит команду `\mspace`. Её обязательный аргумент также выражается в единицах mu .

$$a \quad b \quad c \quad \backslash (a \mkern 25\text{mu} b \mspace{15\text{mu}} c \backslash)$$

6.2 Дублирование знаков при переносе формулы

При переносе формулы на другую строку $\text{T}_\text{E}_\text{X}$, вопреки российской традиции, не дублирует знак:

Теорема Пифагора утверждает, что $c^2 = a^2 + b^2$.
 Теорема Пифагора утверждает, что $\backslash (c^2 = a^2 + b^2 \backslash)$.

Для решения этой задачи надо воспользоваться командой $\text{T}_\text{E}_\text{X}$ `\discretionary`. Она имеет три аргумента. Внутри строки команда печатает текст из третьего аргумента. Но если она попадает в конец строки, то тогда на этом месте печатается первый аргумент, а в начале следующей — второй. В математической моде третий аргумент должен быть пустым.

Для дублирования знаков типа « \Rightarrow » или « \Leftarrow » определим новую команду с одним аргументом под названием `\hm`:

```
\newcommand{\hm}[1]{#1\nobreak\discretionary{}{\hbox{\ensuremath{#1}}}{}}
```

Команда ставится непосредственно перед символом, который должен быть продублирован при переносе строки:

Теорема Пифагора утверждает, что $c^2 = a^2 + b^2$.
 Теорема Пифагора утверждает, что $\backslash (c^2 \hm= a^2 + b^2 \backslash)$.

Внутри строки команда `\hm` ничего не печатает:

Уравнение $c^2 = a^2 + b^2$ даёт ...
 Уравнение $\backslash (c^2 \hm= a^2+b^2 \backslash)$ даёт `\dots`

Поскольку длинные выделенные формулы разбиваются на части вручную, дублировать знак, на котором происходит перенос формулы, также надо вручную.

6.3 Неразрывный дефис

Если в тексте встречаются выражения типа « n -мерный», то надо предотвратить перенос строки сразу после дефиса. Для этой цели пакет `amsmath` ввёл команду `\nobreakdash`. Переносы в слове, следующим за дефисом, можно разрешить, если добавить после дефиса пробел нулевой ширины.

Состояние системы описывается n -мерным вектором. Состояние системы описывается n -мерным вектором.

Команду `\nobreakdash` можно использовать для подавления возможного переноса и после короткого тире в выражениях типа «2–7». Для этого надо перед `--` поставить команду. Если определить новую команду, скажем, `\ndash`:

```
\newcommand{\ndash}{\nobreakdash--}
```

то выражение «2–7» будет набираться как `2\ndash 7`.

6.4 Невидимые символы

Иногда бывает полезно включить в формулу символ, который сам не печатается, но место занимает. Такие невидимые символы называются «фантомами». Если в формуле написать

```
\phantom{формула}
```

то результат будет такой же, как если бы *формула* была сначала напечатана, а затем аккуратно стёрта с бумаги. Пример:

Знак радикала $\sqrt{}$. Знак радикала $\sqrt{}$.

Вертикальный фантом формулы создаётся командой `\vphantom`. Он не занимает места по горизонтали. В частности, команда `\mathstrut` — это сокращение для `\vphantom{()}`. Формулы типа $\text{Fe}_2^{+2}\text{Cr}_2\text{O}_4$ смотрятся плохо, поскольку нижние индексы находятся на разной высоте. Проблему можно решить так:

```
\( \mathrm{Fe}_2^{+2}\text{Cr}_2^{\vphantom{+2}}\text{O}_4^{\vphantom{+2}} \)
```

даёт $\text{Fe}_2^{+2}\text{Cr}_2\text{O}_4$.

Горизонтальные фантомы создаются командой `\hphantom`:

На пустое место можно вписать На пустое место $\sin^2\alpha$ формулу вручную. можно вписать формулу вручную.

