PGFPlots – Plotting in LaTeX

Consistent and high-quality plotting combined with LaTeX

Christian Feuersänger

Journée GUTenberg 2012, April 14 2012

Outline

What is PGFPlots?

Visualization tool.

- motivation: user provides data + labels, pgfplots does the rest
- font consistency
- LaTeX axis descriptions
- document–wide line–styles, color schemes, markers, . . .
- high-quality
- embedded solution (no 3rd party tools)

Outline

Use–case: Scientific data analysis

- input: three data tables “results of scientific experiment”

```
dof         l2_err      level
5           8.312e-02   2
17          2.547e-02   3
49          7.407e-03   4
129         2.102e-03   5
321         5.874e-04   6
769         1.623e-04   7
1793        4.442e-05   8
4097        1.207e-05   9
9217        3.261e-06   10
```

- one table per parameter $d = 2$, $d = 3$, $d = 4$
- aim: degrees of freedom “`dof`” versus L_2 error “`l2_err`”
- target quantity: show *slope* of a line in loglog plot
  $\log e(N) = \boxed{-\alpha} \log N$

Use–case: Scientific data analysis

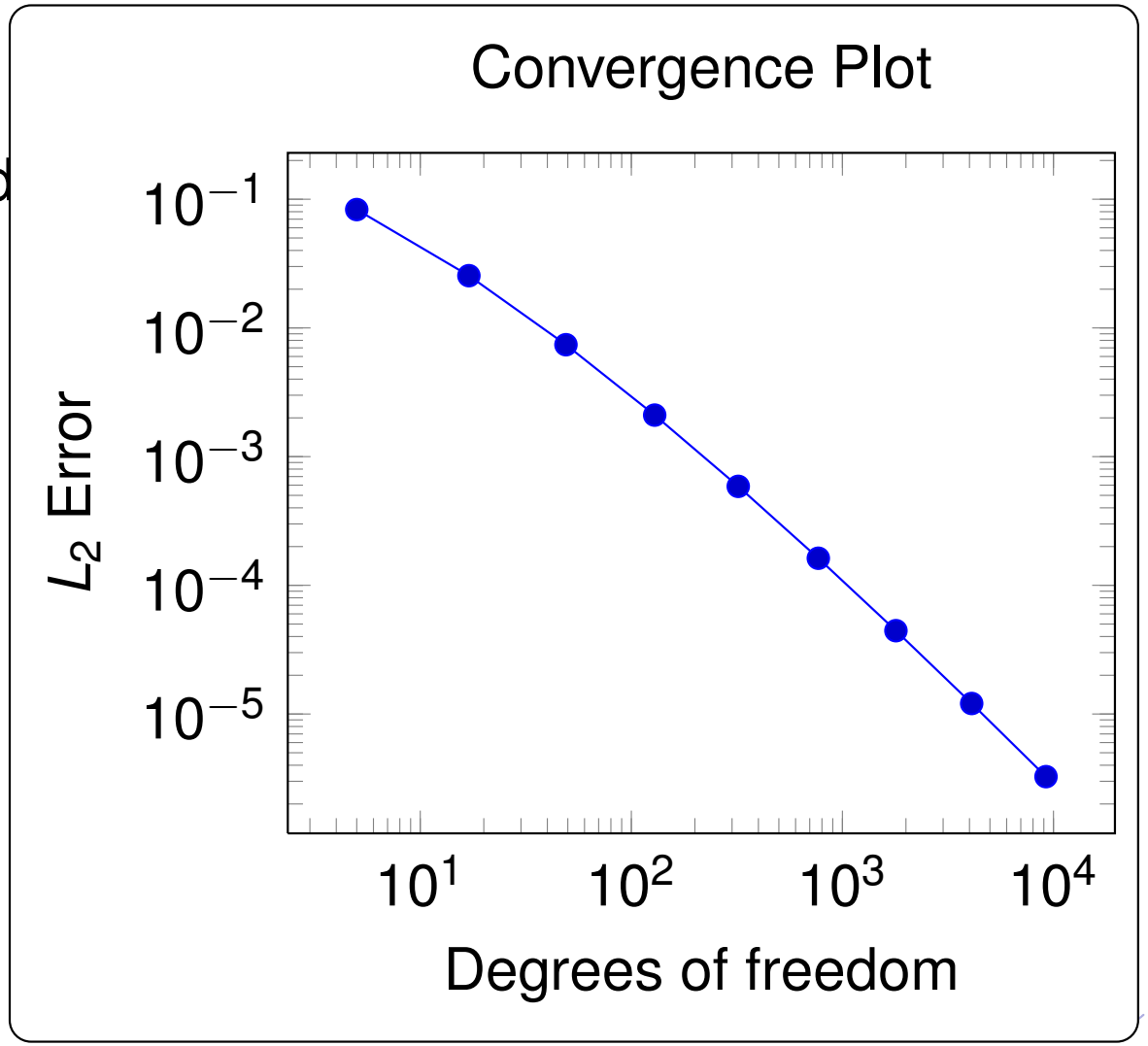
- Step 1: getting the data into TeX

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}

\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 1: getting the d

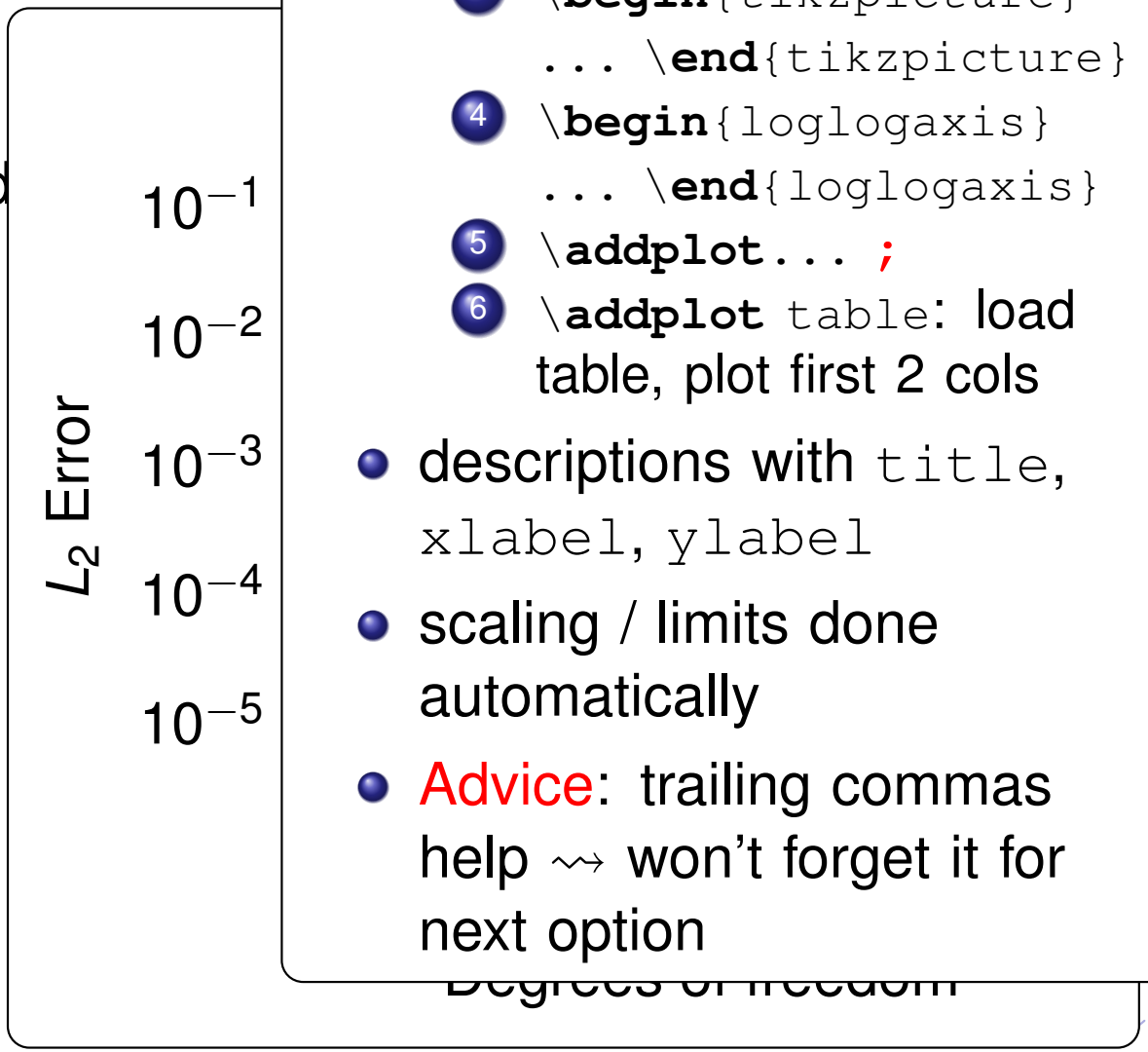
```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}

\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\end{loglogaxis}
\end{tikzpicture}
```



