Тема 1

1. Команда gnuplot mapping, параметры примеры использования

Если данные предоставляются для разделения в сферических или цилиндрических координатах, команда **set mapping** должна используется для указания gnuplot, как их интерпретировать.

Синтаксис:

set mapping {cartesian | spherical | cylindrical | cylindrical }
set mapping {Декартово(по умолчанию) | Сферическая | Цилиндрическая}

Для сферической системы координат данные занимают два или три столбца (или с использованием записей). Первые два интерпретируются как азимутальный и полярный углы тета и фи (или «долгота» и «широта») в единицах измерения **set angles**. Радиус r берется из третьего столбца, если он есть, или устанавливается равным единице, если третьего столбца нет.

Отображение:

x = r * cos(theta) * cos(phi)

y = r * sin(theta) * cos(phi)

z = r * sin(phi)

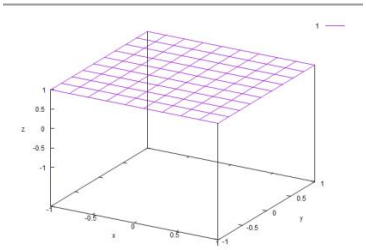
В цилиндрической системе координат данные снова занимают два или три столбца. Первые два интерпретируются как тета (в единицах, **set angles**) и z. Радиус либо берется из третьего столбца, либо устанавливается равным единице, как в сферическом случае. Отображение:

x = r * cos(theta)

y = r * sin(theta)

z = z

Пример:



set xlabel "x"

set ylabel "y"

set zlabel "z"

set xrange [-1:1]

set yrange [-1:1]

set zrange [-1:1]

set mapping spherical

set angles degrees

splot 1

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

Трехмерные координаты

Графики поверхности создаются с помощью команды splot. Обычно поверхность отображается под произвольным углом обзора, так, чтобы он представлял трехмерную поверхность. В таком случае оси X, Y и Z видны на графике. В иллюзия 3D улучшается за счет удаления скрытых линий или элементов поверхности с сортировкой по глубине.

В splot, нулевая точка оси Z размещается выше плоскости XY. Чтобы переместить начало координат на плоскость XY, используется **set ticslevel** . Когда **ticslevel=0**, ноль оси Z перемещается на плоскость XY.

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

Создание графических файлов

Set output Имя.расширение - Команда для сохранения

Gnuplot поддерживает несколько десятков графических форматов

Set terminal jpeg

Set terminal Имя_терминала Опция – команда для выбора терминала

Show terminal

Опции:

1) linewidth<lw> - ширина линии

2) interlace / nointerlace – переплетение графиков

3) enhaced/no enchanced – влияет на отступ шрифтов, стрелочки

4) dashleght – длина пунктирных линий

5) toht – имя шрифта

6) размер шрифта: tiny, small, medium,lange,giant

7) nounded – углы

8) font scale – маштабирование шрифта

9) size <x>,<y> - размер по X и Y

10)background <rgb цвет> - фон

Команда pngcairo

Создает график в формате png с использованием библиотеки pango

set term pngcairo

{{no}enhanced} {mono|color}

{{no}transparent} {{no}crop} {background <rgbcolor>

{font <font>} {fontscale <scale>}

{linewidth <lw>} {rounded|butt|square} {dashlength <dl>}

{size <XX>{unit},<YY>{unit}}

По умолчанию размер – 640x480 пикселей.

Команда pdfcairo

Создает график в формате pdf с использованием библиотеки pango.

График успешно транслируется командой pdfcario.

```
set term pdfcairo
                        {{no}enhanced} {mono|color} {solid|dashed}
                        {font <font>}
                        {linewidth <lw>} {rounded|butt} {dashlength <dl>}
                        {size <XX>{unit},<YY>{unit}}
```

По умолчанию: размер рисунка 5x3 дюйма, толщина линий 0.75

4.Графические объекты в документах LaTeX

В системе LATEX существует набор команд, с помощью которого можно создавать несложные рисунки. Для этой цели предназначено окружение picture:

\begin{picture}(<ширина>,<высота>)(x0,y0)

…

\end{picture}

которое формирует графический бокс заданных размеров.

Параметры (<ширина>,<высота>), задаваемые как неотрицательные числа, являются обязательными и определяют ширину и высоту бокса. Они указывают размер свободного пространства, которое необходимо зарезервировать для размещения графического объекта.

Вторая пара параметров (x0,y0) является необязательной и задаёт координаты левого нижнего угла графического бокса. Если эти параметры не указывать, то у левого нижнего угла будут координаты (0,0).

