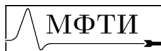


Библиотека векторной графики TikZ

Кирилл Павлов

Московский физико-технический институт



19 ноября 2011 г.

- 1 История создания, обзор возможностей
- 2 Сравнение с альтернативными подходами
- 3 Библиотеки *TikZ*
- 4 Картинки

- 1 История создания, обзор возможностей
- 2 Сравнение с альтернативными подходами
- 3 Библиотеки *TikZ*
- 4 Картинки

Определения

Определение

TikZ /PGF — Язык описания векторной графики на основе алгебраических и геометрических структур.

PGF (Portable Graphics Format) — низкоуровневый язык.

TikZ (TikZ ist kein Zeichenprogramm = TikZ is not a drawing program) – надстройка над PGF.

Зачем нужен TikZ ?

- Создание и генерация векторных изображений, 3D графики, анимации
- Вставка графики в pdf (.dvi, .ps), интерпретируется L^AT_EX-ом. Можно вставлять векторную и растровую графику вместе

Note: не нужно использовать TikZ для вставки графиков.

Автор: Prof. Dr. Till Tantau (1975 –)



- 1992 One of the national winner in the German computer science competition
- 1993 Silver medaillist at the internation olympiade of informatics in Mendoza, Argentina
- 1993 Stipend by the Studienstiftung des Deutschen Volkes
- 1994 Dr. Habbenawild award for an outstanding abitur
- 1999 Erwin-Stephan-Award of the TU Berlin
- 2002 Best student paper award at MFCS 2002
- 2003 Dissertation thesis is nominated for the GI-dissertation awards
- 2007 Teaching award of the students of the faculty of technology and natural sciences

<http://www.tcs.uni-luebeck.de/en/mitarbeiter/tantau/>

Getting started

Для начала нужно подключить окружение `\usepackage{tikz}` и необходимые библиотеки `\usetikzlibrary{}`.

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{
    arrows
    ,shapes
    ,trees
    ...
}
\begin{document}
...
\end{document}
```

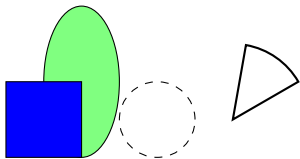
Окружение команд

Используйте `\tikz` для однострочных команд



```
\tikz \fill[orange] (0,0) circle (1ex);
```

Используйте `\begin{tikzpicture} ... \end{tikzpicture}` для больших изображений



```
\begin{tikzpicture}
\draw[style=dashed] (2,.5) circle (0.5);
\draw[fill=green!50] (1,1)
ellipse (.5 and 1);
\draw[fill=blue] (0,0) rectangle (1,1);
\draw[style=thick]
(3,.5) -- +(30:1) arc(30:80:1) -- cycle;
\end{tikzpicture}
```


Примитивы TikZ

- **node:** ○

```
\tikz \node at ( 0,0) [circle,draw] {};
```

- **path:** —

```
\tikz \draw (0pt,0pt) -- (20pt,6pt);
```

Paths

- Базовые элементы, являются последовательностью прямых или изогнутых линий
- Могут быть нарисованы (**drawn**), заполнены (**filled**), перекрыты другими путями (**clipping**)



```
\path[draw, line width=2pt] (0, 0)--(1, 1)--(2, 0)--cycle;
```



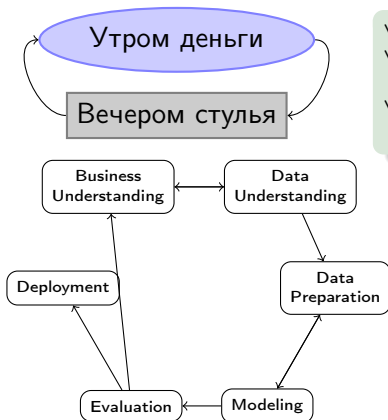
```
\path[fill=blue!20] (0, 0) -- (1, 1) -- (2, 0) -- cycle;
```



```
\begin{tikzpicture}
\path[clip, draw] (0, 0) -- (1, 1) -- (2, 0) -- cycle;
\path[fill=blue!50] (1, 1) circle (1);
\end{tikzpicture}
```

Nodes

- Базовые элементы, могут быть добавлены к путям
- Узлам можно давать имена и обращаться к ним в дальнейшем



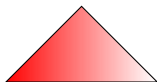
```

\begin{tikzpicture}
\node[first] (morning) {Утром деньги};
\node[second] (evening) [below of=morning] {Вечером стулья};
\draw [->] (morning) to [out=0,in=0] (evening);
\end{tikzpicture}

```

CRISP-DM — методология ведения проектов интеллектуального анализа данных

Shading



