

Отчет по изучению L^AT_EX

Афанасьев Даниил

11 апреля 2025 г.

Содержание

1	Для чего Л^AT_EX	3
2	Создание документа	5
3	Набор математических формул	16
3.1	Греческий алфавит	18

1 Для чего \LaTeX

\LaTeX — это система компьютерной вёрстки, которая широко используется для создания научных, технических и академических документов. Она особенно популярна в математике, физике, информатике и других науках, где требуется высококачественное оформление формул, таблиц и графиков. Основные преимущества \LaTeX включают:

- **Высокое качество типографики:** \LaTeX автоматически управляет шрифтами, отступами, межстрочными интервалами и другими аспектами вёрстки, что позволяет создавать профессионально выглядящие документы.
- **Удобство работы с математическими формулами:** \LaTeX предоставляет мощные инструменты для набора сложных математических выражений.
- **Кроссплатформенность:** Документы, созданные в \LaTeX , могут быть скомпилированы на любой операционной системе (Windows, macOS, Linux).
- **Разделение содержания и оформления:** Автор может сосредоточиться на содержании, а \LaTeX позаботится о внешнем виде документа.
- **Поддержка больших документов:** \LaTeX идеально подходит для создания книг, диссертаций и других объёмных работ.

В настоящее время нет нужды доказывать преимущества издательской системы \LaTeX в оформлении статей и книг научно-технической и особенно физико-математической тематики. Она прочно заняла ведущие позиции во всех издательствах, специализирующихся на выпуске научной литературы

и периодики, так как сводит к минимуму редакционную обработку рукописей и существенно сокращает сроки их опубликования. В её основе лежит концепция программной верстки, заложенная Дональдом Е. Кнудом, создавшим в 1978 г. язык программирования и компилятор T_EX, предназначенные для верстки текста, насыщенного математическими формулами. Создавая документ, автор пишет программу, команды которой указывают, что нужно сделать с той или иной частью текста. Выполняя данную программу, компилятор T_EX верстает документ.

В 1984 г. Лесли Лэмпорт разработал на основе T_EX удобный и компактный макроязык L^AT_EX, кардинально минимизировавший количество команд разметки, необходимых для форматирования рукописи. Он обогатил концепцию программной верстки понятием стиля, или класса документа. Размечая текст, автор использует достаточно небольшой набор команд, имеющих различные настройки. Настройки помещаются в стилевой файл, определяющий вид сверстанного документа. Изменение стиля верстки не требует изменения текста программы, за исключением команды, загружающей нужный стиль.

Введенные дополнения оказались очень эффективными. К началу 90-х гг. ведущие издательства научной литературы разработали собственные стили, отвечающие их полиграфическим традициям, и ввели L^AT_EX в издательский процесс. Появилось большое количество пакетов, т.е. дополнительного программного обеспечения, существенно расширивших возможности L^AT_EX. С годами их число неуклонно растет. Так как T_EX и L^AT_EX развиваются и распространяются усилиями сообщества свободного программирования, любой пользователь имеет право разработать новый пакет, решающий какую-то задачу верстки, зарегистрировать его и присовокупить к дистрибутиву L^AT_EX, насчитывающему в настоящее время более трех тысяч пакетов. Как правило, такие пакеты не конфликтуют с ядром L^AT_EX, однако могут конфликтовать между собой. С этой точки зрения

