KU LEUVEN

Vector graphics in PGF/Ti*k*Z

I.e. how to make kick-a*s graphics for your academic documents

Giuliano Bernardi[1] & Joseph Szurley[2]

[1]CONNEXOUNDS
[2]Google

March 01, 2024, Leuven, Belgium

0 – Outline

1. Introduction
2. Quick TikZ graphics overview
   - Basic commands and concepts
   - Creating scientific graphs with `pgfplots`
3. Generate/edit Tikz code

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

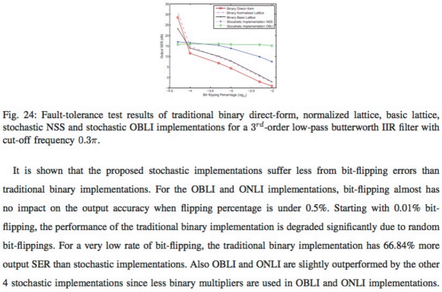

Binary Direct–form
Binary Normalized Lattice
Binary Basic Lattice
Stocahstic Implementation NSS
Stocahstic Implementation OBLI
Output SER (dB)
35
30
25
20
15
10
5
0
-5
-4.5
-4
-3.5
-3
-2.5
-2
Bit-flipping Percentage ($\log_{10}$)

Fig. 24: Fault-tolerance test results of traditional binary direct-form, normalized lattice, basic lattice, stochastic NSS and stochastic OBLI implementations for a 3^{rd}-order low-pass butterworth IIR filter with cut-off frequency 0.3π.

It is shown that the proposed stochastic implementations suffer less from bit-flipping errors than traditional binary implementations. For the OBLI and ONLI implementations, bit-flipping almost has no impact on the output accuracy when flipping percentage is under 0.5%. Starting with 0.01% bit-flipping, the performance of the traditional binary implementation is degraded significantly due to random bit-flippings. For a very low rate of bit-flipping, the traditional binary implementation has 66.84% more output SER than stochastic implementations. Also OBLI and ONLI are slightly outperformed by the other 4 stochastic implementations since less binary multipliers are used in OBLI and ONLI implementations.

Figure: Rasterized graphic (IEEE TSP 2016)

1 – Motivation

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

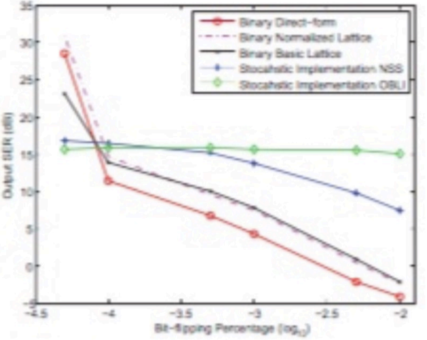


Figure: Rasterized graphic: zoom 400%

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

scientist: hmm what happens if I include this figure at 0.3\columnsize

paper: *sucks*

scientist:

https://knowyourmeme.com/memes/surprised-pikachu

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

scientist: hmm what happens if I include this figure at 0.3\columnsize

paper: *sucks*

scientist:

https://knowyourmeme.com/memes/surprised-pikachu

1 – Motivation

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

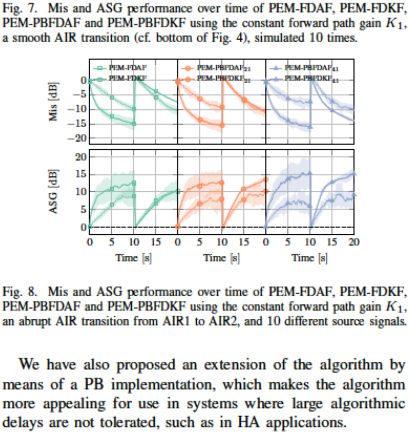


Fig. 7. Mis and ASG performance over time of PEM-FDAF, PEM-FDKF, PEM-PBFDAF and PEM-PBFDKF using the constant forward path gain K_1, a smooth AIR transition (cf. bottom of Fig. 4), simulated 10 times.