```
\tikz \shade[left color=red, draw]
      (1,1)--(2,2)--(3,1)--cycle;
```



```
\tikz \shade[top color=red,
              bottom color=green] (0,0) rectangle (2,1);
```



```
\tikz \shade[shading=ball,
              ball color=blue] (0,0) ellipse (1 and 0.5);
```



```
\begin{tikzpicture}
\shade[shading=ball, ball color=red] (0,0) circle (.3);
\shade[shading=ball, ball color=yellow] (.7,0) circle (.3);
\shade[shading=ball, ball color=green] (1.4,0) circle (.3);
\end{tikzpicture}
```



```
\tikz \shade[left color=gray, right color=green,
              draw=green!50!black] (4,0)-- +(30:1cm)
              arc (30:60:1cm) -- cycle;
```

Изогнутые линии

- Специфицируются несколько контрольных точек между точками пути
- Кривая строится из начальной точки пути в направлении контрольной, затем изменяет направление в сторону второй контрольной и входит в конечную точку пути (кубическая интерполяция Безье).

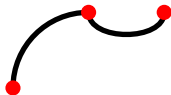


```
\tikz \draw[line width=2pt] (0, 0) ..
controls(1,1) .. (2.5, 0);
```



```
\tikz \draw [line width=2pt] (-1,0) .. controls
(-1,0.555) and (-0.555,1) .. (0,1) ..
controls (0.555,1) and (1,0.555) .. (1,0);
```

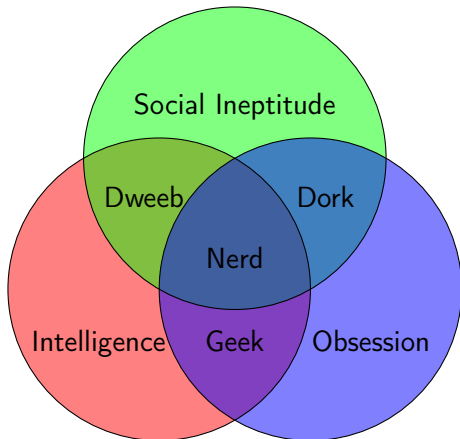
- Можно так же указать направление кривой в начальной и конечной точках.



```
\tikz \draw[line width=2pt] (0,0) to [out=90, in=180]
(1,1) to [out=-90, in=-90] (2,1);
```

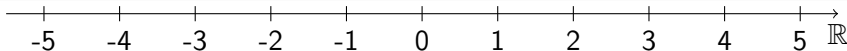
Clipping and Scope

- Используйте команду **clip** для скрепки изображений. Рисоваться будут только области внутри них
- Окружение **scope** нужно чтоб изображения не выходили за границу области



```
\begin{scope}
\clip \firstcircle;
\clip \secondcircle;
\clip \thirdcircle
\fill[color=rgb]; % Nerd
\end{scope}
```

Циклы



```
\draw[->] (-5.5,0) -- (5.5,0) node [below] {\mathbb{R}};
\foreach \x in {-5,...,5}
\draw (\x, 0.1) -- (\x, -0.1) node [below] {\x};
```



```
\foreach \x in {1,3,...,10}
\shade[ball color=red!\x 0!green] (\x,0) circle (3mm);
```



```
\foreach \x in {9,...,1}
\draw[fill=blue!\x 0] (-0.1*\x - 1, -0.1*\x )
rectangle (0.1*\x + 1, 0.1*\x );
```

Ссылки вне текущего изображения

- Добавьте в глобальные опции

```
\tikzstyle{every picture}+=[remember picture]
```

- Добавьте опцию `overlay` всем `paths`, содержащим `nodes` вне текущего изображения
- Запустите `pdflatex` дважды.
- Слово `"paths"` сверху на самом деле `node`:

```
...всем \tikz[baseline,inner sep=0] \node[anchor=base]  
(n1) {paths};, содержащим ...
```

- Отрисовка стрелки:

```
\tikz[overlay]\draw[thick,green,->] (n2)  
to [out=40, in=-90] (n1);
```


Вставка в beamer

Экспонентный класс распределений

$$p(\mathbf{y} | \boldsymbol{\theta}) = \exp \left(\mathbf{T}(\mathbf{y})^\top \boldsymbol{\eta}(\boldsymbol{\theta}) - b(\boldsymbol{\theta}) + c(\mathbf{y}) \right)$$

- Достаточная статистика
- Предиктор
- Некоторые функции

Вставка в beamer

Экспонентный класс распределений

$$p(\mathbf{y} | \boldsymbol{\theta}) = \exp \left(\mathbf{T}(\mathbf{y})^\top \boldsymbol{\eta}(\boldsymbol{\theta}) - b(\boldsymbol{\theta}) + c(\mathbf{y}) \right)$$

- Достаточная статистика
- **Предиктор**
- Некоторые функции


Вставка в beamer

Экспонентный класс распределений

$$p(\mathbf{y} | \boldsymbol{\theta}) = \exp \left(\mathbf{T}(\mathbf{y})^\top \boldsymbol{\eta}(\boldsymbol{\theta}) - b(\boldsymbol{\theta}) + c(\mathbf{y}) \right)$$

- Достаточная статистика
- Предиктор
- Некоторые функции

Перед запуском

- 1 Поставьте последнюю версию *TikZ*
`http://tex.stackexchange.com/questions/2044/how-to-install-a-current-version-of-tikz`.
- 2 Скачать (если требуются) необходимые пакеты (*tikzlibrary*) и положить их в папку с компилируемым *.tex* файлом.
`http://dante.ctan.org/get/graphics/pgf/base/tex/generic/pgf/libraries/`
- 3  в помощь

- 1 История создания, обзор возможностей
- 2 Сравнение с альтернативными подходами
- 3 Библиотеки *TikZ*
- 4 Картинки

Альтернативные подходы

- ❶ Стандартный модуль $\text{\LaTeX}\{\text{picture}\}$
- ❷ Пакет **pstricks** <http://www.tug.org/PSTricks/main.cgi/>
- ❸ Пакет **xypic** <http://www.tug.org/applications/Xy-pic/>
- ❹ Пакет **dratex**
- ❺ Программа **metapost**
<http://www.tug.org/metapost.html>
<http://ru.wikipedia.org/wiki/MetaPost>
- ❻ Програма **xfig** <http://www.xfig.org/>

Стандартный модуль $\text{\LaTeX}\{\text{picture}\}$

Позволяет создавать простые графики, но не многим больше. Используется, в основном, для вставки графических файлов (`\includegraphics{}`)

<http://www.math.uiuc.edu/~hildebr/tex/packages.html>

Пакет `dratex`

Достаточно мал для создания графики. Его может не хватить.

Пакет **pstricks**

Достаточно мощный для создания графики (любых мыслимых типов), но не портируемый, не работает с pdf_{tex} и чем либо, что не создает PostScript код. По сравнению с pgf имеет больше дополнений (пакетов). Для **pstricks** существует много пакетов для конкретных целей

Стоит заметить, что **pstricks** может много того, что невозможно для pdf. В частности, он может получить доступ к языку PostScript, что позволит такие вещи, как inline function plotting (`plot(x,y)`).

Рисунки, которые нельзя сделать в *TikZ*, но можно в **pstricks**

Пакет хуріс

Старый пакет сложный для понимания.

Программа The metapost

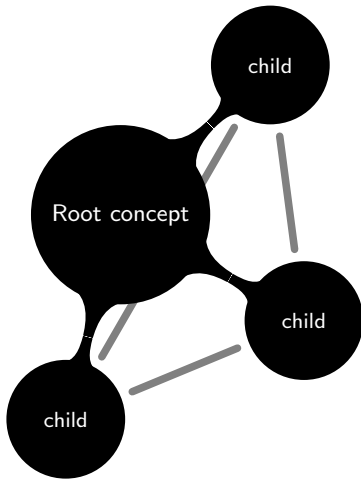
Серьезная альтернатива pgf. Однако, это внешняя программа. К её недостаткам можно отнести долгую компиляцию и проблемы с пометкой (label) картинок.

Програма xfig

Альтернатива TikZ для пользователей, которые не любят программировать. <http://www.xfig.org/>

- 1 История создания, обзор возможностей
- 2 Сравнение с альтернативными подходами
- 3 Библиотеки TikZ**
- 4 Картинки

Mindmap