6.5 Видимые символы, незанимающие места

При вёрстке на каждый символ отводится столько места, сколько он занимает на самом деле. Однако, формула, являющаяся аргументом команды `\lefteqn` печатается, но при этом никакого места по горизонтали под эту формулу не отводится. Пример:

\aleph $\lefteqn{a}\backslash$

Команда `\smash`, подобно команде `\lefteqn`, печатает формулу, но при этом $\text{T}_\text{E}_\text{X}$ считает, что эта формула не занимает никакого места по вертикали. Пакет `amsmath` модифицирует команду `\smash`. Теперь она имеет необязательный аргумент, который может принимать значение `t` и `b`. В первом случае $\text{T}_\text{E}_\text{X}$ считает, что формула имеет нулевую высоту, а во втором — нулевую глубину. Полная высота при этом сохраняется. Пример:

$\sqrt{\lambda_i}$ $\sqrt{\lambda_i}$ $\left[\sqrt{\lambda_i} \quad \quad \sqrt{\smash[b]{\lambda_i}} \right]$

7 Теоремы, законы и др.

В математических текстах встречаются теоремы, леммы, определения, аксиомы и т.д. \LaTeX позволяет организовать нумерацию подобных структур таким образом, чтобы номера и ссылки на них создавались автоматически. Для этого имеется команда `\newtheorem`, которая создаёт окружения для набора различного вида «теорем». Объявления новых окружений делается в преамбуле документа. Существует два варианта команды `\newtheorem`:

```
\newtheorem{env} [theorem] {type}
\newtheorem{env} {type} [section]
```

В обоих случаях команда имеет два обязательных аргумента. Первый аргумент *env* — это название нового типа окружения, которое мы создаём. Оно не должно совпадать с именем уже существующего окружения или счётчика. Для каждого *env* создаётся счётчик с тем же именем, если в списке аргументов отсутствует *theorem*. При наличии этого аргумента для нумерации будет использоваться определённый ранее счётчик *theorem*: *env* и *theorem* будут иметь единую нумерацию. Второй аргумент *type* — это название типа «теоремы». По умолчанию он печатается полужирным шрифтом, а формулировка — курсивом. Абзац, идущий после окружения начинается с абзацным отступом, если после закрывающей окружение команды идёт пустая строка, и без отступа в противном случае.

Во втором варианте команды `\newtheorem` аргумент *section* — это имя уже существующего счётчика, обычно счётчика команд секционирования. Фактически выбирается раздел, внутри которого будут нумероваться «теоремы». Создаваемый счётчик *env* является внутренним для *section*: он будет автоматически обнуляться при каждом изменении значения счётчика *section*.

Создаваемые нами окружения могут иметь один необязательный аргумент, который обычно используется для указания автора; он будет печататься после номера «теоремы».

Переходим к примерам. Для этих примеров новые «теоремы» созданы в преамбуле документа, но для иллюстрации того, как это делается, показаны в правой колонке с исходным текстом непосредственно перед окружениями.

В первом примере мы создаём окружение `laws` (законы) и присваиваем ему заголовок «Закон». В каждой секции окружения `laws` будут нумероваться независимо друг от друга.

Закон 7.1 *Если существует два или более способа сделать нечто, и один из этих способов может привести к катастрофе, то кто-то обязательно это сделает.*

Закон 7.1 установил Мёрфи.

Во втором примере мы создаём окружение `theorems` (теоремы) и присваиваем ему заголовок «Теорема».