Use–case: Scientific data ana

- Step 1: getting the d

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}

\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\end{loglogaxis}
\end{tikzpicture}
```



- PGFPlots:
  1. `\usepackage{pgfplots}`
  2. `compat=1.5`
  3. `\begin{tikzpicture}` ... `\end{tikzpicture}`
  4. `\begin{loglogaxis}` ... `\end{loglogaxis}`
  5. `\addplot...;`
  6. `\addplot table`: load table, plot first 2 cols
- descriptions with `title`, `xlabel`, `ylabel`
- scaling / limits done automatically
- Advice: trailing commas help ⇝ won't forget it for next option

Use–case: Scientific data analysis

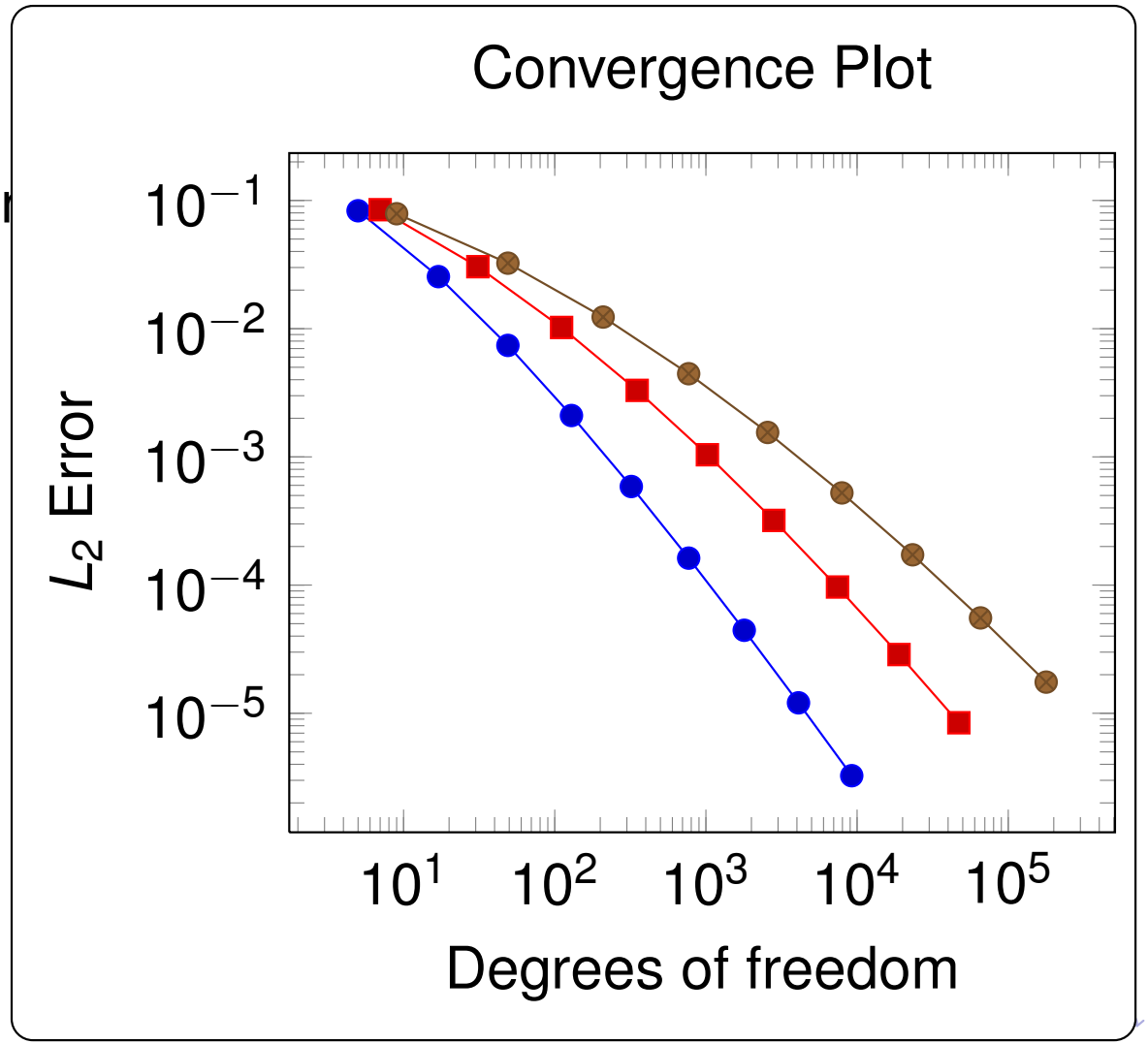
- Step 2: add remaining data files

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 2: add remainin

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

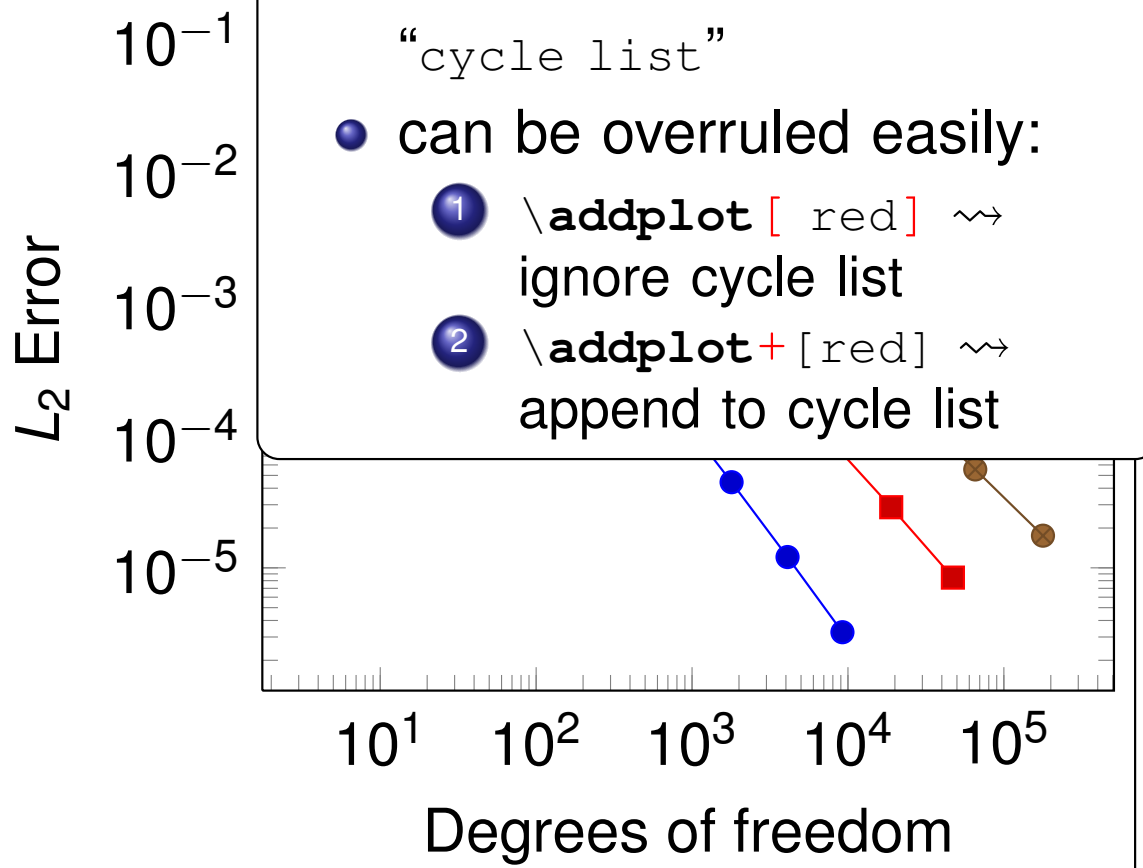


Use–case: Scientific data ana

- Step 2: add remainin

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

