

## Раздел 1

### 1. Криволинейные трехмерные координаты. Связь между декартовыми, сферическими и цилиндрическими координатами.

Криволинейные координаты - это система координат для евклидова пространства, в которой координатные линии могут быть искривлены. Эти координаты могут быть получены из набора декартовых координат с помощью преобразования, которое локально обратимо (взаимно-однозначное отображение) в каждой точке. Это означает, что можно преобразовать точку, заданную в декартовой системе координат, в ее криволинейные координаты и обратно.

Хорошо известными примерами криволинейных систем координат в трехмерном евклидовом пространстве ( $R^3$ ) являются цилиндрические и сферические полярные координаты.

Декартова координатная поверхность в этом пространстве является координатной плоскостью; например,  $z = 0$  определяет плоскость  $x$ - $y$ . В том же пространстве координатная поверхность  $r = 1$  в сферических полярных координатах является поверхностью единичной сферы, которая изогнута. Формализм криволинейных координат обеспечивает единое и общее описание стандартных систем координат.

Криволинейная система координат может быть проще в использовании, чем декартова система координат для некоторых приложений. Движение частиц под действием центральных сил обычно легче решить в сферических полярных координатах, чем в декартовых координатах; это верно для многих физических задач со сферической симметрией, определенной в  $R^3$ .

### 2. Команда `gnuplot mapping`, параметры примеры использования

Если данные предоставляются для `splot` в сферических или цилиндрических координатах, то команда `set mapping` должна использоваться для указания `gnuplot`, как их интерпретировать.

Синтаксис:

```
set mapping {декартово | сферическое | цилиндрическое отображение}
```

По умолчанию используется декартова система координат.

### 3. Команда `set term`, терминалы `pdfcairo` и `pngcairo`. Параметры, примеры использования.

Команда `set term` устанавливает формат файла, во втором случае (`pop`) она возвращает исходное значение, в новых версиях `gnuplot` это эквивалентно (как раз окошко с тулбаром)

Если графики планируется вставлять в документ LaTeX, то если используется pdflatex, то в новых версиях gnuplot можно использовать драйвер pdfcairo.

Терминальное устройство pdfcairo генерирует выходные данные в формате pdf. Собственно рисование выполняется с помощью cairo, библиотеки 2D-графики, и rango, библиотеки для разметки и рендеринга текста.

Синтаксис:

установить срок pdfcairo

```
{{no}enhanced} {mono|color} {solid|dashed} {font <font>} {fontscale <font><scale>}
```

```
{linewidth <lw>} {rounded|butt} {dashlength <lw><dl>} {size <dl><XX><unit>,<XX><YY><unit>}
```

Этот терминал поддерживает расширенный текстовый режим, который позволяет встраивать команды шрифта и другие команды форматирования (индексы, надстрочные индексы и т. Д.) в метки и другие текстовые строки.

#### 4. Графические объекты в документах LaTeX.

Для вставки рисунков понадобится пакет **graphicx**. Его, как и другие пакеты LaTeX, необходимо указать в преамбуле документа:

```
\usepackage[argument]{graphicx}
```

Значение аргумента **argument** может быть следующим:

- **dvips** (аргумент по умолчанию при компиляции с помощью **latex**), если вы собираетесь компилировать документ при помощи **latex** для получения DVI-файла.
- **dvipdfm**, если документ компилируется **latex**, и полученный в результате DVI-файл предполагается конвертировать в PDF при помощи **dvipdfm**.
- **pdftex** (аргумент по умолчанию при компиляции с помощью **pdflatex**), если документ предполагается компилировать **pdftex**, чтобы непосредственно получить PDF-файл.

В дальнейшем мы будем использовать для компиляции документа **pdflatex**, поэтому в преамбуле запишем:

```
\usepackage{graphicx}
```

безо всяких опций.

Теперь укажем путь к графическим файлам. Пусть они помещаются в подкаталог **pictures** текущего каталога

```
\graphicspath{{pictures/}}
```

(текущим считается тот каталог, где находится наш файл \*.tex).

Кроме того, в преамбуле можно указать список расширений, которые будут трактоваться как графические

```
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

Теперь, вставляя в документ файл одного из указанных выше типов, его расширение указывать не обязательно.

Итак, для вставки рисунков к преамбуле документа нужно добавить:

```
\usepackage{graphicx}
\graphicspath{{pictures/}}
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

## Раздел 2

### 1. Окружение tabular. Преамбула, разметка.

Решить описанные выше проблемы поможет окружение tabular. Оно обладает большим количеством настроек, а также его возможности могут быть расширены подключением дополнительных пакетов. При создании таблицы с помощью окружения tabular в первую очередь нужно задать количество колонок и выравнивание в них. Для этого у этого окружения есть обязательный параметр: последовательность из символов «l», «c», «r» и «|». Суммарное количество символов «l», «c» и «r» указывает количество колонок в таблице. При этом символ «l» указывает, что выравнивание содержимого в соответствующей колонке будет происходить по левому краю ячейки, символ «r», что выравнивание будет происходить по правому краю, а символ «c» указывает на центрирование содержимого.