Теорема 1 (Fermat) *Нет целых чисел $n > 2$, x , y и z таких, что $x^n + y^n = z^n$.*

Теорема 2 *Любой линейный оператор A имеет единственный сопряженный.*

```
\newtheorem{laws}{Закон}[section]
\begin{laws}\label{Murphy}
Если существует два или более способа
сделать нечто, и один из этих способов
может привести к катастрофе, то кто-то
обязательно это сделает.
\end{laws}
Закон~\ref{Murphy} установил Мёрфи.
```

```
\newtheorem{theorems}{Теорема}
\begin{theorems}[Fermat]
Нет целых чисел  $n > 2$ ,  $x$ ,  $y$ 
и  $z$  таких, что  $x^n + y^n = z^n$ .
\end{theorems}

\begin{theorems}
Любой линейный оператор  $A$ 
имеет единственный сопряженный.
\end{theorems}
```

Пакет `theorem` из коллекции \LaTeX а `tools` позволяет поменять местами номер и текст заголовка теоремы, вынести заголовок в отдельную строку, вынести номер на левое поле страницы, задать по своему усмотрению шрифт для заголовка и формулировки теорем, задать вертикальный промежуток перед и после теоремы.

Пакет `amsthm` из коллекции \LaTeX а добавляет декларацию

`\newtheorem*{env}{type}`

для печати теорем без номера и окружение `proof` для печати доказательств.

8 Дополнительная нумерация уравнений

Окружение `subequations` из пакета `amsmath` позволяет организовать дополнительную нумерацию в группе логически связанных между собой уравнений. В результате, можно ссылаться как на всю группу сразу, так и на каждое уравнение из группы. В группу включаются *все* нумерованные уравнения из окружений $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ а (`equation`, `gather`, `multline` и т.д.), размещённых в данном окружении `subequations`. Команда `\label` внутри окружения `subequations`, но вне окружений `equation` и т.д., метит всю группу. Поясним сказанное на простом примере, в котором только уравнения из одного окружения `align` составляют группу.

$x = r \sin \theta \cos \phi$	(21a)	<code>\begin{subequations}\label{eq:xyz}</code>
$y = r \sin \theta \sin \phi$	(21b)	<code>\begin{align}</code>
$z = r \cos \theta$	(21c)	<code> x &= r \sin\theta \cos\phi \\</code>
		<code> y &= r \sin\theta \sin\phi \\</code>
		<code> z &= r \cos\theta \label{eq:z}</code>
		<code>\end{align}</code>

Используя (21c) ...

Преобразование (21) ...

```
Используя~\eqref{eq:z} \dots
\end{subequations}\par
Преобразование~\eqref{eq:xyz} \dots
```

Формат дополнительной нумерации можно задать самому следующим образом. Номер группы хранится в счётчике `parentequation`. Номер уравнения по-прежнему печатает команда `\theequation`, но она при входе в окружение `subequations` автоматически переопределяется следующим образом:

```
\renewcommand{\theequation}{\theparentequation\alpha{equation}}
```

Поэтому дополнительная нумерация идёт строчными латинскими буквами. Если мы хотим, чтобы дополнительная нумерация в группе печаталась русскими буквами, то надо сделать переопределение в этой группе, используя команду `\asbuk` стиля `russian` пакета `babel`:

```
\begin{subequations}
\renewcommand{\theequation}{\theparentequation\asbuk{equation}}
...
\end{subequations}
```

После этого будем иметь

$$x = r \sin \theta \cos \phi \tag{22a}$$

$$y = r \sin \theta \sin \phi \tag{22б}$$

$$z = r \cos \theta \tag{22в}$$

Если вместо формата `\alpha` использовать формат `\roman`, то дополнительная нумерация в группе будет печататься римскими цифрами `i`, `ii`, `iii` и т.д.

8.1 Нумерация уравнений вручную

Команда `\tag` из пакета `amsmath` позволяет заменить обычный номер у нумерованных уравнений, а также добавить к нумерованным формулам номер произвольного вида. Команду `\tag` можно разместить до или после уравнения, но перед командой `\\` в окружениях с многострочными уравнениями. После этого на месте обычного номера печатается в круглых скобках аргумент команды `\tag`, который обрабатывается в текстовой моде. В окружениях, дающих нумерованные формулы,

такие уравнения не меняют значение счётчика `equation`. Это позволяет не только давать нестандартные номера уравнениям, но и организовать дополнительную нумерацию вручную. Пример:

Уравнение

$$F_1 = -F_2 \quad (23)$$

можно записать в виде

$$F_1 + F_2 = 0 \quad (23')$$

Используя (23) или (23'),

Команда `\tag*` аналогична команде `\tag`, но она не ставит круглых скобок вокруг «номера» формулы.