Числовые параметры внутри окружения \picture, если не указывается иное, измеряются в единицах длины unitlength и записываются только числами без указания единицы измерения. По умолчанию единица длины unitlength равна одном пункту, т. е. \unitlength=lpt. Чтобы длины измерялись, на пример, в миллиметрах, нужно в преамбуле или в основном тексте присвоить новое значение единице измерения длины: \unitlength=1mm («единицу» перед mm указывать обязательно) Размеры могут задаваться не только целыми, но и десятичными числами, в которых необходимо использовать десятичную точку. Отметим некоторые общие правила работы с окружением picture. Внутри окружения не должно быть пустых строк. Графический бокс рассматривается TEX'ом как один большой символ, поэтому, если окружение picture поместить в сере дину абзаца, то бокс будет помещён в строку, причём соседние строки раздвинутся таким образом, чтобы он поместился. Следовательно, лучше помещать окружение picture между абзацами (после пустой строки или команды \par).

Также можно вставлять окружение picture внутри других окружений, таких как figure, table, center и пр.

Команды \thinlines и \thicklines позволяют переключаться между тонкими и толстыми линиями соответственно. По умолчанию установлены тонкие линии

Для изменения толщины вертикальных и горизонтальных линий используют команду \linethickness{<Толщина>} где параметр задаёт толщину линий и выражается в любых TEX'овских единицах измерения длины. Например, строка \linethickness{7.5mm}

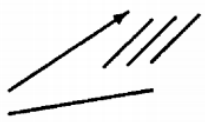
Рисование простейших объектов

Рассмотрим команды, с помощью которых внутри окружения pictur е можно размещать графические объекты. Команда \put(x,y){<Графический объект>} помещает графический объект так, чтобы его точка привязки находилась в точке (x,y). Заметим, что вместо <Графического объекта> может находиться обычный текст. Команду \put можно и не использовать, но тогда точка привязки будет находиться в текущей точке рисунка. С помощью команды \multiput(x,y) (dx,dy){n}{<Графический объект>} можно нарисовать п копий графического объекта. Параметр (x,y) задаёт положение первого объекта, а параметр (dx,dy) определяет смещение каждой последующей копии относительно предыдущей.

Рассмотрим графические объекты, которые можно размещать внутри окружения picture . Команда \line (<x-наклон>,<y-наклон>)}{<длина>} рисует отрезок заданной из точки привязки.

Команда \vector (<x-наклон>,<y-наклон>)}{<длина>} создает вектор.

Пример 9.1.



```
\begin{picture}(80,80)\thicklines
\put(10,10){\line(6,1){60}}
\put(10,20){\vector(3,2){50}}
\multiput(50,30)(10,1){3}{\line(1,1){20}}
\end{picture}
```

Команда \circle{<Диаметр>} рисует окружность с центром в точке привязки и заданного

Пример 9.2.

```
\begin{picture}(90,90)\thicklines
\put(30,40){\circle{40}}
\put(70,40){\circle*{20}}
\end{picture}
```

С помощью команды \qbezier [<количество точек>] (Xa,Ya) (Xb,Yb) (Xc,Yc) можно провести кривую Безье через три точки с координатами (Xa,Ya),(Xb,Yb) и (Xc,Yc).

Пример 9.3.

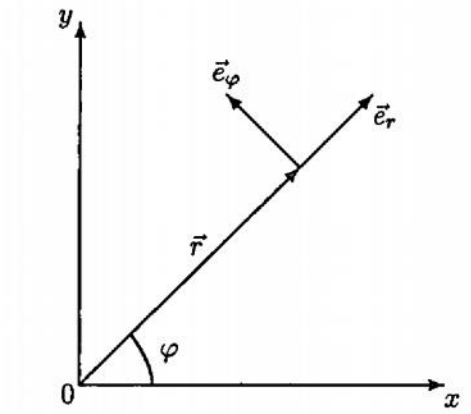
```
\begin{picture}(100,100)\thicklines
\qbezier(1,1)(48,10)(20,65)
\put(10,10){\qbezier[50](41,1)(88,10)(60,65)}
\end{picture}
```

Команда \oval(<ширина>,<высота>)[<часть>] рисует овал, т. е. прямоугольник с закруглёнными углами. Параметр является необязательным и позволяет рисовать часть овала: t — верхняя половина, b — нижняя поло вина, 1 — левая половина, г — правая половина.