все программное обеспечение следует разделить на стандартный \LaTeX , состоящий из ядра и небольшого числа пакетов, поддерживаемых и развиваемых основной командой разработчиков, и нестандартные пакеты, разработанные и поддерживаемые многочисленными энтузиастами. В практике выпуска научной периодики издательства используют стандартный \LaTeX , остальные ресурсы могут отсекаются на стадии подачи рукописи в редакцию, поэтому нестандартные пакеты можно использовать лишь на свой страх и риск в документах, не предназначенных для публикации. Начиная с 1993 г. стандартом является версия $\text{\LaTeX}2$, ставшая за прошедшие годы очень стабильной. Много лет ведется разработка $\text{\LaTeX}3$, направленная на полную совместимость с форматом unicode, гибкую работу со шрифтами, интеграцию ряда пакетов непосредственно в ядро \LaTeX и расширение возможностей компилятора за счет использования скриптов языка Lua. Одна из основных задач проекта — расширение возможности основных команд за счет введения дополнительных настроек без изменения их интерфейса, чтобы при переходе к новому стандарту сохранить возможность использовать ранее созданные документы. Многие наработки незаметно для пользователей уже внесены в $\text{\LaTeX}2$.

2 Создание документа

Для работы в \LaTeX необходимы компиляторы, набор программных средств — пакетов в терминологии \LaTeX , и графическая оболочка, обеспечивающая комфортное редактирование рукописи, а также управление компиляторами и средства просмотра готовых документов. Вдобавок к этому для формирования списков литературы желательно иметь средства работы с библиографическими базами.

В настоящее время во всех операционных системах имеется одинаковый набор средств работы в \LaTeX . Освоив их, автор может

использовать любую ОС, не испытывая неудобства работы с незнакомым программным обеспечением.

Рукопись можно редактировать в обычном текстовом редакторе, но намного удобней пользоваться специализированными программами имеющимися во всех ОС. Лучшей графической оболочкой является программа TEXstudio, имеющая собственные средства просмотра pdf-документов. Ресурсы L^AT_EX содержит дистрибутив MiKTeX, устанавливающий базовый набор компиляторов и программных средств, необходимых для верстки. Дистрибутив TEXLive, развивающийся ОС Linux, содержит самые свежие разработки компиляторов и программны средств. Удобный интерфейс работы с библиографическими базами предоставляет написанная на языке Java программа JabRef.

Все перечисленные программы распространяются и используются свободно. Они активно развиваются и потому описывать их интерфейс — задач неблагодарная, если не бессмысленная.

Установка и настройка программ

MiKTeX можно установить непосредственно из интернета в полном или базовом варианте. Полный дистрибутив занимает несколько гигабайтов, при этом для работы нужна лишь малая его часть. Базовый комплект содержит только основные программные средства, а остальные добавляются по мере надобности. По умолчанию недостающие компоненты загружаются из интернета, что не всегда удобно, так как их получение зависит от доступности сети. В то же время набор необходимых средств формируется довольно быстро, и после непродолжительной работы острая потребность в интернете пропадает.

Стандартным путем является загрузка с сайта <http://MikTeX.org/> и запуск программы basic-MiKTeX-•-x#.exe, устанавливающей базовый

комплект MiKTeX. Здесь • обозначает текущую версию MiKTeX, а # — разрядность компьютера. Запустите программу и выберите вариант установки «Install MikTeX only for me». В этом случае дистрибутив установится в папки, к которым вы будете иметь полный доступ, а для добавления пакетов не потребуются права администратора. После установки запустите программу miktex-console, войдите в меню «Updates» и обновите установленные программы и пакеты. Выполнив данные действия вы получите базовый комплект MiKTeX, который по мере необходимости будет автоматически устанавливать недостающие пакеты из интернета. Его следует подстроить, добавив шрифты для генерации PostScript- и pdf документов. Для этого вновь запустите программу MikTeX Console, откройте меню «Packages», в строке поиска введите название пакета cm-super, нажмите правой клавишей мыши на появившуюся строку с его описанием и в выпавшем меню выберите «Install package». После этого MiKTeX полностью готов к работе.

