

# Графики и простые рисунки в L<sup>A</sup>T<sub>E</sub>X

Какойто Какойтов

13 июня 2016 г.

## 1 Подготовка к работе

Итак, запускаем любой L<sup>A</sup>T<sub>E</sub>X редактор: TeXworks, TeXStudio и др. Создаём новый файл, а в нём:

- `\documentclass[a4paper,12pt]{article}` - указываем класс и некоторые параметры документа(статья формата A4 и базовый шрифт - 12pt);
- `\usepackage[utf8]{inputenc}` - кодировка;
- `\usepackage[english,russian]{babel}` - поддержка русского языка;
- `\usepackage{tikz}` - включаем tikz;
- `\usepackage{pgfplots}` - включаем расширенный pgf;
- `\usepackage{pgfplotstable}` - то же;
- `\title{Название}` - параметры титульной страницы  
`\author{Автор}`  
`\date{Сегодня}`
- `\begin{document}` - начало документа  
`\maketitle` - выводим титульный лист  
Здесь начинаем рисовать:)  
`\end{document}`

## 2 Основы

### 2.1 Простая команда TIKZ

Здесь используется простая команда `\tikz{код}`:



```
\tikz{\draw (0, 0)--(1, 0)--(1, 1)--(0, 1)--(0, 0)}
```

`draw` - рисовать что-то (линию, точку);

`(0, 0)` - координаты точки `(x, y)` - левый нижний угол;

`(0, 0)–(1, 0)` - линия между двумя точками;

```
\tikz[xscale=.5, yscale=2]{\draw (0, 0)--(1, 1)}
```

изменение размера по вертикали (`yscale`) и горизонтали (`xscale`);

```
\tikz{\draw[x=10pt,y=10pt] (0, 0)--(1, 2)}
```

можно задать свои единицы измерения (`pt`, `px`, `sm`, `mm`). По-умолчанию в  $\text{\LaTeX}$  используются свои ед. измерения.

### 2.2 TIKZ-окружение

Но эта короткая команда из одной строки используются для рисования простой фигуры или линии. Если нужно что-то посложнее, используется окружение `tikzpicture`. При этом точка отсчёта координат будет находиться в начале строки и при рисовании различных линий и фигур надо это учитывать.



`\begin{tikzpicture}` - окружение рисунка

`\draw [line width=2mm] (0, 0)--(2, 1);` - линия толщиной 2мм

`\draw[dashed] (2, 1)--(4, 0);` - прерывистая линия

`\draw[loosely dotted] (4,0)--(6,1);` - разреженная прерывистая линия

`\draw[red] (6, 1)--(8, 0);` - линия красного цвета

`\draw[green] (8, 0) -- ++(0, 0) -- +(1, 1) -- +(1, 0) -- +(2, 0);`

- Обратите внимание, при рисовании этой линии строка: `(8, 0)–+++(0, 0)` обнулила точку начала координат и каждая новая точка будет отклады-

ваться от неё:  $+(1,1)$  «плюс обязателен»  
`\end{tikzpicture}`

Толщина линий:

ultra thin   very thin   thin   semithick   thick   very thick   ultra thick

Прерывистость линий:

loosely dashed   densely dashed   loosely dotted   densely dotted

## 2.3 Фигуры



- прямоугольник

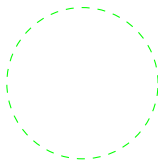
`\draw [thin, blue] (0, 0) rectangle(2, 1.5);`

thin, blue - тонкая линия синего цвета;

(0,0) - точка отсчёта(левый нижний угол);

rectangle - прямоугольник;

(2,1.5) - координаты угла, противоположного точке начала коорд.



- круг

`\draw[dashed,green] (0, 0) circle[radius=1];`

thin, blue - прерывистая линия зелёного цвета;

(3, 1) - точка отсчёта;

circle - круг;

radius=1.2 - радиус 1.2см, можно записать сокращённо: (1.2)

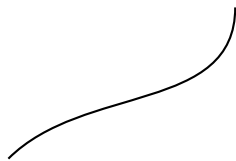


- дуга

`\draw[purple] (0, 0) arc [radius=1, start angle=90, end angle=240];`

purple - фиолетовая линия;

(3, 1) - точка отсчёта;  
 arc - дуга;  
 radius=1 - с радиусом 1(с центром в воображаемой точке);  
 start angle=90 - точка начала дуги;  
 end angle=240 - точка конца дуги;  
 параметры в [] можно записать сокращённо - (240:90:1);