- multiple `\addplot` commands
- no user options ⇝ automatically chosen *styles* for each plot
  ⇝ customizable "`cycle list`"
- can be overruled easily:
  1. `\addplot[ red]` ⇝ ignore cycle list
  2. `\addplot+[red]` ⇝ append to cycle list



Use–case: Scientific data analysis

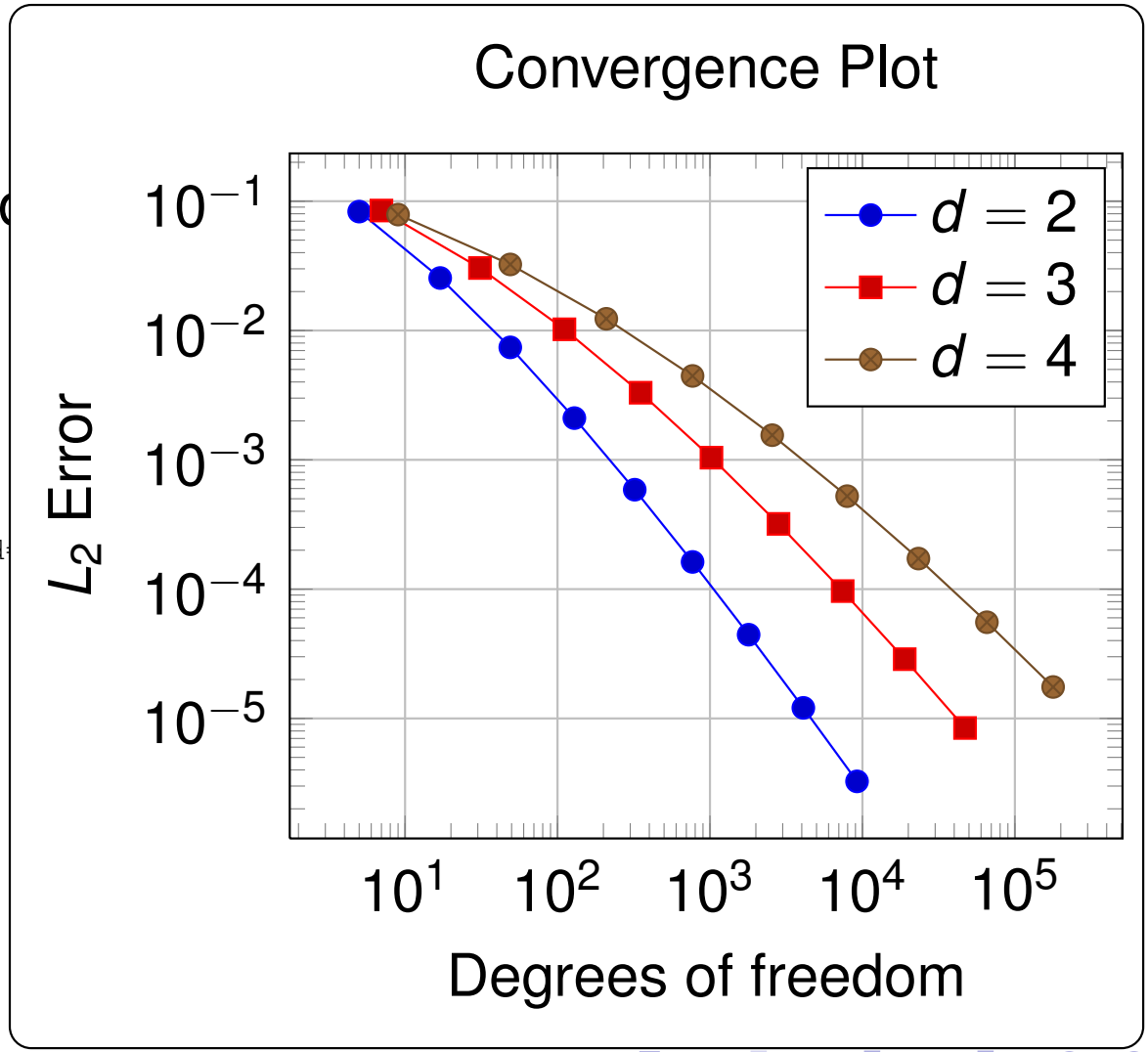
- Step 3: add a legend + grid

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 3: add a legend

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$}
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

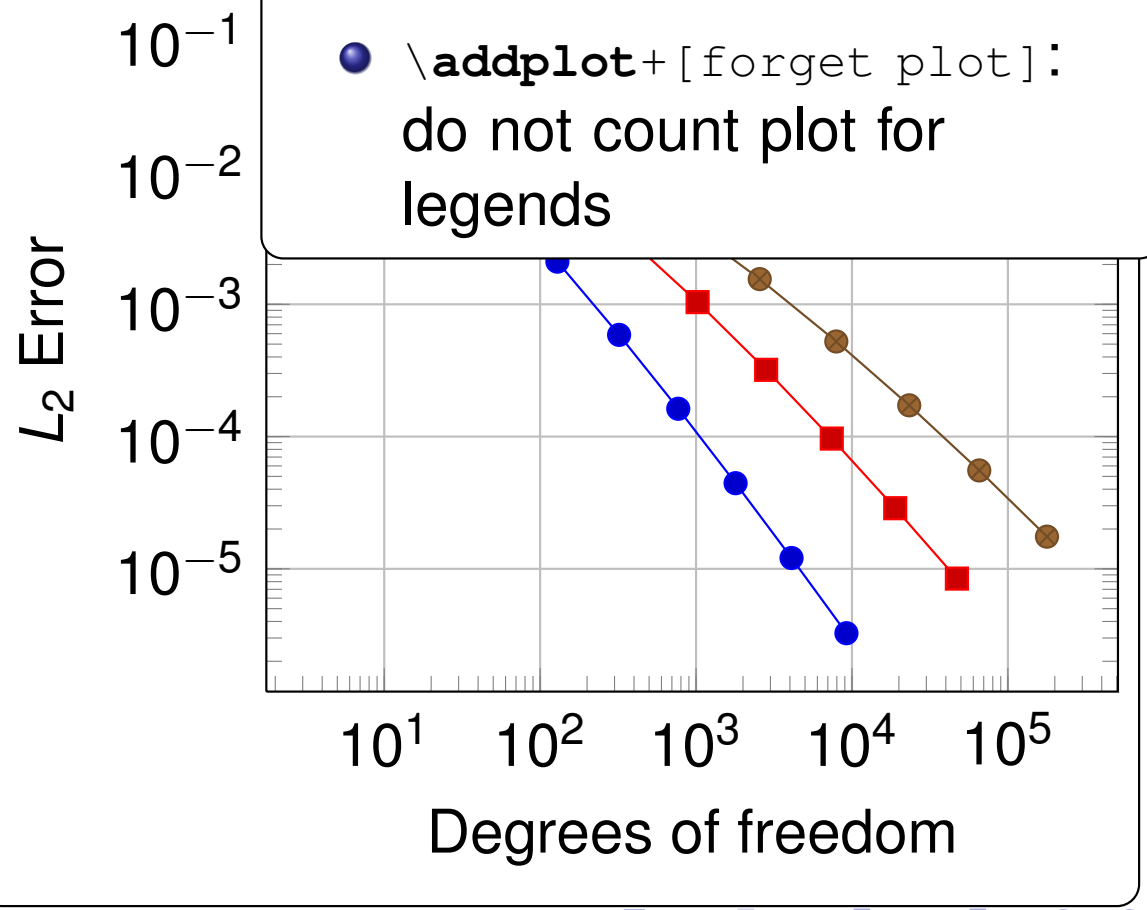


Use–case: Scientific data ana

- Step 3: add a legend

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

- `legend entries` ⇝ simplest way of getting a legend
- `grid=major|minor|both|none` (per axis: other options)
- `\addplot+[forget plot]`: do not count plot for legends



Use–case: Scientific data analysis

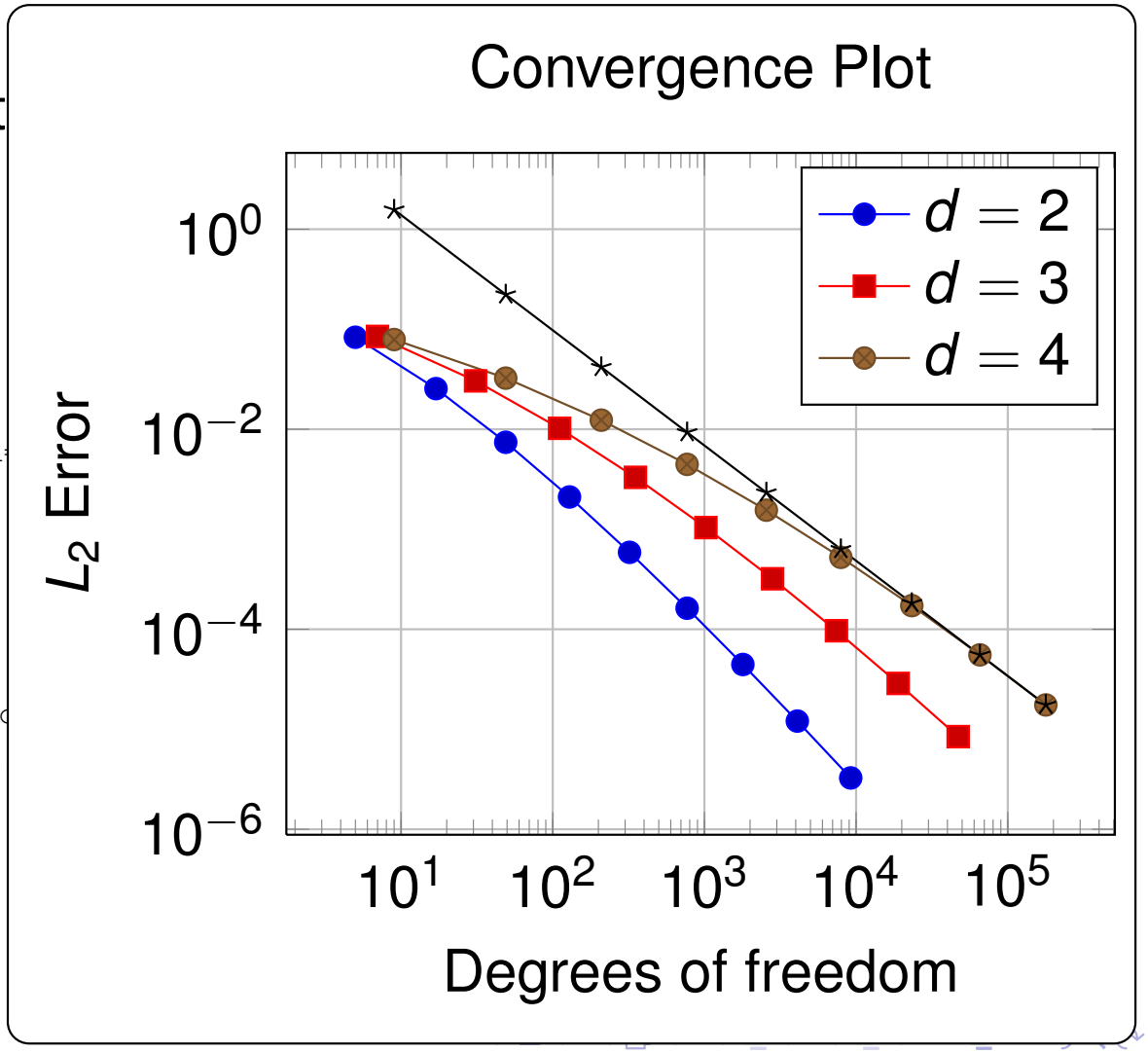
- Step 4: add a selected fit-line