Вдобавок к MiKTeX нужно установить программу TeXstudio, загрузив ее с сайта <https://texstudio.org>. В ходе установки она автоматически настроит взаимодействие с MiKTeX, поэтому ее следует устанавливать вслед за ним. Установив ее, загрузите русский словарь, используемый для проверки орфографии набираемого текста. Для этого запустите TeXstudio и в меню «Options \Rightarrow Configure TeXstudio» выберите «Language Checking», открыв ссылку «LibreOffice», найдите и скачайте последнюю версию русского словаря, а затем, нажав кнопку «ImportDicionary...», установите его. Если вы преимущественно работаете с русскими текстами, имеет смысл загружать его при запуске TeXstudio. Это можно сделать, выбрав его в меню «Default Language».

Установив MiKTeX и TeXstudio, вы получите полный набор средств для работы с L^AT_EX.

Загрузка и сохранение файлов

TEXstudio, по умолчанию, настроен на работу с файлами в кодировке UTF-8, которую без особой нужды лучше не менять, так как TEXstudio использует ее, редактируя рукопись. Файлы, имеющие другие кодировки, при загрузке перекодируются в UTF-8, а при сохранении возвращаются в исходную. На обоих этапах могут возникнуть неприятности.

Если загруженный текст превращается в абракадабру, измените кодировку загрузки и повторите загрузку. Для этого в меню кодировок в нижней строке окна редактора или меню «Edit \Rightarrow Setup Encoding» выберите нужную кодировку² и нажмите клавишу «Reload With».

После того как текст станет читаемым, снова откройте меню, выберите кодировку UTF-8 и нажмите клавишу «Change To», чтобы файл сохранялся в кодировке UTF-8. Если это не сделать, кодировка загрузки будет использована и при сохранении.

Если абракадабру содержит сверстанный документ, проверьте кодировку текста, объявленную в команде

`\usepackage[кодировка]inputenc`

Компиляция и просмотр документа

Организуя процесс компиляции, TEXstudio записывает рукопись в файл и указывает компилятору путь к нему. Таким образом, внесенные в текст изменения автоматически сохраняются перед компиляцией.

В подготовке документа участвует несколько программ, каждая из которых делает свою часть работы. При работе они генерируют ряд файлов и посредством них обмениваются информацией. Для успешного выполнения своей функции каждая из них нуждается в данных, подготовленных другими программами. Чтобы сверстать документ в окончательном виде, зачастую нужно последовательно запустить несколько

программ в строго определенном порядке. TEXstudio формирует цепочку компиляции, запускает ее и следит за сообщениями компиляторов во время их работы.

Основным компилятором является программа `pdftex`. Для ее запуска используется программа `pdflatex`, передающая ей требуемые настройки. При компиляции в качестве внешних ресурсов помимо текста рукописи и рисунков загружаются следующие файлы: форматный файл L^AT_EX (`•.fmt`), класс документа (`•.cls`) и пакеты (`•.sty`) и (`•.tex`). В скобках указаны расширения имен файлов. Кроме этого загружаются файлы, созданные в ходе предшествующих компиляций: оглавление (`•.toc`), предметный указатель (`•.ind`), библиография (`•.bbl`) и информация, необходимая для нумерования разделов, формул, цитируемых источников и т.д. (`•.aux`).

В процессе работы компилятор TEX создает документ (`•.pdf` или `•.dvi`) и генерирует ряд служебных файлов, содержащих сведения для предметного указателя (`•.idx`), оглавление (`•.toc`) и отчет о компиляции (`•.log`). В файл с расширением `•.aux` заносится информация о порядке следования в тексте рукописи рисунков, таблиц, цитируемых источников и других объектов.

Если документ сохраняется в формате `dvi`, программа `dvips` конвертирует его в PostScript.

Предметный указатель составляют программы `MakeIndex`, `xindy` или `xindex`. Из файла с расширением `•.idx` они считывают список терминов, формируют указатель и сохраняют его в файле с расширением `•.ind`, записав отчет о компиляции в файл с расширением `•.blg`.

Программа `bibtex` создает библиографию. Из файла с расширением `•.aux` она считывает список цитируемых источников, загружает библиографические базы (`•.bib`) и стиль оформления библиографии (`•.bst`), формирует список литературы, сохраняет его в файл с расширением `•.bbl` и направляет в файл с расширением `•.blg` отчет о компиляции.