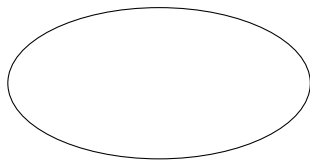
- кривая

```
\draw (0,0) to [out=45,in=270] (3,2)
```

кривая толстая линия с началом в точке (0, 0);

выходит из неё под углом  $45^\circ$

и заходит в точку (3,2) под углом  $270^\circ$



- эллипс

```
\draw (0, 0) ellipse (2 and 1);
```

эллипс с центром в точке (0, 0);

вертикальный радиус - 2;

горизонтальный радиус - 1;



- кривая Безье

```
\draw (0,0) .. controls (1,1) and (2,1) .. (2,0)
```

кривая Безье, выходящая из точки (0, 0)

и приходящая в точку (2, 0);

точки контроля кривой (1,1) и (2,1);

## 2.4 Заливка фигур цветом



```
\draw[fill] (4, 1) circle[radius=.5];
```

По умолчанию - заливка чёрным цветом

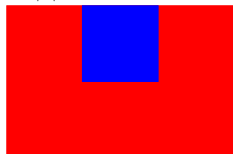


Красный прямоугольник

```
\draw[fill=red, red] (0, 0) rectangle(2, 1)
```

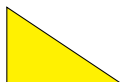
fill=red - заливка красным цветом

ещё один red после - обводка фигуры красным же цветом



Фигуры можно накладывать друг на друга(своя команда для каждой фигуры):

```
\begin{tikzpicture}
\draw[fill=red, red] (0, 0) rectangle(3, 2);
\draw[fill=blue, blue] (1, 1) rectangle(2, 2);
\end{tikzpicture}
```



Заливку и рамку фигуры можно указать такой командой...

```
\draw[fill=yellow] (0, 0)--(0, 1)--(1.5, 0)--(0, 0)
```



...или такой

```
\filldraw[fill=magenta,draw=green,thick] (7, 0) to [out=90, in=180] (8, 1) --(8,
```



```
\path[fill=magenta] (9, 0)--(9, 1) to [out=0, in=90] (10, 0)--(9,0);
```

## 2.5 Градиенты



- градиент-заливка

```
\shade (0,0) rectangle (2,1)
```

По умолчанию переход от серого(gray) к белому(white)



```
\shade[top color=cyan,bottom color=orange] (0,0) rectangle (2,1.5)
```

Градиент с указанием цветов(сверху вниз)



```
\shadedraw[left color=green,right color=blue,draw=black, thick]
(0,0) rectangle (2.5,1)
```

Градиент справа налево с чёрной рамкой(-draw в названии команды указывает на наличие рамки)



```
\shade[inner color=white,outer color=magenta] (0,0) rectangle +(2,2)
```

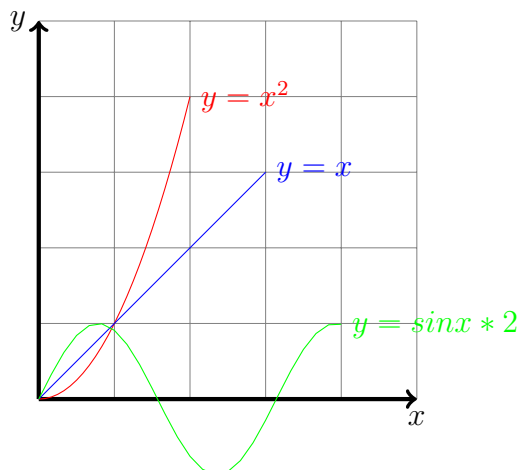
Радиальный градиент(внутри белый, снаружи малиновый)



```
\shade[ball color=red] (14,.5) circle (.5cm)
```

Предустановленный градиент в виде шара

## 2.6 Коорд-я сетка и графики функций



```
1. \draw[help lines] (0, 0) grid (5, 5)
```

Рисует сетку с ячейкой в 1см(по умолчанию) тонкими линиями

2. `\draw[ultra thick, ->] (0, 0)--(0, 5) node[left] {$y$}`

Ось Y в виде стрелки, где:

`node[left] {$y$}` - узел наименования(подпись) в виде буквы y слева от стрелки

3. `\draw[ultra thick, ->] (0, 0)--(5, 0) node[below] {$x$}`

Здесь тоже самое для оси X

4. `\draw[red, domain=0:2] plot (\x, \x*\x) node[right] {$y=x^2$}`

Этой строкой непосредственно рисуем график:

`domain=0:2` - в каких пределах оси X отобразить график

`plot (\x, \x*\x)` - сам график, где: `\x` - икс(неизвестное), `\x*\x` - формула графика -  $x^2$

5. `\draw[blue, domain=0:3] plot (\x, \x) node[right] {$y=x$}`

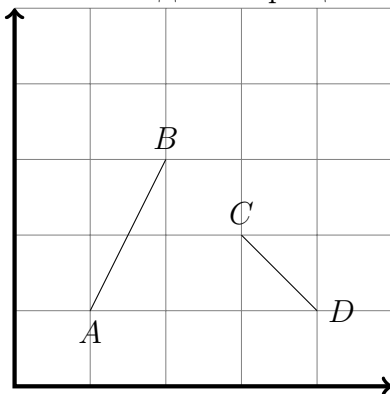
График попроще

6. `\draw[green, domain=0:4] plot (\x, {\sin(\x*2 r)}) node[right] {$y=\sin x*2$}`

Здесь выражение функции обернуто в скобки `{}`

## 2.7 Координатные точки

Как правило если нам нужна простая линия, кривая или фигура, можно её рисовать как мы делали выше. Если же рисунок сложнее и через одну точку проходит несколько линий лучше задать её с помощью специальной команды и обращаться к ней по заданному имени. Например:



`\coordinate[label=below:$A$] (A) at(1,1);`

`coordinate` - команда установки воображаемой точки по указанным далее координатам

`label=below:$A$` - необязательный параметр, для отображения на рисунке названия данной точки(A) и некоторое её смещение вниз(below) относительно себя, чтобы было красивее:)

(A) - краткое название точки для дальнейшего её использования

`at(1,1)` - собственно сами координаты точки

Другие точки:

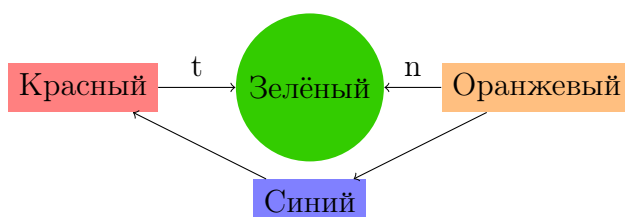
```
\coordinate[label=$B$] (B) at(2,3);
\coordinate[label=$C$] (C) at(3,2);
\coordinate[label=right:$D$] (D) at(4,1);
```

```
\draw (A)--(B) (C)--(D);
```

начертить оба отрезка можно одной командой, используя ранее данные имена точкам

## 3 Простая диаграмма

### 3.1 Nodes в диаграмме



```
1.\begin{tikzpicture}[node distance=3cm]
```

окружение tikz с указанием для всех узлов(node) расстояния между ними(от центра до центра), для отдельных node расстояние можно указать внутри окружения

```
2.\node[fill=red!50] (node1) {Красный};
```

node - создаём ноду,

fill= - заполняем её цветом(red!50 - означает берём 50% красного и смешиваем с остальными 50% по-умолчанию белого)

node1 - имя ноды для дальнейшего использования

{...} - здесь название ноды

```
3.\node[shape=circle, fill=green!80!red, right of=node1]
```

```
(node2) {Зелёный};
```

shape=circle - фигура ноды в виде круга(если не указано - прямоугольник)

fill=green!80!red - заполнение цветом(80% зелёного и 20% красного)

right of=node1 - указываем место, где нода будет изображена(справа от первой, на расстояние, определённое ранее - 3см)

затем имя и название ноды

```
4.\node[fill=blue!50, below of=node2, node distance=1.5cm]
```

```
(node4)
```



Синий; - здесь определяем расстояние самостоятельно

```
5.\node[fill=orange!50, right of=node2] (node3) {Оранжевый};
```

```
6.\draw[->](node1) edge node[above] {t} (node2);
```

рисуем между нодами(node1 и node2) стрелку

edge - показывает как рисовать её(от края одной, до края другой ноды)

node[above] {t} - определяет ещё одну вспомогательную ноду, надпись

над(above) стрелкой в виде буквы t

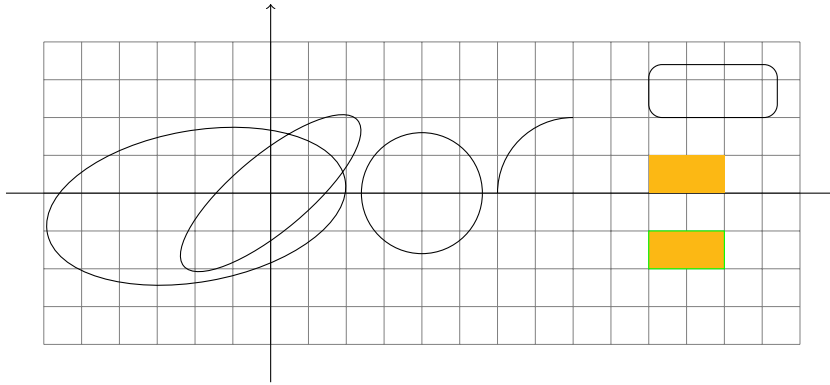
```
7.\draw[->](node3) edge node[above] {n} (node2);
```

```
8.\draw(node3) edge[->] (node4);
```