Уравнение

```
\begin{equation}\label{eq:1}
  F_1 = -F_2
\end{equation}
```

можно записать в виде

```
\begin{equation}\label{eq:2}
  F_1 + F_2 = 0 \tag{\ref{eq:1}$'$}
\end{equation}
```

Используя `\eqref{eq:1}` или `\eqref{eq:2}`,

9 Разное

9.1 Коммутативные диаграммы

Пакет `amscd` из коллекции $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ а вводит окружение `CD`, в котором легко создавать простые диаграммы без диагональных стрелок. Приведём один пример.

$$\begin{array}{ccc}
 S^{W_\Lambda} \otimes T & \xrightarrow{j} & T \\
 \downarrow & & \downarrow \text{End } P \\
 (S \otimes T)/I & \xlongequal{\quad} & (Z \otimes T)/J
 \end{array}$$

```

\[\begin{CD}
S^{W_\Lambda} \otimes T @>j>> T \\
@VVV @VV\text{End } P\VV \\
(S \otimes T)/I @= (Z \otimes T)/J
\end{CD}\]
```

В командных скобках `CD` команды `@>>>`, `@<<<`, `@VVV` и `@AAA` дают, соответственно, правую стрелку, левую стрелку, стрелку вниз и стрелку вверх. Для горизонтальных стрелок, материал между первым и вторым символами `>` или `<` печатается в виде верхнего индекса, а материал между вторым и третьим символами — в виде нижнего индекса. Аналогично, материал между первым и вторым или вторым и третьим символами `A` или `V` вертикальных стрелок будет печататься как левый или правый «боковой индекс».

Более мощные средства для построения диаграмм содержат пакеты `hupic` и `pstricks`. Для `hupic` требуются специальные шрифты со стрелками (включены в дистрибутив пакета) или драйвер PostScript'a. Пакет `pstricks` работает только с PostScript'ом.

9.2 Формулы в рамке

Команда `\boxed` из пакета `amsmath` рисует рамку вокруг формулы:

$$\boxed{W_t - F \subseteq V(P_i) \subseteq W_t}$$

```

\[\boxed{W_t - F \subseteq V(P_i) \subseteq W_t}\]
```

9.3 Команды, пригодные для любой моды

Аргумент команды `\ensuremath` всегда обрабатывается как математическая формула, независимо от того, в текстовой или в математической моде она встречается. Приведём пример применения этой полезной команды.

Если в документе часто используются матрицы Паули, причём как поодиночке, так и в сложных формулах, то целесообразно для их набора создать новую команду в виде:

```
\newcommand{\s}[1]{\ensuremath{\hat{\sigma}}_{#1}}
```

Теперь матрицы Паули внутри текста можно набирать командой `\s`, не переключаясь в математическую моду:

Матрицы Паули $\hat{\sigma}_i \dots$. Имеем

Матрицы Паули `\s{i} \dots`. Имеем
`\[\s{x}\s{y} = i\s{z} \]`

$$\hat{\sigma}_x \hat{\sigma}_y = i \hat{\sigma}_z$$

9.4 Изменение размеров формулы

Формулу можно выделить крупным шрифтом, используя декларации выбора размера шрифта в текстовой моде:

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \quad (24)$$

```
{\Large
\begin{equation}
\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}
\end{equation} }
```

Видно, что номер формулы также печатается крупным шрифтом. Используя команду `\tag*` (см. стр. 39), его можно напечатать шрифтом нормального размера. В этом случае, правда, придётся вручную увеличивать на единицу значение счётчика `equation`:

$$\left(\sum_{k=1}^n x^k \right)^2 \quad (25)$$

```
\addtocounter{equation}{1}
{\Large
\begin{equation}
\Bigl( \sum_{k=1}^n x^k \Bigr)^2
\tag*{\normalsize (\theequation)}
\end{equation} }
```

Разделители, которые печатаются модифицированными в пакете `amsmath` командами `\big`, `\Big` и т.д., также изменяют свой размер.