PEM-FDAF
PEM-FDKF
PEM-PBFDAF$_{21}$
PEM-PBFDKF$_{21}$
PEM-PBFDAF$_{41}$
PEM-PBFDKF$_{41}$
Mis [dB]
0
−5
−10
−15
−20
ASG [dB]
20
15
10
5
0
0 5 10 15 0 5 10 15 0 5 10 15 20
Time [s]
Time [s]
Time [s]

Fig. 8. Mis and ASG performance over time of PEM-FDAF, PEM-FDKF, PEM-PBFDAF and PEM-PBFDKF using the constant forward path gain K_1, an abrupt AIR transition from AIR1 to AIR2, and 10 different source signals.

We have also proposed an extension of the algorithm by means of a PB implementation, which makes the algorithm more appealing for use in systems where large algorithmic delays are not tolerated, such as in HA applications.

Figure: Vector graphic

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

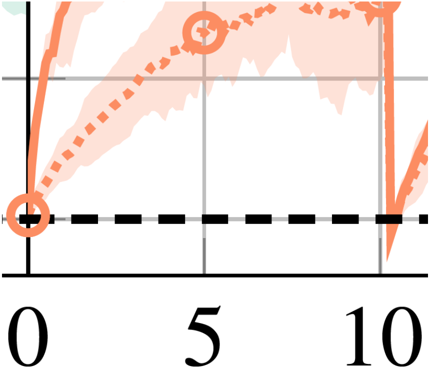


Figure: Vector graphic: zoom 1600%

1 – Motivation

Electronic Artwork

- Picture is worth a 1000 words
- Artwork elucidates main points and results
- Rasterized images scale poorly: psd, .tif, .jpg, .gif, and .bmp.
- Vector artwork **reduced** or **enlarged** in size indefinitely without loss of quality: .eps, .ps, .pdf
- Simple version control and parametrization

1 – Motivation

Vector Graphics without TikZ

- MATLAB, Python - eps, pdf output
- Sometimes it looks like it doesn't belong
- Size of image known *a-priori*: wrong font size
- Iterative process: time consuming

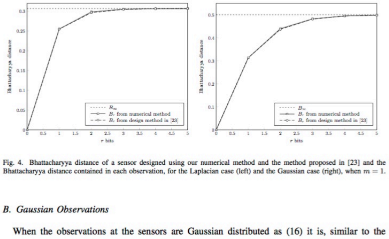


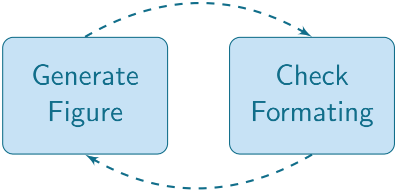
Fig. 4. Bhattacharyya distance of a sensor designed using our numerical method and the method proposed in [23] and the Bhattacharyya distance contained in each observation, for the Laplacian case (left) and the Gaussian case (right), when $m = 1$.

B. Gaussian Observations

When the observations at the sensors are Gaussian distributed as (16) it is, similar to the

Vector Graphics without TikZ

- MATLAB, Python - eps, pdf output
- Sometimes it looks like it doesn't belong
- Size of image known *a-priori*: wrong font size
- Iterative process: time consuming



Generate Figure

Check Formating

PGF

- Portable Graphics format
- Written in TeX
- Platform-/format- independent
- Works with (pdf-)(LaTeX)TeX and ConTeXt
- Creates either PDF or PS output
- Basic layer

1 – Introduction

PGF

- Portable Graphics format
- Written in TeX
- Platform-/format- independent
- Works with (pdf-)(LaTeX) and ConTeXt
- Creates either PDF or PS output
- Basic layer

Tikz

- Ti*k*Z ist *kein* Zeichenprogramm
- Descriptive syntax based on METAFONT
- (Best?) Frontend for PGF
- Lots of high-level libraries

1 – Pros/cons of TikZ

Pros

- Quick creation of simple graphics
- Integration with LaTeX(superior typography, portability, free)
- High-quality vector graphics (precise positioning, scaling of lines)
- Customizability (lots of Ti*k*Z libraries)
- Use of macros and parametrizability
- Version control friendly

1 – Pros/cons of TikZ

Pros

- Quick creation of simple graphics
- Integration with LaTeX(superior typography, portability, free)
- High-quality vector graphics (precise positioning, scaling of lines)
- Customizability (lots of Ti*k*Z libraries)
- Use of macros and parametrizability
- Version control friendly