здесь между нодами вспомогательной ноды нет, пишем команду попроще

```
9.\draw(node1) edge[<-] (node4);
```

### 3.2 Разное



```
\draw[help lines, step=.5cm] (-3,-2) grid (7, 2);
```

сетка со своим шагом, диапазоном координат и системой отсчёта

```
\draw[rotate=10] (-1,0) ellipse (2 and 1) [rotate=30] (0,0)
```

```
ellipse(1.5 and .5);
```

два эллипса нарисованные одной командой с поворотом вокруг своей оси

```
\draw (2, 0) circle(.8)
```

создание круга, краткая запись

```
\draw (4, 1) arc(90:180:1);
```

создание дуги круга, краткая запись

```
\draw[rounded corners=5] (5, 1) rectangle(6.7, 1.7);
```

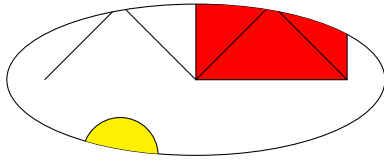
прямоугольник со скруглёнными углами

```
\fill[yellow!70!red] (5, 0) rectangle(6, .5);
```

создание прямоугольника, закрашенного смешанным цветом

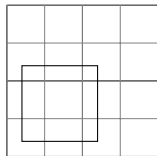
```
\draw[fill=yellow!70!red, draw=green] (5,-.5) rectangle(6,-1)
```

прямоугольник, нарисованный зелёной линией и залитый смешанным цветом



```
\draw[clip] (0, 0) ellipse(2.5 and 1);
```

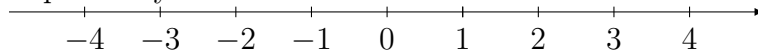
Здесь с помощью команды `clip` эллипс выступает в роли рамки, внутри которой видны другие фигуры, а снаружи нет



```
\draw[shift={(.2,.2)}] (0,0) rectangle(1,1);
```

Смещение фигуры на 0.2 единицы относительно своего места.

Как правило смещение используется в циклах для рисования чего-либо через равные промежутки, например использование цикла для маркировки луча:



```
\draw[-latex] (-5,0)--(5,0);
```

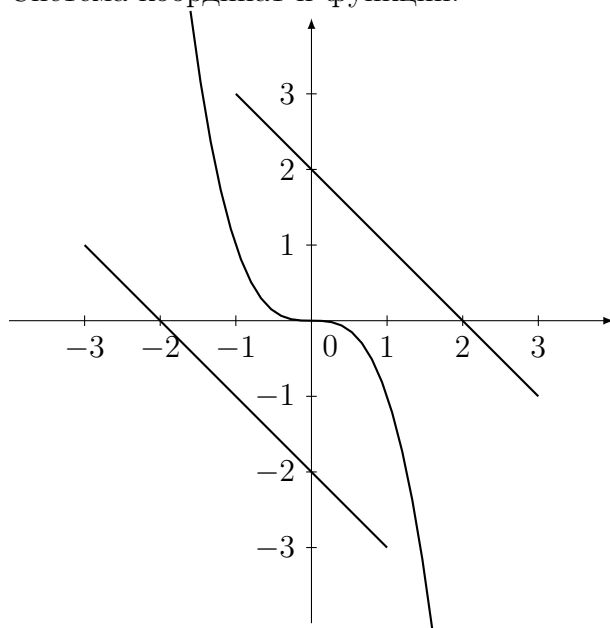
```
\foreach \x in {-4,-3,-2,-1,0,1,2,3,4,}
```

```
\draw[shift={(\x,0)}] (0,.07)--(0,-.07) node[below]{$\x$};
```

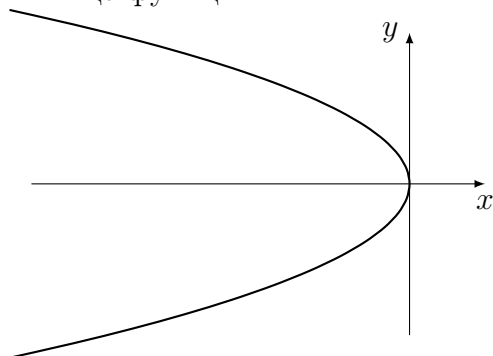
## 4 Различные рисунки

### 4.1 Графики функций

Система координат и функции.