```

\begin{tikzpicture}
[small mindmap, text=white]
\node [concept] {Root concept}
[clockwise from=45]
child { node[concept] (c1) {child}}
child { node[concept] (c2) {child}}
child { node[concept] (c3) {child}};
\begin{pgfonlayer}{background}
\draw [concept connection] (c1) edge (c2)
edge (c3)
(c2) edge (c3);
\end{pgfonlayer}
\end{tikzpicture}

```

Mindmap

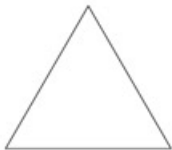


L-systems (Системы Линденмайера)

(картинки из презентации Алены Фаворской)

<http://evelands.net/evan/lssystem02.php>

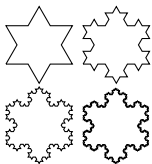
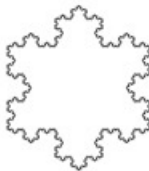
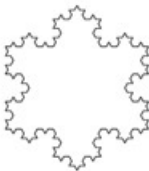
Koch curve



initiator

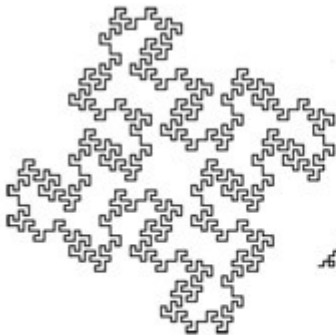


generator