Cons

- Steep learning curve
- No WYSIWYG
- Long compilation time
- Cryptic errors → Might make you want to smash your laptop[a]

[a] Perhaps less so nowadays with LLMs not getting bored at listening to you

2 – Using TikZ

How to set up TikZ

```
\usepackage{tikz}
\usetikzlibrary{...}
```

How to set up TikZ

```
\usepackage{tikz}
\usetikzlibrary{...}
```

The `\tikzpicture` environment

```
\begin{tikzpicture}
  ⋮
\end{tikzpicture}
```

2 – Using TikZ

How to set up TikZ

```
\usepackage{tikz}
\usetikzlibrary{...}
```

The `\tikzpicture` environment

```
\begin{tikzpicture}
  ⋮
\end{tikzpicture}
```

The `\tikz` command

```
\tikz ...
```

How to set up TikZ

```
\usepackage{tikz}
\usetikzlibrary{...}
```

The \tikzpicture environment

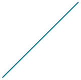
```
\begin{tikzpicture}
  ⋮
\end{tikzpicture}
```

Calling a .tikz file

```
\begin{figure}
    \input{my_tikzpic.tikz}
    \caption{...}
    \label{...}
\end{figure}
```

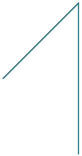
The \tikz command

```
\tikz ...
```



Drawing a line.

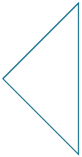
```
\begin{tikzpicture}
    \draw (0,0) -- (1,1);
\end{tikzpicture}
```



Drawing a line.

```
\begin{tikzpicture}
    \draw (0,0) -- (1,1) -- (1,-1);
\end{tikzpicture}
```

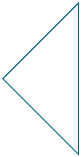
2 – The command \draw



Drawing a line.

```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```

2 – The command \draw

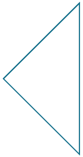


Drawing a line.

```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```

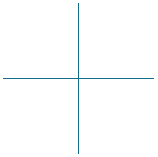
Drawing simple shapes.

```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```



Drawing a line.

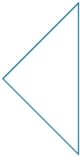
```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```



Drawing simple shapes.

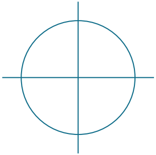
```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```

2 – The command \draw



Drawing a line.

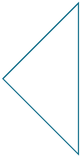
```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```



Drawing simple shapes.

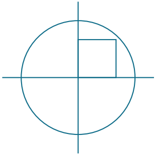
```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```

2 – The command \draw



Drawing a line.

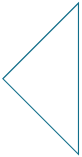
```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```



Drawing simple shapes.

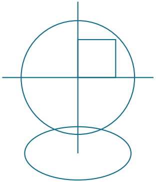
```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```

2 – The command \draw



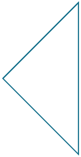
Drawing a line.

```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```



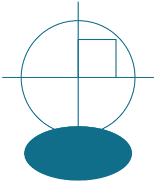
Drawing simple shapes.

```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```



Drawing a line.

```
\begin{tikzpicture}
    \draw (0,0) -- (1,1)  -- (1,-1) -- cycle;
\end{tikzpicture}
```



Drawing simple shapes.

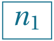
```
\begin{tikzpicture}
    \draw (-1,0) -- (1,0);
    \draw (0,-1) -- (0,1);
    \draw (0,0) circle (0.75);
    \draw (0,0) rectangle (0.5,0.5);
    \draw (0,-1) ellipse (20pt and 10pt);
    \filldraw (0,-1) ellipse (20pt and 10pt);
\end{tikzpicture}
```

n_1

Drawing a node.

```
\begin{tikzpicture}
    \node (n1) at (0,0) {$n_1$};
\end{tikzpicture}
```

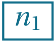
2 – The command \node



Drawing a node.