Таким образом, помимо самого документа компиляторы генерируют значительное число дополнительных файлов, поэтому для каждого нового документа рекомендуется создать собственную рабочую папку. Файлы, генерируемые в процессе компиляции, наследуют имя файла с текстом рукописи, которое помещается в команду `\jobname` , используемую для генерации других имен, например `\jobname.pdf`.

Файлы с расширениями `•.aux`, `•.toc`, `•.lof`, `•.lot`, `•.bbl` и `•.ind` используются в последующих компиляциях. При смене класса документа или изменении набора загружаемых пакетов они являются потенциальным источником ошибок, так как содержащиеся в них команды могут стать неопределенными, или начнут конфликтовать с другими командами. В таком случае файлы с данными расширениями следует удалить и начать компиляцию «с чистого листа». В `TEXstudio` для этого предусмотрено меню «Tools \Rightarrow Clean Auxiliary Files». Меню «Tools» содержит также средства запуска различных комбинаций компиляции. На панели задач имеются иконка , запускающая компилятор `LATEX`, и иконка , инициирующая цепочку компиляций, формирующую все компоненты документа, включая библиографию и предметный указатель.

`TEXstudio` имеет окно просмотра pdf-документов, расположенное справа от окна редактора. После компиляции в нем показывается страница, содержащая абзац, в котором находится курсор окна редактора.

Окна редактирования и просмотра между собой связаны. Наведя курсор и нажав клавишу «Control» вместе с левой клавишей мыши, можно переключиться в соответствующее место рукописи или документа. Тот же переход можно совершить, нажав правую клавишу мыши и выбрав «Go to Source» или «Go to PDF» в появляющемся меню.

В ОС Windows для просмотра компилируемых документов не следует пользоваться программой `AdobeReader` . Открывая файл, она блокирует его изменение, что вызывает сбой работы компиляторов, которые не могут

сохранить сверстанный документ.

Основные понятия и синтаксис L^AT_EX

Программа L^AT_EX состоит из текста, команд и комментариев. Строго говоря, имеются еще и формулы, которые также состоят из текста, т.е. символов, обозначающих переменные, команд и комментариев и потому, с точки зрения синтаксиса, не являются особыми объектами.

Текст разбивается на слова, разделяемые пробелами, и абзацы, разделяемые пустыми строками. Между словами можно поставить любое количество пробелов, а между абзацами любое количество пустых строк, при компиляции они игнорируются. В непустых строках символы перевода строки считаются обычными пробелами.

Верстая документ, компилятор сам разбивает текст на строки и устанавливает промежутки между словами и строками, исходя из заданного стиля верстки. Дополнительные пробелы следует использовать для наглядного представления структуры рукописи и конструкций L^AT_EX. Сравните, например, два варианта кода, один из которых не очень удобен для восприятия:

```
1      $$\delta_{ij}=\begin{cases}1,&i=j\\0,&i\neq j\end{cases}$$,
```

а второй — прост и понятен:

```
1      $$ \delta_{ij} =  
2      \begin{cases}  
3          1, & i=j, \\  
4          0, & i\neq j.  
5      \end{cases} $$
```

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

При компиляции они дают одинаковый результат. Комментарий открывается символом процента % и ограничивается концом строки. В комментариях, состоящих из нескольких строк, каждая из них должна начинаться символом процента. Текст и команды, находящиеся в комментарии, при компиляции полностью игнорируются. Комментарии могут находиться в любом месте рукописи, в том числе в аргументах и параметрах команд. Комментарий не является пробелом и потому не разрывает текст, но попытка разбить комментарием имя команды вызовет ошибку.