```
\tikz \draw [scale=5/(3^1)][l-system={Koch curve, angle=60,
axiom=F++F++F, order=1}] lindenmayer system;
```

Quadratic and Triangle Koch

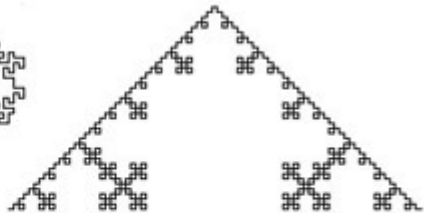


$$n = 2, \delta = 90^\circ$$

F-F-F-F

$F \rightarrow F+FF-FF-F-F+F+F$

F-F-F+F+FF+FF-F

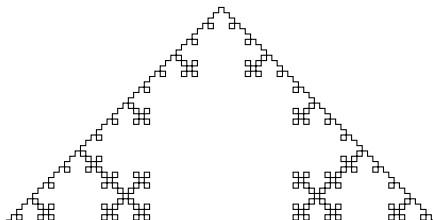
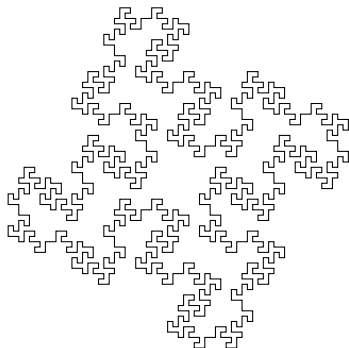


$$n = 4, \delta = 90^\circ$$

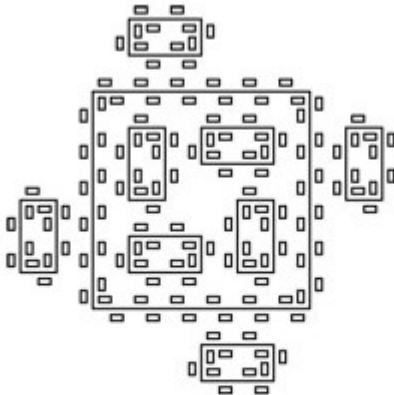
-F

$F \rightarrow F+F-F-F+F$

Quadratic and Triangle Koch



Islands and Lakes



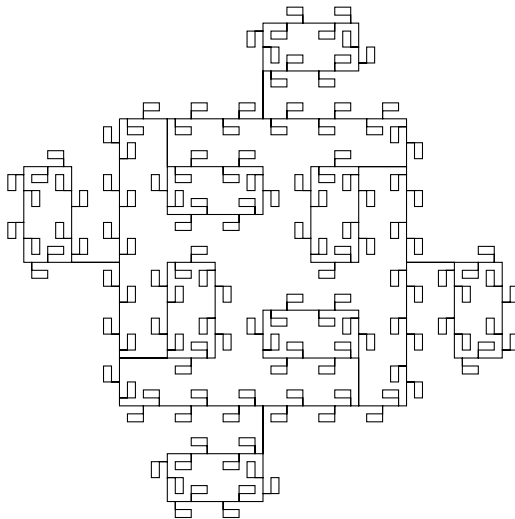
$n = 2, \delta = 90^\circ$

$F + F + F + F$

$F \rightarrow F + f - FF + F + FF + Ff + FF - f + FF - F - FF - Ff - FFF$

$f \rightarrow fffff$

Islands and Lakes



Trees



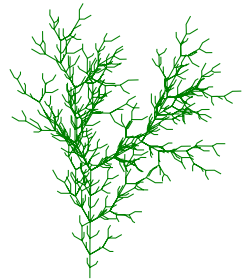
a
 $n=5, \delta=25.7^\circ$
 F
 $F \rightarrow F[+F]F[-F]F$



b
 $n=5, \delta=20^\circ$
 F
 $F \rightarrow F[+F]F[-F][F]$



c
 $n=4, \delta=22.5^\circ$
 F
 $F \rightarrow FF[-F+F+F] + [F-F-F]$

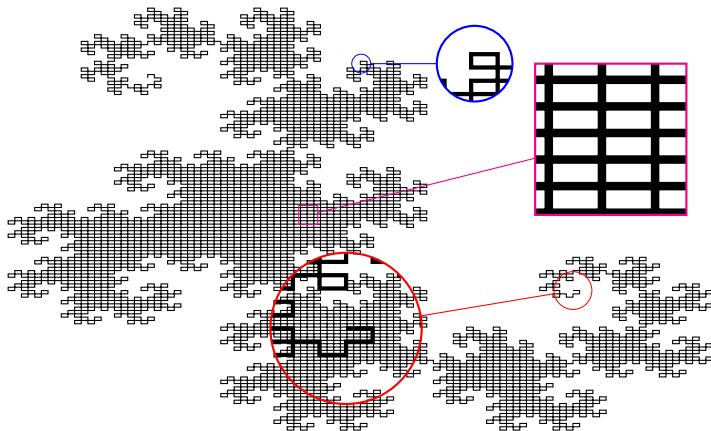


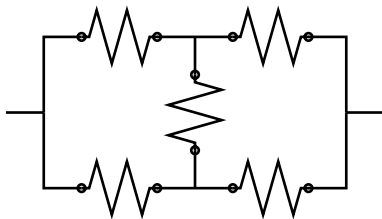
$n=4, \delta = 30^\circ$
 F
 $F \rightarrow FF[-F+F] + [+F-F]$

Spy library



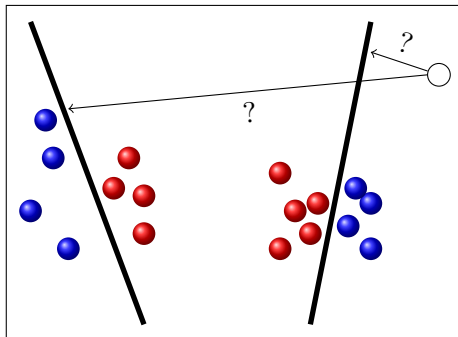
Spy library





- 1 История создания, обзор возможностей
- 2 Сравнение с альтернативными подходами
- 3 Библиотеки *TikZ*
- 4 Картинки**





- ❶ <http://www.texample.net/tikz/> — примеры
- ❷ <http://mirror.ctan.org/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf> — документация по TikZ v. 2.10
- ❸ <http://www.ctan.org/pkg/pgf> — страница пакета pgf

- 1 Что является базовыми элементами изображения в *TikZ* ?

- 1 Что является базовыми элементами изображения в TikZ ?
nodes + paths

2 Что выведет программа?

```
\begin{scope}[fill=orange]
\fill[clip] (0,0) rectangle (10,10);
\foreach \angle in {0,30,...,360}
\path[fill=red] (5,5) -- +(\angle:10)
    arc(\angle:\angle+15:10) -- cycle;

\node[anchor=base,inner sep=0] (image) at (4.55,2.2)
    {\includegraphics[scale=1.2]{pictures/tp}};
\end{scope}
```

Типичный физтех

знает, что выведет
программа

Если у Вас есть вопросы, то сейчас самое время их задать!