Ещё функция



### 4.2 Отрезки и плоские фигуры

Нахождение точек на прямой:

```
\coordinate[label=below:$A$] (A) at(0,0);
```

```
\coordinate[label=below:$B$] (B) at(3,2);
```

- определяем координатные точки А и В на рисунке: [название точки на рисунке] (сокращенное название для последующего использования в командах) at(x, y)-координаты точки

`\draw (A)--(B);`

- прочертим между ними линию

`\draw (A) ($(A)!.5!(B)$) coordinate [label=below:$K$] (K);`

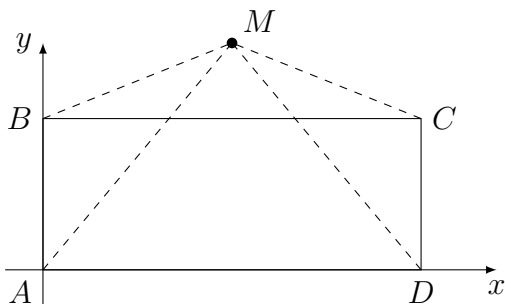
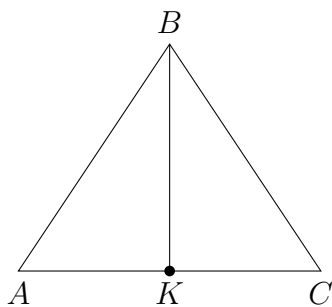
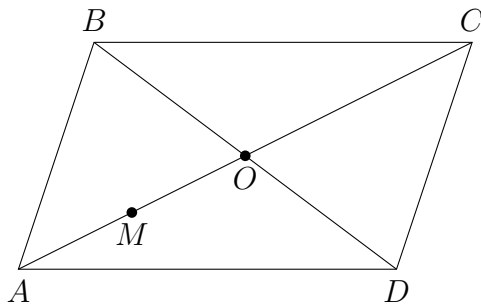
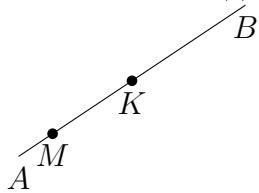
- нахождение точки K на прямой AB:  $AK = 1/2 AB$  (не чертя лишнюю линию)

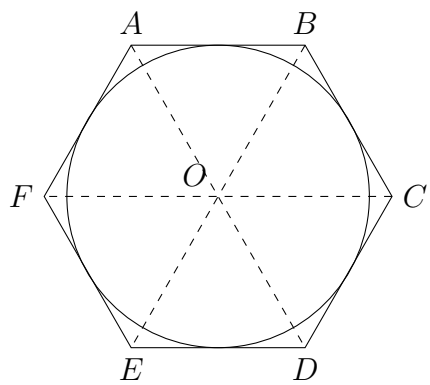
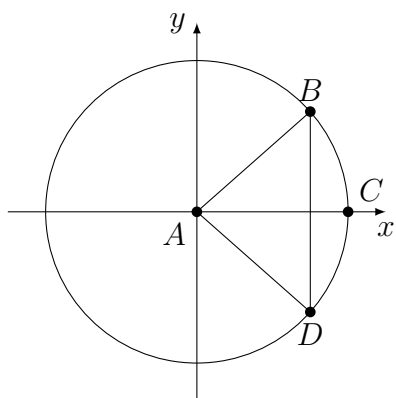
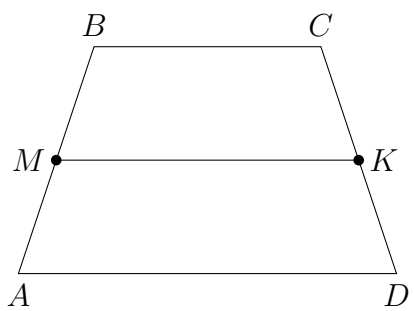
`\draw (A) ($(A)!.3!(K)$) coordinate [label=below:$M$] (M);`

- нахождение точки M:  $AM = 1/3 AK$

`\fill (K) circle(2pt) (M) circle(2pt);`

- обозначим найденные точки черными кружками





### 4.3 Объёмные фигуры