Команды и окружения

Признаком команды является обратный слэш, за которым следует ее имя, параметры и аргументы, например

$$\backslash\text{samplecommand}[\text{параметр}]\{\text{аргумент}\}.$$

Имя команды состоит из латинских букв и звездочки *, которая может стоять в конце него. Аргументы команд заключаются в фигурные скобки, а параметры в квадратные. Прописные и строчные буквы в именах различаются, поэтому, например, команды $\backslash AA \rightsquigarrow \mathring{A}$ и $\backslash aa \rightsquigarrow \mathring{a}$ производят разные символы. \LaTeX не имеет какого-либо специального признака окончания имени команды. Им служит любой символ, отличный от букв и звездочки. После имени с равным успехом могут стоять как пробел, так и скобки, знаки препинания цифры и т. д., в любом случае компилятор распознает команду. Строение команд не ограничено каким-либо шаблоном. Наличие и количество аргументов и параметров вводится при их определении. \LaTeX не регламентирует порядок следования аргументов и

параметров, однако есть общие правила, которым они подчиняются. Во-первых, аргументы и параметры всегда следуют за именем команды. Во-вторых, они могут содержать другие команды, пробелы, комментарии и т.д. И наконец, между именем, аргументами и параметрами допускаются пробелы и комментарии, но не должно быть пустых строк. Аргумент содержит информацию, без которой команда не может быть выполнена, или объекты, с которыми она должна совершить какое-то действие. Если аргумент не выделен явно фигурными скобками, вместо него используется символ (или команда), следующий непосредственно за командой, например, $\textbf{AA} \rightsquigarrow \textbf{A}$. Если полученный таким образом аргумент не удовлетворяет требованиям команды, компилятор выдаст сообщение об ошибке. В частности, аргументом команды `\hspace{длина}` служит длина, характеризующаяся величиной и размерностью. Ее использование `\hspace 1cm` некорректно, так как число 1, получаемое в качестве аргумента, размерности не имеет и потому длиной не является. Описанный алгоритм действий компилятора называется правилом наследования аргумента. Параметр служит для настройки каких-либо аспектов выполнения команды. В определении команды ему присваивается определенное значение. Если при выполнении команды оно не требует корректировки, параметр можно опустить вместе с квадратными скобками. В качестве примера использования параметров и аргументов приведем команду: `\rule[смещение]{длина}{толщина}`, рисующую линию заданной длины и толщины и смещенную при необходимости относительно базисной линии строки. Например, команда `\rule[5pt]{25pt}{1pt}` рисует смещенную вверх горизонтальную линию, команда `\rule[-5pt]{1pt}{10pt}` — смещенную вниз вертикальную линию, а команда `\rule{6pt}{6pt}` — несмещенный квадратик. Для часто выполняемых действий используются односимвольные команды, или служебные символы, упрощенный синтаксис которых является исключением из правил. Они собраны в первой

колонке таблицы, во второй описано их назначение, а в последней приведены команды, используемые, чтобы напечатать сами символы. Часть из них также имеют нестандартный синтаксис. Отметим, что команды нижнего и верхнего индекса имеют аргумент, подчиняющийся приведенному выше правилу наследования. Сравните, $\$x_2^j\$ \rightsquigarrow x_i^2 j$ и $\$x_{ij}^2\$ \rightsquigarrow x_{ij}^2$. Большое число символов, используемых в тексте и формулах, заданы командами, например $\backslash dag \rightsquigarrow \dagger$. Потребность в них сокращается по мере расширения использования в \LaTeX кодировки Unicode, но пока еще они являются весомым компонентом формул и порой необходимы в тексте. Следует учитывать, что компилятор автоматически уберет пробел, следующий за именем команды, поэтому символ, заданный командой и отделенный от последующего текста лишь одним пробелом, сольется с ним: $\backslash d dag \text{ пример} \rightsquigarrow \ddagger \text{пример}$. Чтобы избежать этого, команду-символ можно заключить в фигурные скобки — $\{\backslash d dag\} \dots$, поставить пару скобок после нее — $\backslash d dag\{\} \dots$, или задать пробел явно командой « \gg » — $\backslash d dag \text{ пример} \rightsquigarrow \backslash d dag \text{ пример}$. Одной из основных конструкций \LaTeX является окружение:

```

\begin{имя окружения} аргументы и параметры
область действия окружения
\end{имя окружения}
  