Пример 9.4.

```
\begin{picture}(100,70)\thicklines
\put(30,30){\oval(50,30)}
\put(70,30){\oval(50,40)[r]}
\end{picture}
```

Пример 9.5.




```
\thicklines\unitlength=1cm
\begin{picture}(6,6)
\put(0.5,0.5){\vector(1,0){5}}
\put(0.5,0.5){\vector(0,1){5}}
\put(0.25,0.25){0}
\put(3.5,0.2){$x$}
\put(0.2,5.5){$y$}
\put(0.5,0.5){\vector(1,1){3}}
\put(2,2.3){$\vec r$}
\qbezier(1.5,0.5)(1.5,0.8)(1.2,1.2)
\put(1.6,0.9){$\varphi$}
\put(3.5,3.5){\vector(1,1){1}}
\put(3.5,3.5){\vector(-1,1){1}}
\put(4.5,4.1){$\vec e_r$}
\put(2.3,4.7){$\vec e_{\varphi}$}
\end{picture}
```

Импортирование графики

Чтобы вставить в документ более сложное изображение, созданное в какой-то другой программе, необходимо использовать предоставляемую пакетом graphicx команду \includegraphics[<список параметров>]{<Имя файла>} где параметр задает имя графического файла с расширением.

Тема 2

1. Окружение tabular. Преамбула, разметка.

В преамбуле нужно добавить пакет **\usepackage{array}**

Окружение **tabular** обладает большим количеством настроек, а также его возможности могут быть расширены подключением дополнительных пакетов.

Разметка

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}{lrcc}** 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

Пример:

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

```
\begin{tabular}{crrrr}
  & 2 & 3 & 4 & 5 \\
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125 \\
\end{tabular}
```

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.

Пример:

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

```
\begin{tabular}{c||rrrr|}
  & 2 & 3 & 4 & 5 \\
\hline
\hline
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125 \\
\hline
\end{tabular}
```

Помимо обязательного параметра у окружения **tabular** есть еще необязательный параметр, который отвечает за вертикальное выравнивание таблицы относительно окружающего текста, если таблица используется внутри абзаца, а не отдельно. Этот параметр может иметь одно из трех значений **«t», «c» и «b»**, которые указывают на выравнивание по верхней строке, по центру и по нижней строке соответственно.

Пример:

Ехал Грека через	реку. руку. рак.	Видит Грека — в реке	реку. руку. рак.	Сунул Грека	реку руку рак	в	реку. руку. рак.

```
Ехал Грека через
\begin{tabular}[t]{l}
реку. \\
руку. \\
рак.
\end{tabular}
Видит Грека --- в реке
\begin{tabular}[b]{l}
реку. \\
руку. \\
рак.
\end{tabular}
Сунул Грека
\begin{tabular}[c]{l}
реку \\
руку \\
рак
\end{tabular}
в
\begin{tabular}[t]{l}
реку. \\
руку. \\
рак.
\end{tabular}
```

Если требуется создать сложное распределение элементов в одной из ячеек таблицы, можно использовать вложение одной таблицы в другую. Эта возможность позволяет создавать гораздо более сложные конструкции.

Блочная матрица. Пример:

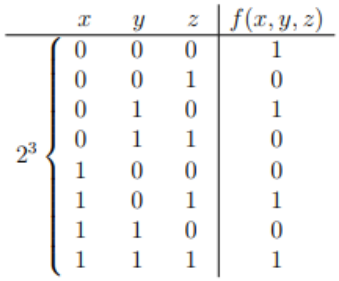
$\begin{matrix}1 & 0\\ 0 & 1\end{matrix}$	**O**	**I**
O	$\begin{matrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{matrix}$	**O**
I	**O**	$\begin{matrix}0 & 1\\ 1 & 0\end{matrix}$

```
\begin{tabular}{c|c|c}
\begin{tabular}{cc}
1 & 0\\
0 & 1
\end{tabular}
  & {\bf O} & {\bf I} \\ \hline
{\bf O} &
       \begin{tabular}{ccc}
       0 & 1 & 0 \\
       1 & 0 & 1 \\
       0 & 1 & 0 \\
       \end{tabular}
      & {\bf O} \\ \hline
{\bf I} & {\bf O} &
                    \begin{tabular}{cc}
                    0 & 1\\
                    1 & 0\\
                    \end{tabular} \\
\end{tabular}
```

Окружение **tabular** можно использовать как в текстовом, так и в математическом режиме, но при использовании в математическом режиме будут наблюдаться некоторые проблемы: 7 даже если вы поместите таблицу, созданную с помощью **tabular** в математическое окружение внутри **tabular** по прежнему будет текстовый режим и вся информация будет отображаться соответствующим шрифтом. Кроме того, математические формулы внутри такой таблицы также потребуют повторного перехода в математический режим. Для использования в режиме редактирования

формул, существует специальное окружение **array**, практически идентичное по возможностям окружению **tabular**.