9.5 Подбор размера разделителя вручную

Выбор размера разделителей \TeX ом может быть неудовлетворительным. Формула

$$\sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \left[1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \right],$$

набранная по всем правилам

```
\[ \sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \left[ 1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \right] \]
```

имеет слишком высоко простирающиеся квадратные скобки. Очевидно, что их надо «вручную» сначала уменьшить по высоте, а затем опустить:

$$\sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \left[1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \right] \quad (26)$$

Чтобы сократить набор формулы (26), была введена команда для печати скобок:

```
\newcommand{\ang}[1]{\raisebox{-5pt}{$\left#1\right.$}}
```

Видно, что высота скобки в конструкции «разделитель без пары» задана линейкой `\rule` высотой 22pt, а опущена скобка на 5 pt командой `\raisebox`. Исходный текст формулы (26):

```
\[ \sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \ang{[}
1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \ang{]} \]
```

Алфавитный указатель

Символы			
\!	36	\arg	27
&	7–9, 29	array	29
'	12	\arraycolsep	30
\(4	\arrayrulewidth	30
(21	\arraystretch	30
\)	4	\Arrowvert	22
)	21	\arrowvert	22
*	4	\asbuk	39
+	13	\ast	13
\,	36	\asymp	14
,	16		
-	13	B	
/	21	\backepsilon	14
\:	36	\backprime	25
:	14	\backsim	14
\;	36	\backsimeq	14
;	16	\backslash	21
<	14	\bar	17
=	14	\barwedge	13
>	14	\Bbbk	25
\[6	\because	14
[21	\beta	15
\	7–10, 29	\betaup	16
*	10	\beth	25
\{	21	\between	14
\	21	\Big	22
\}	21	\big	22
\]	6	\bigcap	19
]	21	\bigcirc	13
^	12	\bigcup	19
—	12	\Bigg	22
	21	\bigg	22
		\Biggl	22
		\biggl	22
		\Biggm	22
		\biggm	22
		\Biggr	22
		\biggr	22
		\Bigl	22
		\bigl	22
		\Bigm	22
		\bigm	22
		\bigodot	19
		\bigoplus	19
		\bigotimes	19
		\Bigr	22
		\bigr	22
		\bigscup	19
		\bigstar	25
		\bigtriangledown	13
		\bigtriangleup	13
		\biguplus	19
		\bigvee	19
		\bigwedge	19
		\binom	26
		\binoppenalty	5
		\blacklozenge	25
		\blacksquare	25
		\blacktriangle	25
		\blacktriangledown	25
		\blacktriangleleft	14
		\blacktriangleright	14
		bmatrix	31
		\bmod	29
		\boldsymbol	35
		\bordematrix	31
		\bot	24
		\bowtie	14
		\Box	24
		\boxdot	13
		\boxed	40
		\boxminus	13
		\boxplus	13
		\boxtimes	13
		\bracevert	22
		\breve	17
		\bullet	13
		\Bumpeq	14
		\bumpeq	14
		C	
		\Cap	13
		\cap	13
		cases	32
		CD	40
		\cdot	13
		\cdotp	16
		\cdots	12
		\centerdot	13
		\cfrac	18
		\ch	28
		\check	17
		\checkmark	24
		\chi	15
		\chiup	16
		\circ	13
		\circeq	14
		\circlearrowleft	24
		\circlearrowright	24
		\circledast	13
		\circledcirc	13
		\circleddash	13
		\circledR	24
		\circledS	25
		\clubsuit	24
		\colon	16
		\complement	25
		\cong	14
		\coprod	19
		\copyright	24
		\cos	27
		\csc	28
		\cosh	27
		\cot	27