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regression={y=l2_err,
      variance list={1000,800,600,500,400,200,100}}]
  {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 4: add a select

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regressio
      variance list={1000,800,600
  {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

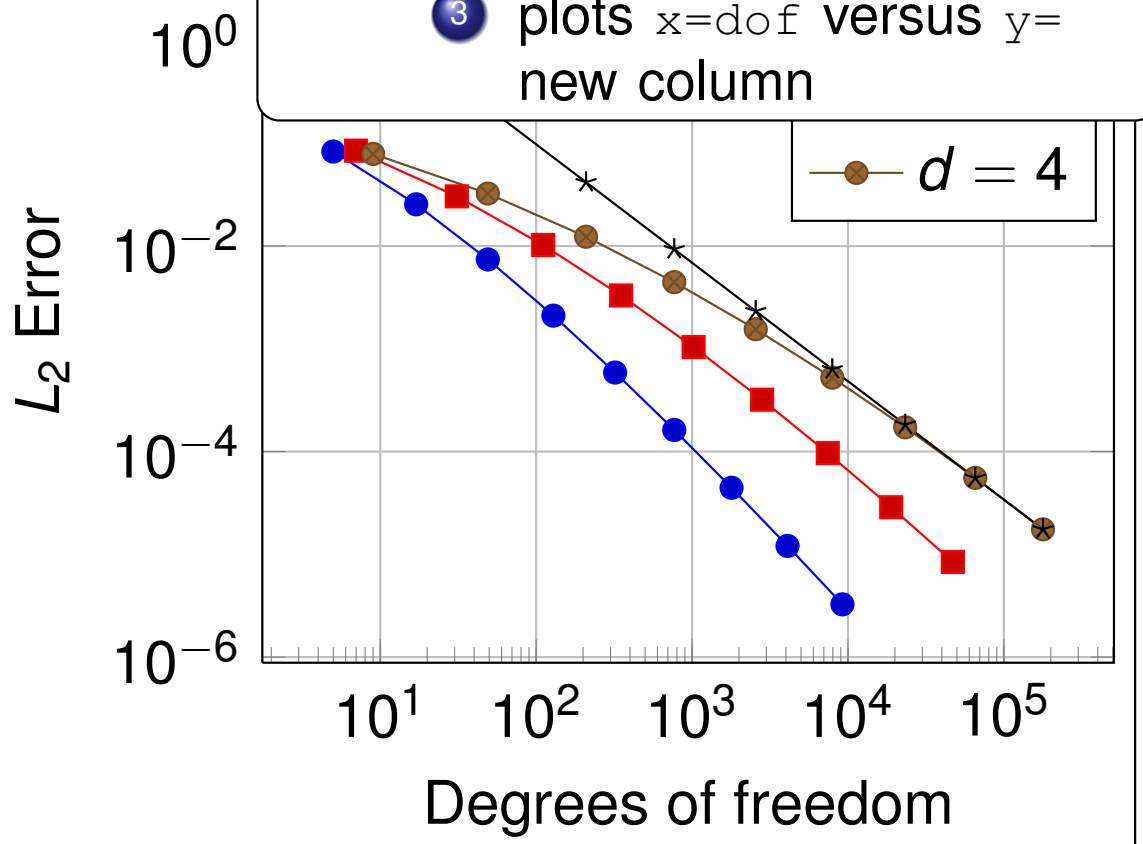


Use–case: Scientific data ana

- Step 4: add a select

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regressio
      variance list={1000,800,600
  {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

- `\addplot table` can compute new columns based on input table
- here:
  1. loads `data_d4.dat`
  2. creates new column
  3. plots `x=dof` versus `y=` new column



Use–case: Scientific data analysis

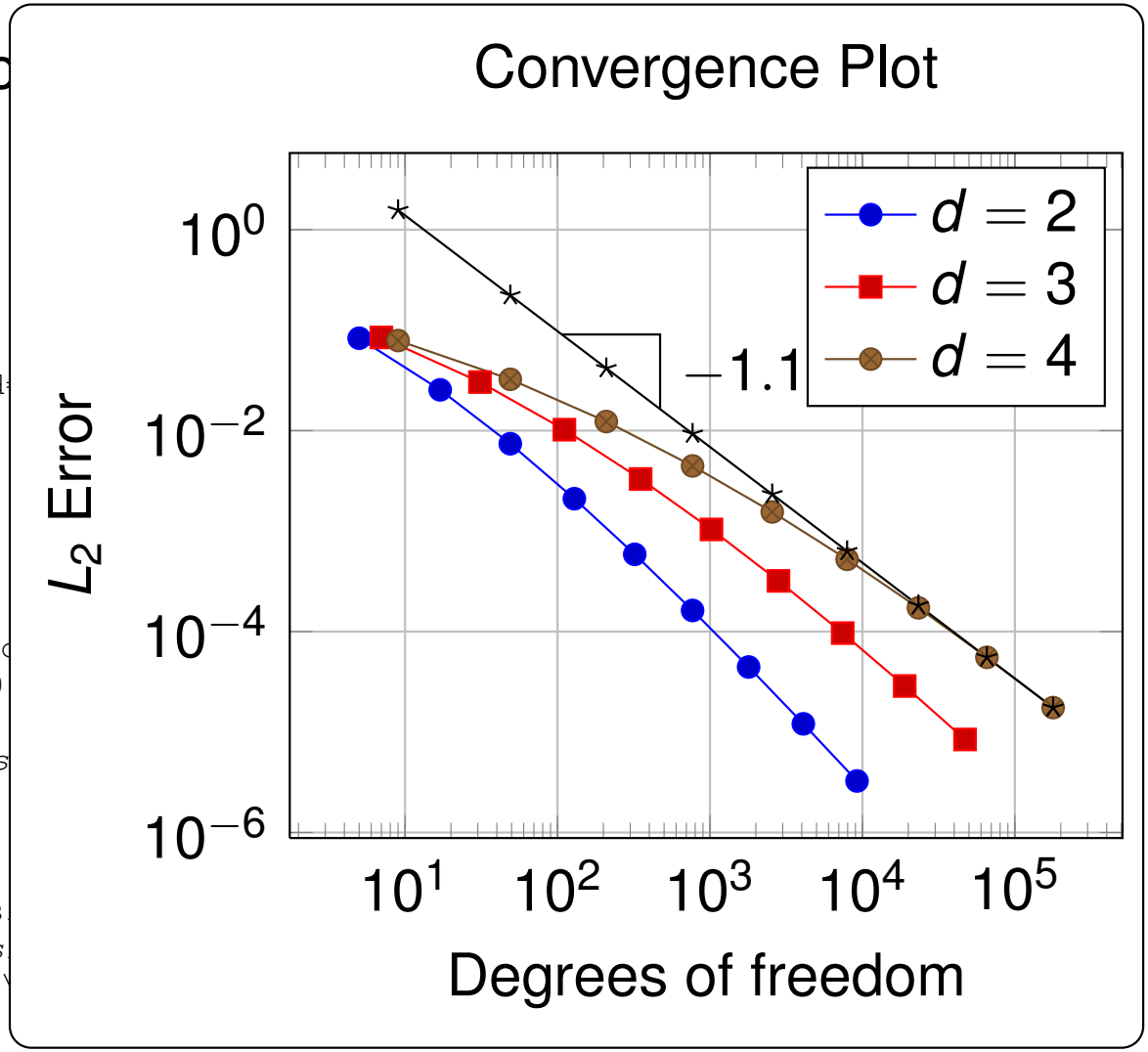
- Step 5: add an annotation using TikZ: a slope triangle

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$},
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regression={y=l2_err,
      variance list={1000,800,600,500,400,200,100}}}]
  {data_d4.dat}
  % save two points on the regression line for drawing the slope triangle
  coordinate [pos=0.25] (A)
  coordinate [pos=0.4] (B)
  ;
\xdef\slope{\pgfplotstableregressiona} % save the slope parameter
\draw (A) -| (B) % draw the opposite and adjacent sides of the triangle
    node [pos=0.75,anchor=west] {\pgfmathprintnumber{\slope}};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 5: add an anno

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$}
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regressio
      variance list={1000,800,600
  {data_d4.dat}
  % save two points on the regres
  coordinate [pos=0.25] (A)
  coordinate [pos=0.4] (B)
  ;
\xdef\slope{\pgfplotstableregress
\draw (A) -| (B) % draw the oppos
    node [pos=0.75,anchor=west] {\
\end{loglogaxis}
\end{tikzpicture}
```