```
\begin{tikzpicture}
    \node[draw] (n1) at (0,0) {$n_1$};
\end{tikzpicture}
```

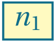
2 – The command \node



Drawing a node.

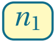
```
\begin{tikzpicture}
    \node[draw, thick] (n1) at (0,0) {$n_1$};
\end{tikzpicture}
```

2 – The command \node



Drawing a node.

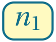
```
\begin{tikzpicture}
    \node[draw, thick, fill=yellow!30!white]
    (n1) at (0,0) {$n_1$};
\end{tikzpicture}
```



Drawing a node.

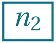
```
\begin{tikzpicture}
    \node[draw, thick, fill=yellow!30!white,
   rounded corners] (n1) at (0,0) {$n_1$};
\end{tikzpicture}
```

2 – The command \node



Drawing a node.

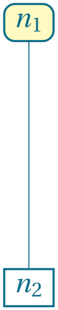
```
\begin{tikzpicture}
    \node[draw, thick, rounded corners] (n1)
    at (0,0) {$n_1$};
\end{tikzpicture}
```



Drawing another node and connect the two.

```
\begin{tikzpicture}
    \node[draw,thick] (n2) at (0,-4) {$n_2$};

\end{tikzpicture}
```

2 – The command \node



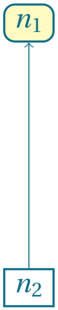
Drawing a node.

```
\begin{tikzpicture}
    \node[draw, thick, rounded corners] (n1)
    at (0,0) {$n_1$};
\end{tikzpicture}
```

Drawing another node and connect the two.

```
\begin{tikzpicture}
    \node[draw,thick] (n2) at (0,-4) {$n_2$};
    \draw (n2) -- (n1);
\end{tikzpicture}
```

2 – The command \node

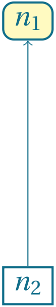


Drawing a node.

```
\begin{tikzpicture}
    \node[draw, thick, rounded corners] (n1)
    at (0,0) {$n_1$};
\end{tikzpicture}
```

Drawing another node and connect the two.

```
\begin{tikzpicture}
    \node[draw,thick] (n2) at (0,-4) {$n_2$};
    \draw[->] (n2) -- (n1);
\end{tikzpicture}
```



Drawing a node.

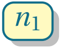
```
\begin{tikzpicture}
    \node[draw, thick, rounded corners] (n1)
    at (0,0) {$n_1$};
\end{tikzpicture}
```

Drawing another node and connect the two.

```
\begin{tikzpicture}
    \node[draw,thick] (n2) at (0,-4) {$n_2$};
    \draw[->] (n2) -- (n1);
\end{tikzpicture}
```

KU LEUVEN

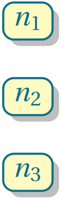
2 – Define a node style



Drawing a node with style.

```
\begin{tikzpicture}
    \tikzset{mynode/.style={draw,
    thick,fill=yellow!30!white,
    rounded corners,drop shadow}}

    \node[mynode] (n1) at (0,0) {$n_1$};




\end{tikzpicture}
```

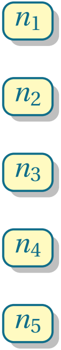
2 – Define a node style



Drawing a node with style.

```
\begin{tikzpicture}
    \tikzset{mynode/.style={draw,
    thick,fill=yellow!30!white,
    rounded corners,drop shadow}}

    \node[mynode] (n1) at (0,0) {$n_1$};
    \node[mynode] (n2) at (0,-1) {$n_2$};
    \node[mynode] (n3) at (0,-2) {$n_3$};



\end{tikzpicture}
```

2 – Define a node style



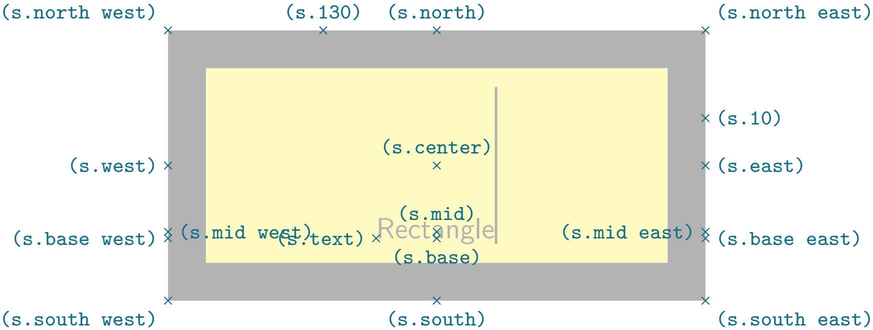
Drawing a node with style.