<code>\lambdaup</code>	16	<code>\Longrightarrow</code>	23	N	
<code>\langle</code>	21	<code>\longrightarrow</code>	23	<code>\nabla</code>	24
<code>\lbrace</code>	21	<code>\looparrowleft</code>	24	<code>\natural</code>	24
<code>\lbrack</code>	21	<code>\looparrowright</code>	24	<code>\ncong</code>	15
<code>\lceil</code>	21	<code>\lozenge</code>	25	<code>\ne</code>	14
<code>\ldotp</code>	16	<code>\lrcorner</code>	22	<code>\nearrow</code>	23
<code>\ldots</code>	12	<code>\Lsh</code>	24	<code>\neg</code>	24
<code>\le</code>	14	<code>\ltimes</code>	13	<code>\negmedspace</code>	36
<code>\leadsto</code>	23	<code>\lVert</code>	21	<code>\negthickspace</code>	36
<code>\left</code>	21	<code>\lvert</code>	21	<code>\negthinspace</code>	36
<code>\Leftarrow</code>	23	<code>\lvertneqq</code>	15	<code>\neq</code>	14
<code>\leftarrow</code>	23	M			
<code>\leftarrowtail</code>	24	<code>\maltese</code>	24	<code>\newtheorem</code>	38
<code>\lefteqn</code>	37	<code>\mapsto</code>	23	<code>\nexists</code>	25
<code>\leftharpoonup</code>	23	<code>math</code>	4	<code>\ngeq</code>	15
<code>\leftharpoonup</code>	23	<code>\mathalpha</code>	11	<code>\ngeqq</code>	15
<code>\leftleftarrows</code>	24	<code>\mathbb</code>	34	<code>\ngeqslant</code>	15
<code>\Leftrightarrow</code>	23	<code>\mathbf</code>	33	<code>\ngtr</code>	15
<code>\leftrightarrows</code>	24	<code>\mathbin</code>	11	<code>\ni</code>	14
<code>\leftrightharpoons</code>	24	<code>\mathcal</code>	33	<code>\nLeftarrow</code>	24
<code>\leftrightsquigarrow</code>	24	<code>\mathclose</code>	11	<code>\nleftarrow</code>	24
<code>\leftroot</code>	17	<code>\mathfrak</code>	34	<code>\nLeftrightarrow</code>	24
<code>\leftthreetimes</code>	13	<code>\mathindent</code>	7	<code>\nleftrightarrow</code>	24
<code>\leq</code>	14	<code>\mathit</code>	33	<code>\nleq</code>	15
<code>\leqq</code>	14	<code>\mathnormal</code>	33	<code>\nleqq</code>	15
<code>\leqslant</code>	14	<code>\mathop</code>	11	<code>\nleqslant</code>	15
<code>\lessapprox</code>	14	<code>\mathopen</code>	11	<code>\nless</code>	15
<code>\lessdot</code>	14	<code>\mathord</code>	11	<code>\nmid</code>	15
<code>\lesseqgtr</code>	14	<code>\mathpunct</code>	11	<code>\nobreakdash</code>	37