```

Оно формируется командами $\backslash begin$ и $\backslash end$, охватывающими часть рукописи, являющуюся областью действия окружения. Их аргумент — имя окружения — определяет операцию, производимую над областью действия. Для имен окружений справедливы требования, предъявляемые к именам команд. Открывающая команда $\backslash begin$ может иметь дополнительные аргументы и параметры, а у закрывающей команды $\backslash end$ они отсутствуют, так как ее задача лишь ограничить область действия производимой операции. Окружения позволяют более наглядно представить структуру рукописи и в этом их основное назначение. Некоторые команды имеют

окружения-аналоги, например команда `\itshape\курсив` и окружение

`\begin{itshape}`

Для верстки большого

объема текста лучше

пользоваться окружениями.

`\end{itshape}`

Обычно команды используются для обработки небольших фрагментов текста в пределах строки или абзаца, тогда как окружения более удобны при верстке нескольких абзацев. Области действия команд тесно связаны с группами — одним из базовых понятий \TeX . Прежде чем приступить к верстке, компилятор считывает множество определений команд и значений переменных, задающих их параметры. Лишь малая часть переменных является глобальными и еще меньшая — статическими, т.е. сохраняющими неизменные значения во время компиляции. Подавляющая же часть представляет собой локальные динамические переменные, значение которых определено только внутри группы. Группой является аргумент и параметр команды, область действия окружения, математическая формула, ячейка таблицы или просто часть текста, заключенная в фигурные скобки `{...}` или между командами `\begingroup` и `\endgroup`. С точки зрения компилятора, сам текст рукописи также представляет собой группу, так как заключен в окружение `document`. Группы вкладываются друг в друга: рукопись содержит окружения и команды, в аргументах и параметрах которых находятся другие окружения и команды со своими аргументами и параметрами, и т.д. Уровень вложенности может быть очень велик. При выходе из вложенной группы измененные локальные переменные восстанавливают свои значения, а вновь введенные переменные и команды становятся неопределенными. Исходя из сказанного, команды, обрабатывающие текст, можно условно разделить на два типа. Область действия одних ограничена их собственным аргументом. Они создают

группу (аргумент) и внутри нее меняют динамическую переменную, например параметр шрифта, как в этом случае: `\textbf{при}мер` \rightsquigarrow **пример**. К данному типу команд относятся и окружения, область действия которых также является группой. Команды другого типа, называемые декларациями, действуют в пределах группы, в которой находятся, начиная с позиции, в которой они стоят. Декларации не действуют на предшествующий текст, но изменяя значение какой-либо динамической переменной, меняют режим верстки следующего за ними текста, вплоть до выхода из группы. Поясним сказанное примером, в котором группы формируются фигурными скобками, а декларации `\it`, `\sf`, `\large` и `\small` изменяют динамические переменные, управляющие начертанием и размером шрифта. Совместив код с результатом его выполнения, выделим группирующие скобки и команды мелким шрифтом и подчеркиванием.

3 Набор математических формул

L^AT_EX был создан с глубоким пониманием потребностей научного сообщества. Его математический движок - это не просто инструмент для набора символов, а целая система, учитывающая тонкости типографских традиций математической печати. Разработчики T_EX и L^AT_EX заложили в него принципы, которые делают математические формулы не только корректными, но и эстетически совершенными.