```
\begin{tikzpicture}
    \tikzset{mynode/.style={draw,
    thick,fill=yellow!30!white,
    rounded corners,drop shadow}}




    \foreach \x in {1,...,5}
     \node[mynode] (n\x) at (0,1-\x)
    {$n_\x$};

\end{tikzpicture}
```

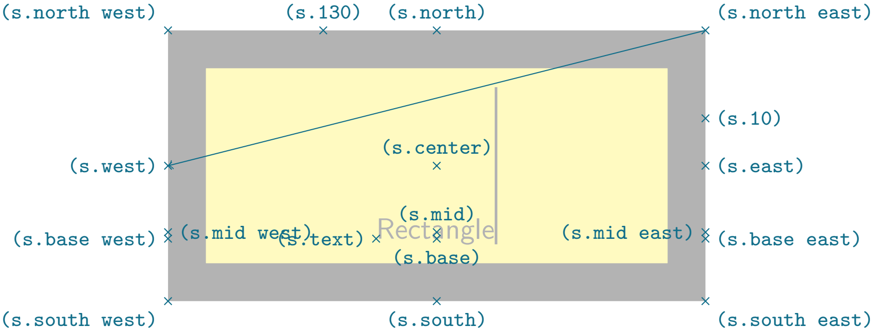
2 – Define a node style

Drawing a node with style.

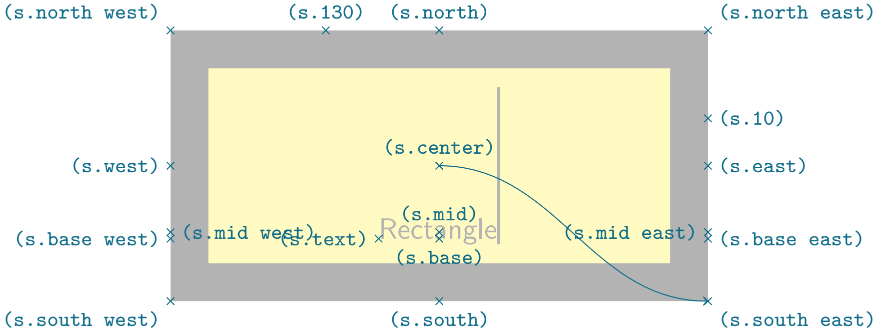
```
\begin{tikzpicture}
    \tikzset{mynode/.style={draw,
    thick,fill=yellow!30!white,
    rounded corners,drop shadow}}




\end{tikzpicture}
```


(s.north west)
(s.130)
(s.north)
(s.north east)
(s.10)
(s.center)
(s.west)
(s.east)
(s.mid)
(s.base west)
(s.mid west)
(s.text)
Rectangle
(s.mid east)
(s.base east)
(s.base)
(s.south west)
(s.south)
(s.south east)

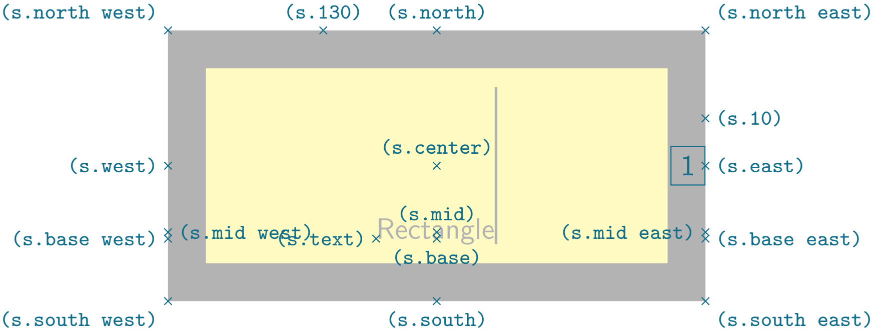
```
\draw[->] (s.north east) -- (s.west);
```



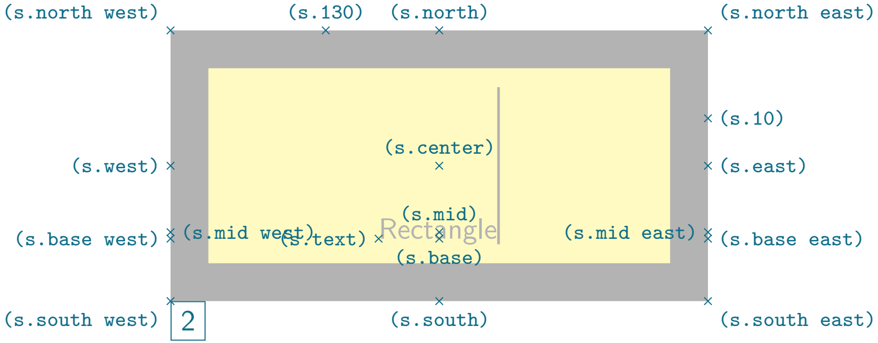
```
\draw[->] (s.center) to [out=0, in=180]  (s.south east);
```