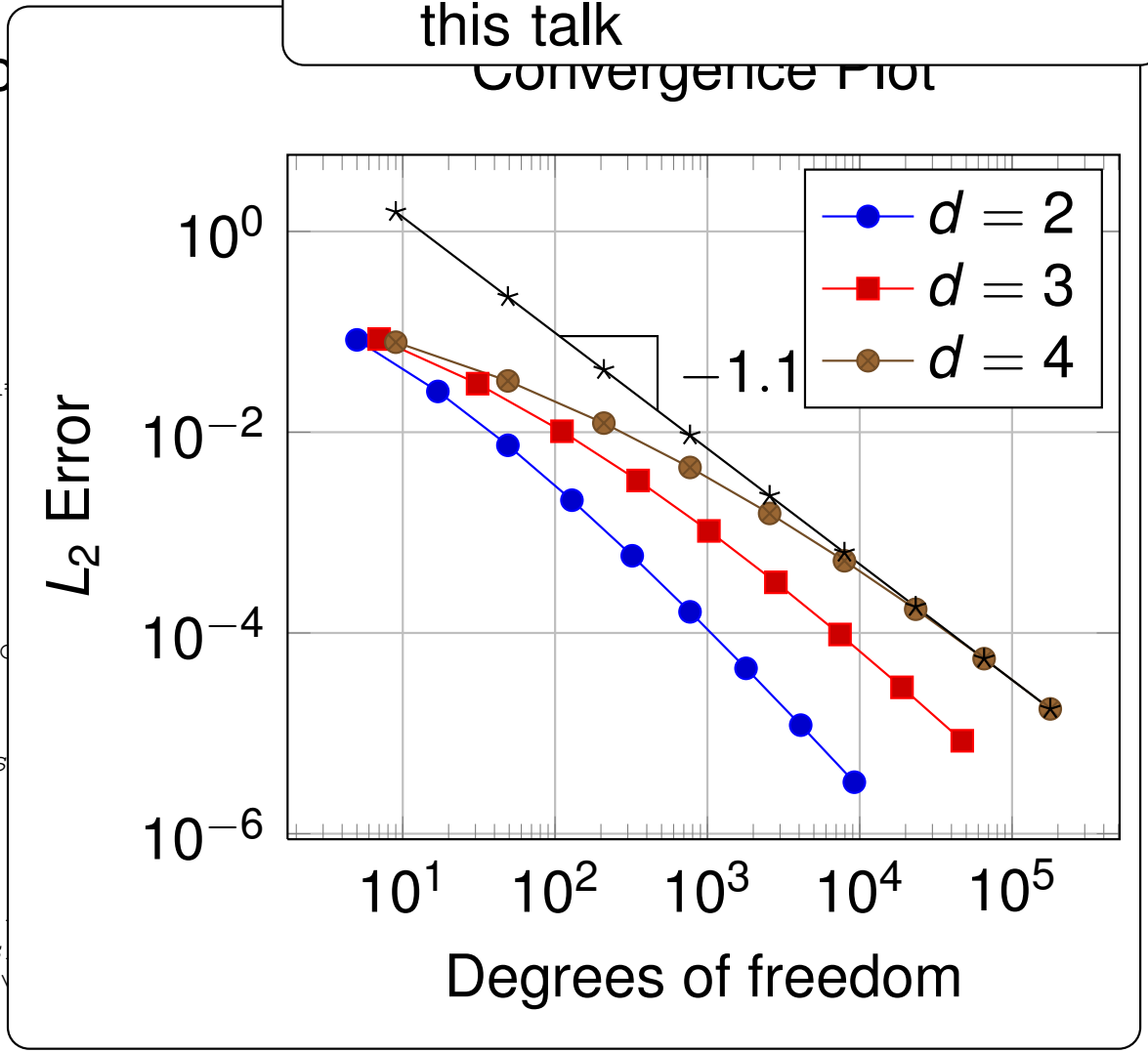


Use–case: Scientific data ana

- Simple integration of TikZ drawing instructions
- TikZ is beyond scope of this talk

- Step 5: add an anno

```
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$}
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\addplot table[
    x=dof,
    y={create col/linear regression={
      variance list={1000,800,600}}}
  {data_d4.dat}
  % save two points on the regression
  coordinate [pos=0.25] (A)
  coordinate [pos=0.4] (B)
  ;
\xdef\slope{\pgfplotstableregressionb
\draw (A) -| (B) % draw the opposite
    node [pos=0.75,anchor=west] {\
\end{loglogaxis}
\end{tikzpicture}
```



Use–case: Scientific data analysis

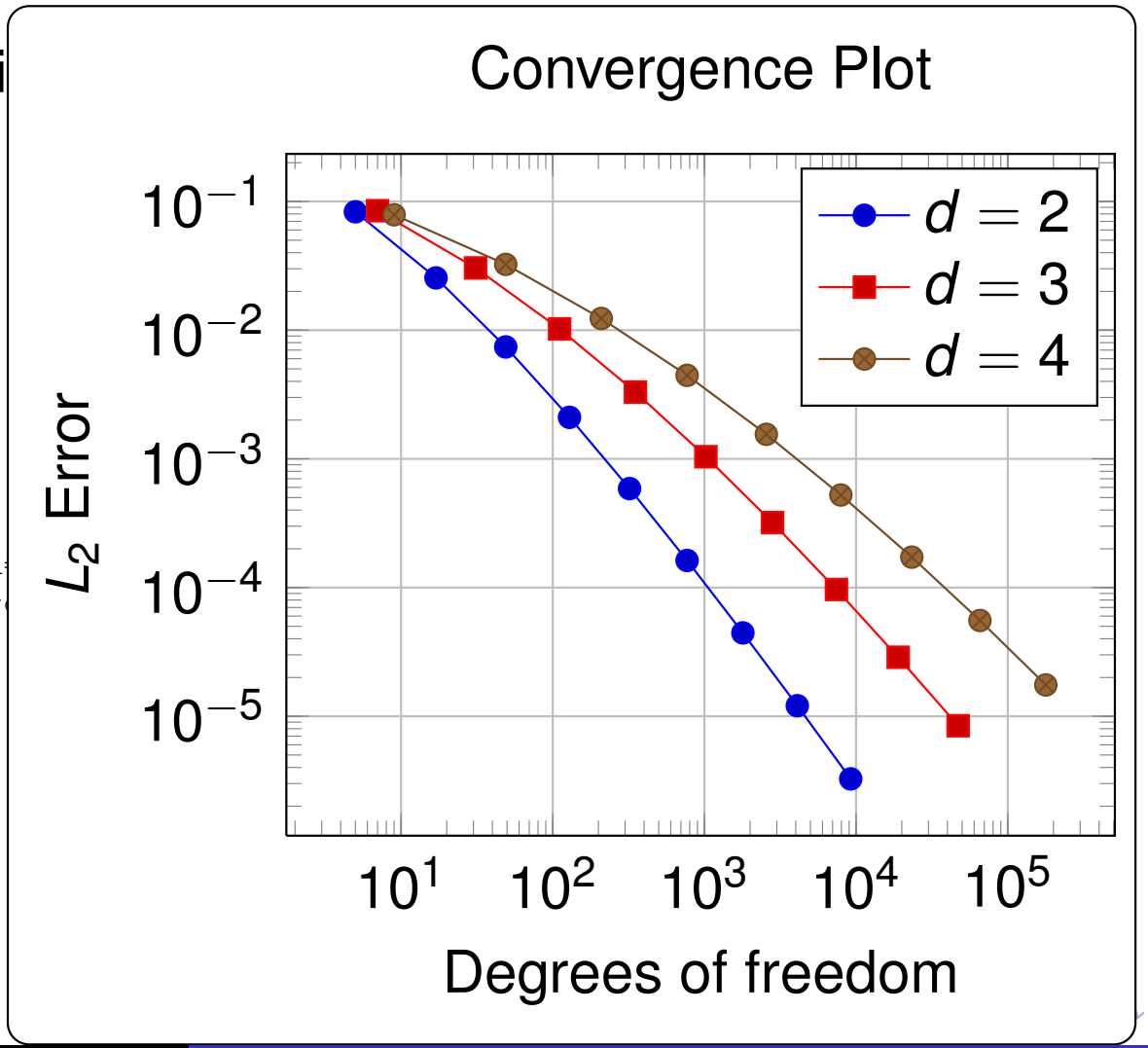
- Step 6: E-docs: enrich with optional information