<code>\lesseqgtr</code>	14	<code>\mathrel</code>	11	<code>\NOD</code>	28
<code>\lessgtr</code>	14	<code>\mathrm</code>	33	<code>\nod</code>	28
<code>\lesssim</code>	14	<code>\mathscr</code>	34	<code>\NOK</code>	28
<code>\lfloor</code>	21	<code>\mathsf</code>	33	<code>\nok</code>	28
<code>\lg</code>	27	<code>\mathstrut</code>	17, 37	<code>\nolimits</code>	19
<code>\lgroup</code>	22	<code>\mathsurround</code>	4	<code>\nonumber</code>	7
<code>\lhd</code>	13	<code>\mathtt</code>	33	<code>\not</code>	14
<code>\lim</code>	28	<code>\mathversion</code>	35	<code>\notag</code>	8
<code>\liminf</code>	28	<code>matrix</code>	31	<code>\notin</code>	14
<code>\limits</code>	20	<code>\max</code>	28	<code>\nparallel</code>	15
<code>\limsup</code>	28	<code>\mbox</code>	32	<code>\nprec</code>	15
<code>\ll</code>	14	<code>\measuredangle</code>	25	<code>\npreceq</code>	15
<code>\llcorner</code>	22	<code>\medspace</code>	36	<code>\nrightarrow</code>	24
<code>\Lleftarrow</code>	24	<code>\mho</code>	24, 25	<code>\nrightarrow</code>	24
<code>\lll</code>	14	<code>\mid</code>	14	<code>\nshortmid</code>	15
<code>\lless</code>	14	<code>\min</code>	28	<code>\nshortparallel</code>	15
<code>\lmoustache</code>	22	<code>\mkern</code>	36	<code>\nsim</code>	15
<code>\ln</code>	27	<code>\mod</code>	29	<code>\nsubseteq</code>	15
<code>\lnapprox</code>	15	<code>\models</code>	14	<code>\nsubseteqq</code>	15
<code>\lneq</code>	15	<code>\mp</code>	13	<code>\nsucc</code>	15
<code>\lneqq</code>	15	<code>\mspace</code>	36	<code>\nsucceq</code>	15
<code>\lnot</code>	24	<code>\mu</code>	15	<code>\nsupseteq</code>	15
<code>\lnsim</code>	15	<code>\multimap</code>	24	<code>\nsupseteqq</code>	15
<code>\log</code>	27	<code>multline</code>	9	<code>\ntriangleleft</code>	15
<code>\Longleftarrow</code>	23	<code>multline*</code>	9	<code>\ntrianglelefteq</code>	15
<code>\longleftarrow</code>	23	<code>\multlinegap</code>	9	<code>\ntriangleright</code>	15
<code>\Longleftrightarrow</code>	23	<code>\muup</code>	16	<code>\ntrianglerighteq</code>	15
<code>\longleftrightarrow</code>	23			<code>\nu</code>	15
<code>\longleftrightarrow</code>	23			<code>\nuup</code>	16
<code>\longmapsto</code>	23			<code>\nVDash</code>	15
				<code>\nVdash</code>	15