```
\node[draw,anchor=east] at (s.east) {1};
```

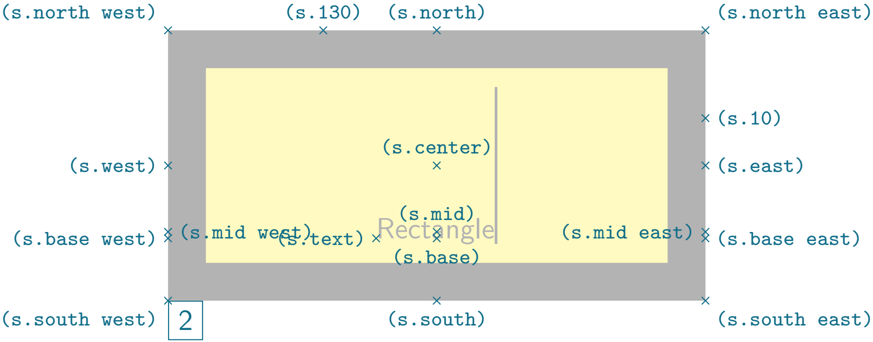


```
\node[draw,anchor=north west] at (s.south west) {2};
```



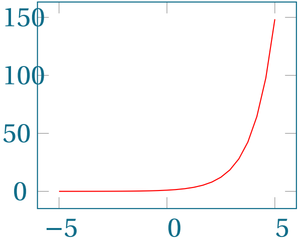
2 – Anchors

```
\node[draw,below right] at (s.south west) {2};
```



The package pgfplots

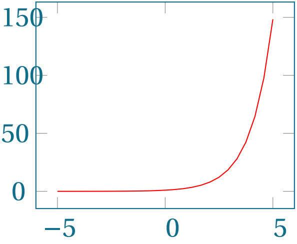
- Separate package, `\usepackage{pgfplots}`
- Built on Ti*k*Z/PGF
- Support function plotting
- Support data plotting
- Might slow down compilation time (externalization)



Plotting a 2D function.

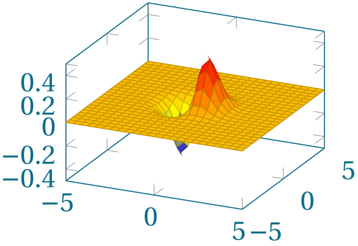
```
\begin{tikzpicture}
  \begin{axis}[width=5cm]
      \addplot[color=red]{exp(x)};
  \end{axis}
\end{tikzpicture}
```

2 – Plotting functions in `pgfplots`



Plotting a 2D function.

```
\begin{tikzpicture}
  \begin{axis}[width=5cm]
      \addplot[color=red]{exp(x)};
  \end{axis}
\end{tikzpicture}
```



Plotting a 3D function.

```
\begin{tikzpicture}
  \begin{axis}[width=5cm]
      \addplot3[surf]
    {exp(-x^2-y^2)*x};
  \end{axis}
\end{tikzpicture}
```

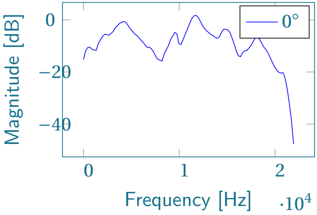
2 – Plotting data in pgfplots

We have the following data saved in HRTF.tsv

Freq. [Hz]	Amplit. [dB]
0	-15.2
221	-11.9
443	-10.6
⋮	⋮
21939	-47.6

We have the following data saved in HRTF.tsv

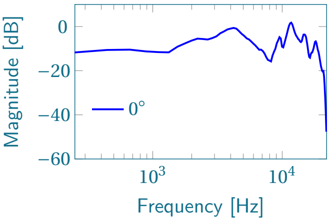
Freq. [Hz]	Amplit. [dB]
0	-15.2
221	-11.9
443	-10.6
⋮	⋮
21939	-47.6

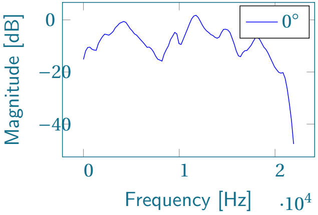


Plotting a collection of data.