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}
\usepgfplotslibrary{clickable}
...
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=4$},
   clickable coords={level \thisrow{level}},
   clickable coords size=5,
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

Use–case: Scientific data analysis

- Step 6: E-docs: enri

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}
\usepgfplotslibrary{clickable}
...
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=
   clickable coords={level \thisr
   clickable coords size=5,
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

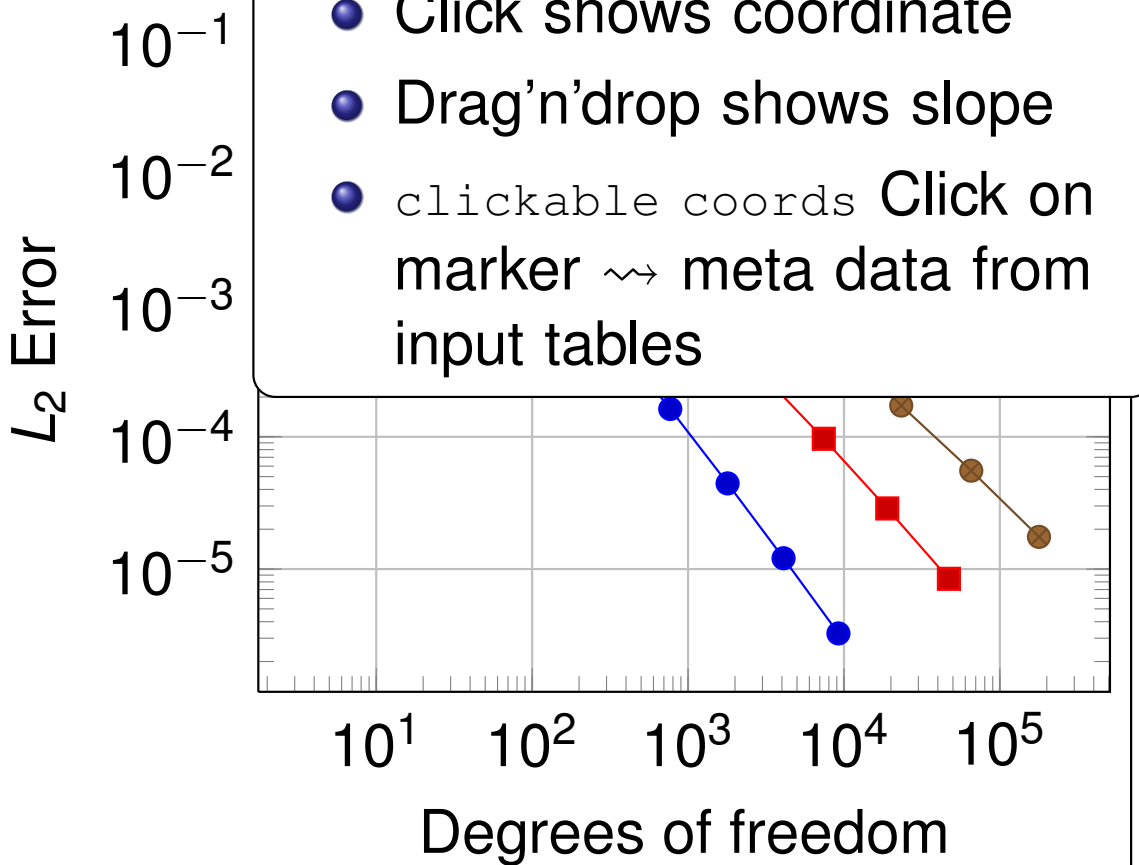


Use–case: Scientific data ana

- Step 6: E-docs: enri

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.5}
\usepgfplotslibrary{clickable}
...
\begin{tikzpicture}
\begin{loglogaxis}[
   title=Convergence Plot,
   xlabel={Degrees of freedom},
   ylabel={$L_2$ Error},
   grid=major,
   legend entries={$d=2$,$d=3$,$d=
   clickable coords={level \thisr
   clickable coords size=5,
]
\addplot table {data_d2.dat};
\addplot table {data_d3.dat};
\addplot table {data_d4.dat};
\end{loglogaxis}
\end{tikzpicture}
```

- `\usepgfplotslibrary{clickable}`
- Acrobat Reader + Javascript
- Click shows coordinate
- Drag'n'drop shows slope
- `clickable coords` Click on marker $\rightsquigarrow$ meta data from input tables



Outline

Use–case: Function visualization

- Task: side-by-side visualization of
  1. function from data file
  2. related function by expression

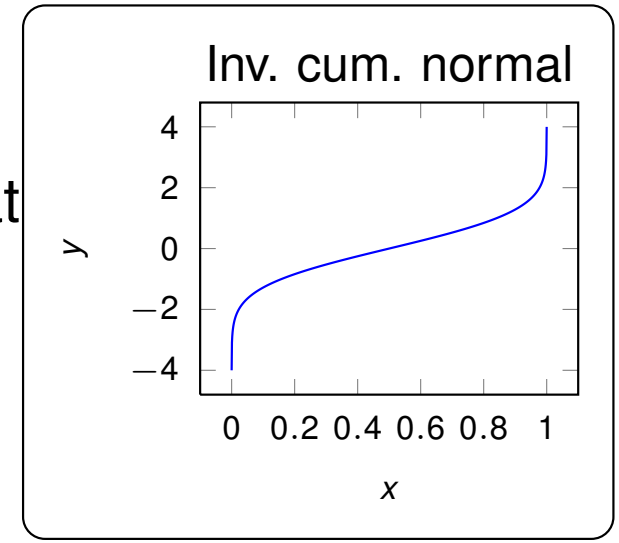
Use–case: Function visualization

- Step 1: function from data file

```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- Step 1: function from dat

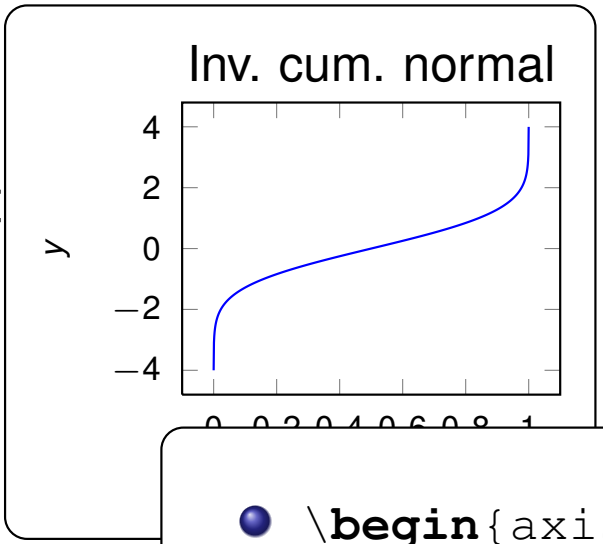



```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- Step 1: function from dat