Сравнение математических режимов

В L^AT_EX существует принципиальное разделение между:

Встроенными (inline) формулами

Эти формулы интегрированы непосредственно в текстовый поток и имеют следующие особенности:

- Используют компактное вертикальное расположение элементов
- Автоматически активируют специальный математический шрифт
- Могут быть оформлены:

Классически: `\(...\)` (рекомендуется)

Альтернативно: `$...$` (исторически сложившийся вариант)

Специально: `\begin{math}...\end{math}`

Пример:

`\(\frac{dy}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} \)`

$\frac{dy}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x}$

Выключными (`display`) формулами

Эти формулы занимают отдельную строку и имеют:

- Увеличенные математические операторы
- Центрированное расположение
- Нумерацию (при необходимости)
- Варианты оформления:

Базовый: `\[...\]` (предпочтительный)

Устаревший: `$$...$$` (может вызывать проблемы с вертикальными пробелами)

Структурный: `\begin{displaymath}...\end{displaymath}`

Пример:

`\[i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r},t) = \hat{H} \Psi(\mathbf{r},t) \]`

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = \hat{H} \Psi(\mathbf{r}, t)$$

3.1 Греческий алфавит

В \LaTeX реализована полная поддержка греческого алфавита с учетом регистра:

Специальные математические символы

Полная классификация:

Бинарные операторы

Отношения

Стрелки

Углубленный анализ индексов и верхних/нижних элементов

Многоуровневые индексы \LaTeX позволяет создавать сложные иерархии индексов:

$\text{\textbackslash}[(x^{\hat{a}\hat{b}})^{\hat{c}_d} \quad \text{\textbackslash}left(\sum_{i=1}^{\infty}\text{\textbackslash}\infty\right)_{j=1}^{\hat{n}} \text{\textbackslash}]$

$$(x^{a^b})^{c_d} \left(\sum_{i=1}^{\infty} \right)_{j=1}^n$$

Позиционирование индексов для операторов Специальные команды управления положением индексов: $\text{\textbackslash}[\quad \text{\textbackslash}sum\text{\textbackslash}limits_{i=1}^{\hat{n}} \quad \text{\textbackslash}quad \text{\textbackslash}prod\text{\textbackslash}nolimits_{j\in J} \text{\textbackslash}]$

$$\sum_{i=1}^n \prod_{j \in J}$$

Нестандартные индексы Использование $\text{\textbackslash}substack$ для многострочных индексов: $\text{\textbackslash}[\quad \text{\textbackslash}sum_{\text{\textbackslash}substack{0 < i < m \\ 0 < j < n}} P(i,j) \text{\textbackslash}]$

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P(i, j)$$

Продвинутое форматирование дробей и корней

Альтернативные формы дробей `\[\tfrac{a}{b} \quad \dfrac{a}{b}`
`\quad \cfrac{1}{1+\cfrac{2}{1+\cfrac{3}{1+x}}}` `\]`

$$\frac{\frac{a}{b}}{\frac{a}{b} \quad \frac{1}{1 + \frac{2}{1 + \frac{3}{1 + x}}}}$$

Тонкая настройка корней `\[\sqrt{\mathstrut a} \quad \sqrt[\beta]{x} \quad \sqrt{\vphantom{\beta}x}` `\]`

$$\sqrt{a} \quad \sqrt[\beta]{x} \quad \sqrt{x}$$

Продвинутые математические конструкции

Многострочные уравнения

$$x = y + z \tag{1}$$

$$a = b + c \tag{2}$$

Матрицы всех видов

$$\begin{matrix} a & b \\ c & d \end{matrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \left| \begin{matrix} a & b \\ c & d \end{matrix} \right| \left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\| \tag{3}$$

Перенос длинных формул

`\beginmultline`
`a + b + c + d + e + f + g + h + i = \\\`
`j + k + l + m + n + o + p + q + r + s`
`\end{multline}`

Дополнительные пакеты для профессиональной математики

Основные математические пакеты

`\usepackage{amsmath, amssymb, amsthm, amsfonts, mathtools, bm}`

Специализированные пакеты

`\usepackage{physics}` - для физических обозначений

`\usepackage{siunitx}` - для единиц измерения

`\usepackage{tensor}` - для тензорной нотации

`\usepackage{braket}` - для квантовой механики