Например, команда `\begin{tabular}{lrc}` будет начинать таблицу с четырьмя колонками, где первая колонка будет выровнена по левому краю, вторая — по правому краю, а две последние колонки будут выровнены по центру.

Очевидно, таблица будет читаться легче, если в нее добавить горизонтальные и вертикальные линии, отделяя первую строку и первый столбец. Это не сложно сделать. Символ «|» в обязательном параметре окружения tabular указывает промежутки между колонками, где будет нарисована вертикальная линия на всю высоту таблицы. Для указания горизонтальных линий в таблице, созданной при помощи окружения tabular служит команда `\hline`. При желании, можно использовать несколько символов «|» для обозначения двойных, тройных и т.д. вертикальных линий. Так же можно использовать несколько команд `\hline` подряд.

### 2. Команда \multicolumn, линейки в таблицах.

Гибкий инструмент для обработки многоколоночных документов в L A T E X-это **multicol** , ниже приведен пример:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\usepackage{multicol}

\begin{document}
\begin{multicols}{3}
[
\section{First Section}
Все человеческое подвержено разложению. И когда судьба зовет, Монархи должны повиноваться.
]
Здравствуйте, вот какой-то текст без смысла. Этот текст должен показать
, как будет выглядеть печатный текст в этом месте.
Если вы прочтете этот текст, вы не получите никакой информации. Действительно? Неужели нет
никакой информации? Есть...
\end{multicols}

\end{документ}
```

Чтобы импортировать пакет, строка

```
\usepackage{multicol}
```

добавляется в преамбулу. После того, как пакет импортирован, среда multicols может быть использована. Среда принимает два параметра.

- Количество столбцов, этот параметр должен быть передан внутри фигурных скобок. В приведенном примере его значение равно 3.
- "Текст заголовка", заключенный в квадратные скобки. Это необязательно и будет отображаться поверх многоколоночного текста.
- Текст заключен внутри тегов `\begin{multicols}` и `\end{multicols}` печатается в многоколоночном формате.

### 3. Пересечения линеек.

Возможностей окружения `{array}` вполне хватает для печати простейших линованных таблиц, но в более сложных случаях возникают проблемы. Если подключить стилевой пакет `{hhline}`, работа с линованными таблицами облегчается.

Итак, предположим, что этот пакет подключен. Тогда для задания горизонтальных линеек становится доступной, наряду с уже известными `\hline` и `\cline`, новая команда `\hhline`, в

аргументе которой описывается как сама линейка, так и ее пересечения с вертикальными линейками.

#### 4. Окружение table. Команда \listoftables.

Окружение `{table}` определяет "плавающие таблицы". Все свойства этого окружения дословно совпадают с соответствующими свойствами окружения `{figure}`, за двумя исключениями: во-первых, подпись, генерируемая командой `\caption`, начинается со слова, определенного в команде `\tablename`, и переопределять, при необходимости, надо именно эту команду, и во-вторых, таблицы нумеруются независимо от иллюстраций. Кстати, подпись к таблице принято делать не снизу, как к иллюстрации, а сверху. Окружение `{table*}` при наборе текста в две колонки определяет таблицы шириной в целую страницу.

В документе можно, при желании, получить автоматически сгенерированные списки иллюстраций и/или таблиц. Для этого используются команды `\listoffigures` (для иллюстраций) и `\listoftables` (для таблиц). Их работа аналогична команде `\tableofcontents`, генерирующей оглавление: материал для этих списков собирается в специальные файлы с расширениями `lof` (для иллюстраций) и `lot` (для таблиц); при каждом запуске LaTeX'a информация, записанная в этих таблицах, относится к предыдущему запуску, так что в самом конце может понадобиться запустить LaTeX лишний раз; наконец, команда `\caption` может принимать необязательный аргумент — вариант подписи под иллюстрацией или таблицей, предназначенный для включения в список иллюстраций или таблиц соответственно. Этот необязательный аргумент записывается (в квадратных скобках, как обычно) **перед** обязательным.

Как окружение `{figure}` не рисует картинок, так и окружение `{table}` только размещает таблицу на страницах документа, но не создает ее текста.

Две команды `\listoffigures(lof)` и `\listoftables(lot)` используются таким же образом, как `\tableofcontents` для создания списков со всеми вашими рисунками и таблицами. часть текста, которая при вёрстке может быть перенесена системой TEX в другую часть документа. Таблицы и рисунки могут располагаться вверху/внизу страницы (колонки), на отдельной странице.

✓ Формат окружения table: `\begin {table} [ [hрасположениеi] ] [ \caption {hзаголовок_таблицыi} ]  
hкод_таблицыi \end {table}`

✓ Аналогично оглавлению, список таблиц может быть сгенерирован командой `\listoftables`.