Пример:



$$
\begin{array}{rrr|c}
x^^ & y^^ & z^^ & f(x,y,z)\\
\hline
2^3 \left\{ % Общая фигурная скобка
\begin{array}{r} % Вложенная таблица для каждого столбца
0\\ 0\\ 0\\ 0\\ 1\\ 1\\ 1\\ 1\\
\end{array}
\right.
&
\begin{array}{r}
0\\ 0\\ 1\\ 1\\ 0\\ 0\\ 1\\ 1\\
\end{array}
&
\begin{array}{r}
0\\ 1\\ 0\\ 1\\ 0\\ 1\\ 0\\ 1\\
\end{array}
&
\begin{array}{r}
1\\ 0\\ 1\\ 0\\ 1\\ 0\\ 1\\ 0\\
\end{array}
\end{array}
$$

2. Команда \multicolumn, линейки в таблицах.

Документы с двумя столбцами можно легко создать, передав параметр \twocolumn в оператор класса документа. Если вам нужна большая гибкость в расположении столбцов или для создания документа с несколькими столбцами, пакет multicol предоставляет для этого набор команд.

Для импорта пакета строка\ usepackage {multicol}добавляется в преамбулу. После импорта пакета можно использовать среду multicols. Окружение принимает два параметра:

1) Число столбцов. Этот параметр должен быть передан в фигурные скобки,

2) «Текст заголовка», который вставляется в квадратные скобки. Это необязательно и будет отображаться поверх текста, состоящего из нескольких столбцов.

Разделение столбцов

Разделение колонок определяется командой \ columnsep.

setlength {\ columnsep} {Разделение столбцов(в ед.измерения)}

Вставка вертикальных линеек

Вертикальную линейку можно вставить в качестве разделителя столбцов, чтобы улучшить читаемость некоторых документов

\ usepackage {цвет}. Эта строка вставлена в преамбулу, чтобы разрешить использование нескольких цветов в документе.

\ setlength {\ columnseprule} {ширина линейки} Это определяет ширину линейки, которая будет использоваться в качестве разделителя столбцов, по умолчанию она равна 0. В этом примере печатается столбец шириной 1 пункт.

\ def \ columnseprulecolor {\ color {синий}} Цвет разделительной линейки установлен на синий.

\ columnbreak Эта команда вставляет точку останова столбца. В этом случае поведение текста отличается от ожидаемого. Вставляется разрыв столбца, затем абзацы перед точкой останова равномерно распределяются, чтобы заполнить все доступное пространство.

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.

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.

Пример:

А	Б	В	Г
Д	Е	Ж	З

```
\begin{tabular}{|c|cc|c|}
\hline А & Б & В & Г\\
\hhline{|=|~~|-|}
Д & Е & Ж & З\\ \hline
\end{tabular}
```

Аргумент команды \hhline устроен следующим образом. Во-первых, в нем сказано, что на территории первой колонки линейка должна быть двойной (символ =), на территории второй и третьей колонок линейки не должно быть вовсе (символ ~ — «тильда»), а на территории четвертой колонки линейка должна быть одинарной (символ -). Если в таблице n колонок, то в аргументе \hhline должны присутствовать n символов -, = или ~, имеющих тот же смысл, что и выше.
Между этими символами, описывающими поведение линейки внутри колонок, расположены символы, описывающие пересечения горизонтальной линейки с вертикальными
Пример получения линий:

На печати	⊢	⊣	+, ⊤, ⊥	⊨	⫤	⫩, ⫪, ⫫
В аргументе \hhline	|-	-|	-|-	|=	=|	=|=
На печати	⊦	⫞	⫩, ⫪, ⫫	⫦	⫣	⫳
В аргументе \hhline	:=	=:	=:=	||-	-||	-||-
На печати	⊩	⫢	⧺	⫵	⫥	⫳, ⫪, ⫫
В аргументе \hhline	|:=	=:|	=::=	#=	=#	=#=
На печати	⫡	⫠	⫟	⫤	⫪	⫫
В аргументе \hhline	|t:=	|b:=	=:t|	=:b|	=:t:=	=:b:=

4. Окружение table. Команда \listoftables.

Плавающие объекты

Чтобы избежать полупустых страниц, таблица или рисунок, не помещающиеся на текущей странице, должны «Плавать», т.е. перемещаться на следующую страницу при заполнении текстом текущей.

LATEX приписывает плавающие объектам порядковые номера, отдельно для таблиц и рисунков. В списки таблиц и рисунков включаются только те из них, которые определены как плавающие объекты.

Плавающие объекты задаются следующими окружениями:

\begin{figure}[<положение>}

Рисунок

\caption[<Короткий заголовок>]{<заголовок>}

\end{figure}

Для вставки рисунков, или для вставки таблиц

\begin{table}{<положение>]

Рисунок или таблица

\caption[<Короткий заголовок>]{<заголовок>}

\end{table}

Необязательный параметр <Положение> определяет правила размещения объекта, он может быть задан через следующие символы или их комбинации:

h – позиция в тексте, где вводится плавающий объект

t – верхняя часть страницы

b – нижняя часть страницы

p - на отдельной странице для плавающих объектов

Команда **\listoftables** создает список таблиц