```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]}
    ]
    \addplot [color=blue]
    table{HRTFs.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```

2 – Plotting data in pgfplots

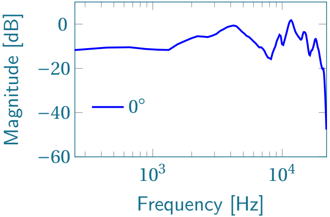


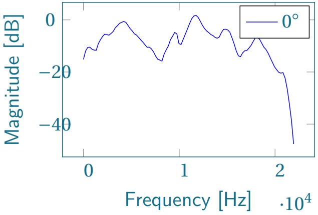


Plotting a collection of data.

```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]},
    xmode=log,
    xmin=250,xmax=22050,
    ymin=-60,ymax=10,
    yticklabel style={text width=1.5em},
    xticklabel style={text width=2em,
      align=center},
    legend style={at={(axis cs:300,-30)},
      anchor=north west,fill=none, draw=none,legend
      cell align=left}
     ]
    \addplot [color=blue,line width=1.5pt]
      table{HRTF.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```

2 – Plotting data in pgfplots

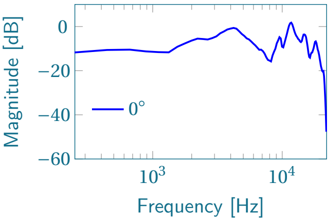


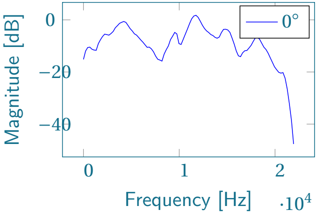


Plotting a collection of data.

```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]},
    xmode=log,
    xmin=250,xmax=22050,
    ymin=-60,ymax=10,
    yticklabel style={text width=1.5em},
    xticklabel style={text width=2em,
      align=center},
    legend style={at={(axis cs:300,-30)},
      anchor=north west,fill=none, draw=none,legend
      cell align=left}
   ]
    \addplot [color=blue,line width=1.5pt]
     table{HRTF.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```

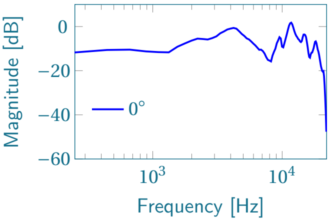
2 – Plotting data in pgfplots

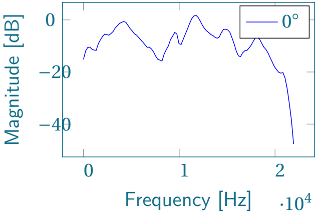




Plotting a collection of data.

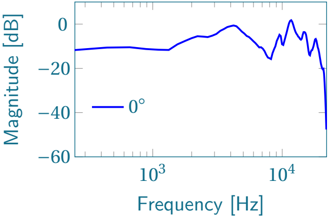
```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]},
    xmode=log,
    xmin=250,xmax=22050,
    ymin=-60,ymax=10,
    yticklabel style={text width=1.5em},
    xticklabel style={text width=2em,
      align=center},
    legend style={at={(axis cs:300,-30)},
      anchor=north west,fill=none, draw=none,legend
      cell align=left}
   ]
    \addplot [color=blue,line width=1.5pt]
      table{HRTF.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```

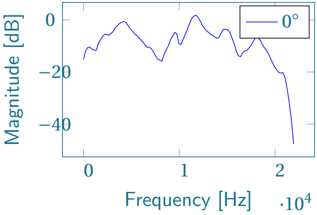




Plotting a collection of data.