<code>\nvDash</code>	15	<code>\psiup</code>	16	<code>\sloppy</code>	5
<code>\nvdash</code>	15			<code>sloppypar</code>	5
<code>\nwarrow</code>	23			<code>\smallfrown</code>	14
		Q		<code>smallmatrix</code>	31
O		<code>\qqquad</code>	36	<code>\smallsetminus</code>	13
<code>o</code>	15	<code>\quad</code>	36	<code>\smallsmile</code>	14
<code>\odot</code>	13			<code>\smash</code>	37
<code>\oint</code>	19	R		<code>\smile</code>	14
<code>\Omega</code>	16	<code>\raisetag</code>	10	<code>\spadesuit</code>	24
<code>\omega</code>	15	<code>\rangle</code>	21	<code>\sphericalangle</code>	25
<code>\omegaaup</code>	16	<code>\rbrack</code>	21	<code>split</code>	9
<code>\ominus</code>	13	<code>\rceil</code>	21	<code>\sqcap</code>	13
<code>\oplus</code>	13	<code>\Re</code>	24	<code>\sqcup</code>	13
<code>\oslash</code>	13	<code>\ref</code>	6	<code>\sqrt</code>	17
<code>\otimes</code>	13	<code>\relpenalty</code>	5	<code>\sqsubset</code>	14
<code>\overbrace</code>	25	<code>\rfloor</code>	21	<code>\sqsubseteq</code>	14
<code>\overleftarrow</code>	26	<code>\rgroup</code>	22	<code>\sqsupseteq</code>	14
<code>\overleftrightarrow</code>	26	<code>\rhd</code>	13	<code>\square</code>	25
<code>\overline</code>	25	<code>\rho</code>	15	<code>\stackrel{\rel}{}</code>	26
<code>\overrightarrow</code>	26	<code>\rhoup</code>	16	<code>\star</code>	13
<code>\overset</code>	26	<code>\right</code>	21	<code>\strut</code>	25
<code>\owns</code>	14	<code>\Rightarrow</code>	23	<code>subarray</code>	20
		<code>\rightarrow</code>	23	<code>subequations</code>	39
		<code>\rightarrowtail</code>	24	<code>\Subset</code>	14
P		<code>\rightharpoondown</code>	23	<code>\subset</code>	14
<code>\P</code>	24	<code>\rightharpoonup</code>	23	<code>\subseteq</code>	14
<code>\parallel</code>	14	<code>\rightleftarrows</code>	24	<code>\subseteqeq</code>	14
<code>\partial</code>	24	<code>\rightleftharpoons</code>	23, 24	<code>\subseteqqq</code>	14
<code>\perp</code>	14	<code>\rightrightarrows</code>	24	<code>\subsetneq</code>	15
<code>\Phi</code>	16	<code>\rightsquigarrow</code>	24	<code>\subsetneqq</code>	15
<code>\phi</code>	15	<code>\rightthreetimes</code>	13	<code>\substack</code>	20
<code>\phiup</code>	16	<code>\risingdotseq</code>	14	<code>\succ</code>	14
<code>\Pi</code>	16	<code>\rmoustache</code>	22	<code>\succapprox</code>	14
<code>\pi</code>	15	<code>\Rrightarrow</code>	24	<code>\succcurlyeq</code>	14
<code>\pitchfork</code>	14	<code>\Rsh</code>	24	<code>\succeq</code>	14
<code>\piup</code>	16	<code>\rtimes</code>	13	<code>\succnapprox</code>	15
<code>\pm</code>	13	<code>\rVert</code>	21	<code>\succneqq</code>	15
<code>pmatrix</code>	31	<code>\rvert</code>	21	<code>\succnsim</code>	15
<code>\pmb</code>	35			<code>\succsim</code>	14
<code>\pmod</code>	29	S		<code>\sum</code>	19
<code>\pod</code>	29	<code>\S</code>	24	<code>\sup</code>	28
<code>\pounds</code>	24	<code>\scriptscriptstyle</code>	11	<code>\Supset</code>	14
<code>\Pr</code>	28	<code>\scriptstyle</code>	11	<code>\supset</code>	14
<code>\prec</code>	14	<code>\searrow</code>	23	<code>\supseteq</code>	14
<code>\precapprox</code>	14	<code>\sec</code>	27	<code>\supseteqqq</code>	14
<code>\preccurlyeq</code>	14	<code>\setminus</code>	13	<code>\supsetneq</code>	15
<code>\preceq</code>	14	<code>\sh</code>	28	<code>\supsetneqq</code>	15
<code>\precnapprox</code>	15	<code>\sharp</code>	24	<code>\surd</code>	24
<code>\precneqq</code>	15	<code>\shortmid</code>	14	<code>\swarrow</code>	23
<code>\precnsim</code>	15	<code>\shortparallel</code>	14		
<code>\precsim</code>	14	<code>\shoveleft</code>	9	T	
<code>\prime</code>	24	<code>\shoveright</code>	9	<code>\tag</code>	39
<code>\Prob</code>	28	<code>\sideset</code>	20	<code>\tag*</code>	40
<code>\prod</code>	19	<code>\Sigma</code>	16	<code>\tan</code>	27
<code>\Proj</code>	28	<code>\sigma</code>	15	<code>\tanh</code>	27
<code>\projlim</code>	28	<code>\sigmaaup</code>	16	<code>\tau</code>	15
<code>proof</code>	39	<code>\sim</code>	14	<code>\tauup</code>	16
<code>\propto</code>	14	<code>\simeq</code>	14	<code>\tbinom</code>	26
<code>\Psi</code>	16	<code>\sin</code>	27	<code>\text</code>	33
<code>\psi</code>	15	<code>\sinh</code>	27		