```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]},
    xmode=log,
    xmin=250,xmax=22050,
    ymin=-60,ymax=10,
    yticklabel style={text width=1.5em},
    xticklabel style={text width=2em,
      align=center},
    legend style={at={(axis cs:300,-30)},
      anchor=north west,fill=none, draw=none,legend
      cell align=left}
   ]
    \addplot [color=blue,line width=1.5pt]
      table{HRTF.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```





Plotting a collection of data.

```
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[%
    width=\fwidth, height=\fheight,
    xlabel={Frequency [Hz]},
    ylabel={Magnitude [dB]},
    xmode=log,
    xmin=250,xmax=22050,
    ymin=-60,ymax=10,
    yticklabel style={text width=1.5em},
    xticklabel style={text width=2em,
      align=center},
    legend style={at={(axis cs:300,-30)},
      anchor=north west,fill=none, draw=none,legend
      cell align=left}
   ]
    \addplot [color=blue,line width=1.5pt]
      table{HRTF.tsv};
    \addlegendentry{$0^{\circ}$};
  \end{axis}
\end{tikzpicture}%
```

3 – Tools to generate/edit PGF/TikZ code

`http://www.texample.net/tikz/resources/`

Multiplatform

- `matlab2tikz` (MATLAB)
- `matplotlib2tikz` (Python)
- `tikzDevice` (R)
- Inkscape TikZ exporter
- TikzEdt

Localized

- TikziT 🍎
- Sublime Text 3 🍎
- kTikz 🍎 🐧
- TikzEdt 🐧 ⊞
- IguanaTex ⊞

Problem

⚠ Recompilation takes a long time

Problem

⚠ Recompilation takes a long time

Solution

Externalization → Saves figure in PDF

3 – Image externalization

Problem

⚠ Recompilation takes a long time

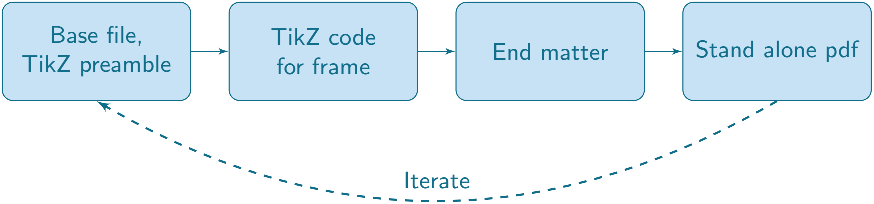
Solution

Externalization → Saves figure in PDF

```
\usepackage{pgfplots}
\usetikzlibrary{external}
\tikzexternalize[prefix=tikz/]

.
.
.

\tikzsetnextfilename{myfigurename}
\begin{tikzpicture}
```

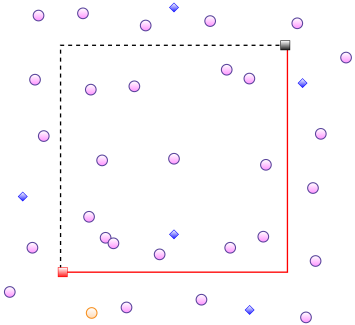
3 – Animations

- Video is worth a million words
- Can generate AVI, MPG = loss of resolution, large file size
- Use knowledge of TikZ and animate package
- Generate each frame as standalone pdf



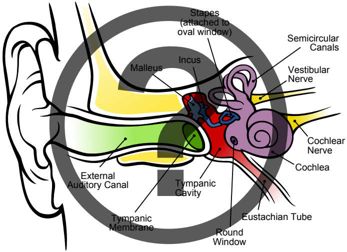
Include in Beamer

```
\animategraphics[<options>]{<frames per second>}...
<name without extension>}{<first frame>}{<last frame>}
```



Thank you for your attention

Questions?!?!



By Chittka, L. & Brockmann, A. [CC BY 2.5 (http://creativecommons.org/licenses/by/2.5)], via Wikimedia Commons

4 – Selected resources and examples

Tantau, Till
The TikZ and PGF Packages
Version 3.1.4a.

Feuersänger, Christian
Manual for Package pgfplots
Version 1.3.

Tantau, Till
The BEAMER class
User guide for version 3.36.

Jacques Crémer
A very minimal introduction to Ti*k*Z

Gérard Tisseau Jacques Duma
Ti*k*Z pour l'impatient

TEXamples
http://www.texample.net/tikz/examples/.

TEX Stackexchange
https://tex.stackexchange.com/.