```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```



- `\begin{axis}..\end{axis}` ⇝ linear scale
- predefined style `tiny` ⇝ standard TEX fonts, no text scaling
- ignore `cycle list`; use `[blue]`

Use–case: Function visualization

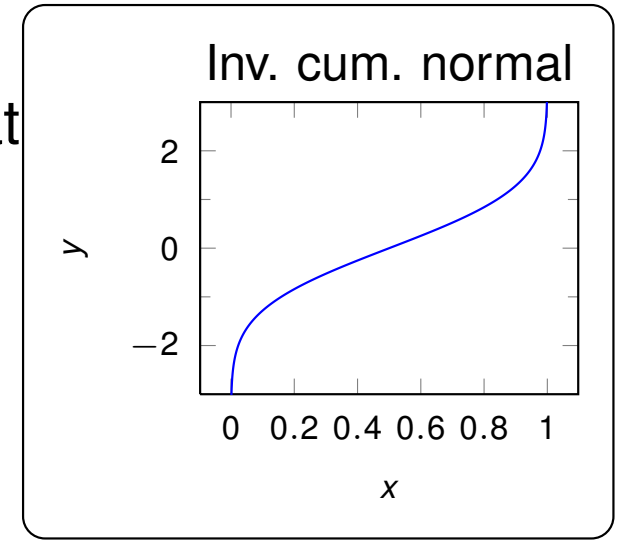
- Step 1: function from data file. Fine tuning

```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- Step 1: function from dat

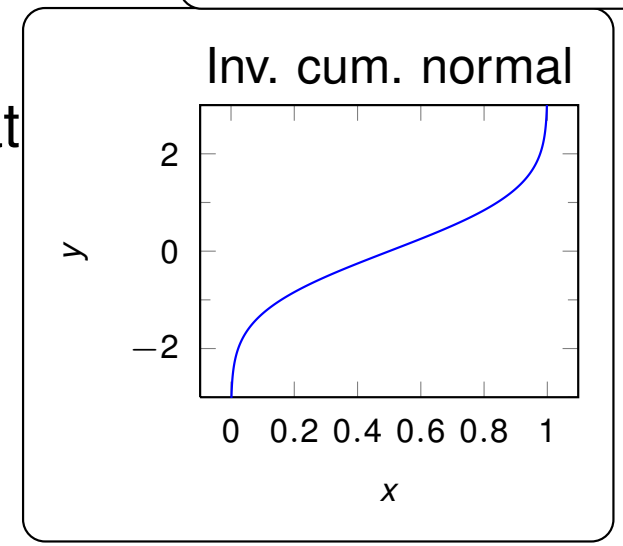
```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```



Use–case: Function visualiza

- adjust displayed limits
- adjust minor tick count

- Step 1: function from dat




```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

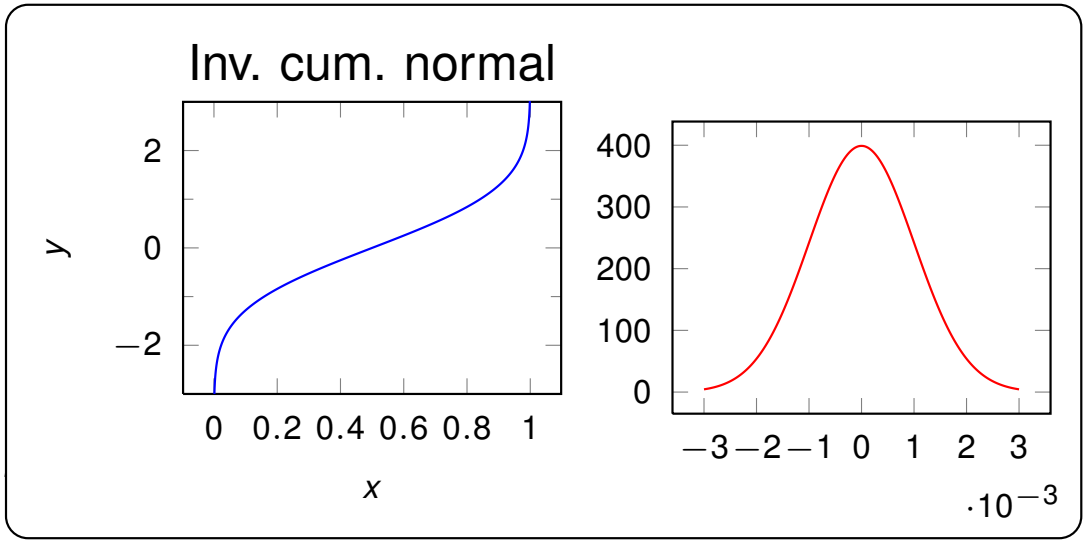
- Step 2: add 2nd plot: function from expression

```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}%<- avoid white space
%
\begin{tikzpicture}
\begin{axis}[
   tiny,
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- Step 2: add 2nd plot: function from expression

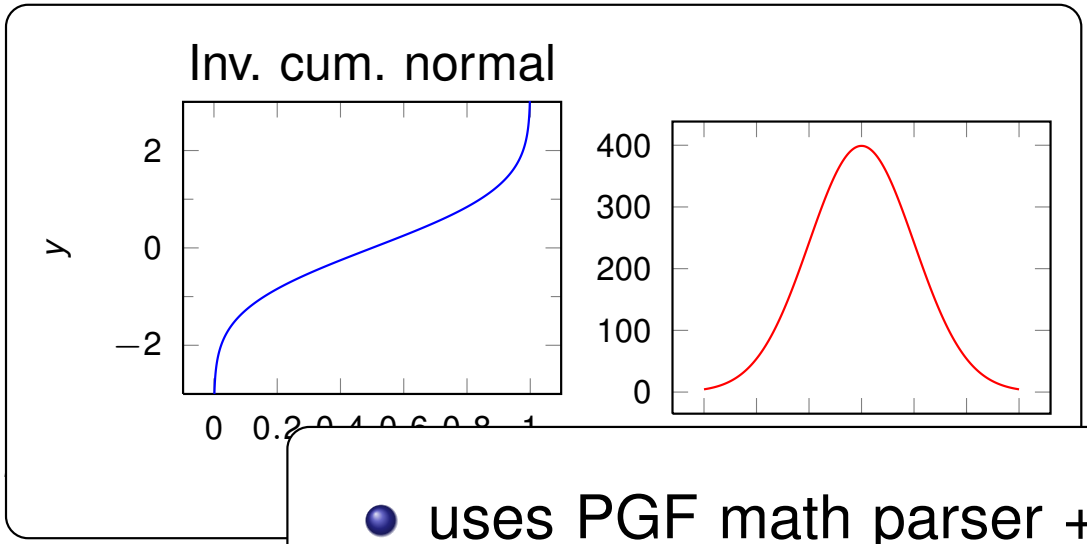
```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}%<- avoid white
%
\begin{tikzpicture}
\begin{axis}[
   tiny,
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```



Use–case: Function visualization

- Step 2: add 2nd plot: function from expression

```
\begin{tikzpicture}
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}%<- avoid white
%
\begin{tikzpicture}
\begin{axis}[
   tiny,
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=2
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * s
\end{axis}
\end{tikzpicture}
```



- uses PGF math parser + floating point unit
- `\newcommand` ⇝ “constants”
- `domain=<a>:<b>, samples=<N>`
- different bounding boxes!

Use–case: Function visualization

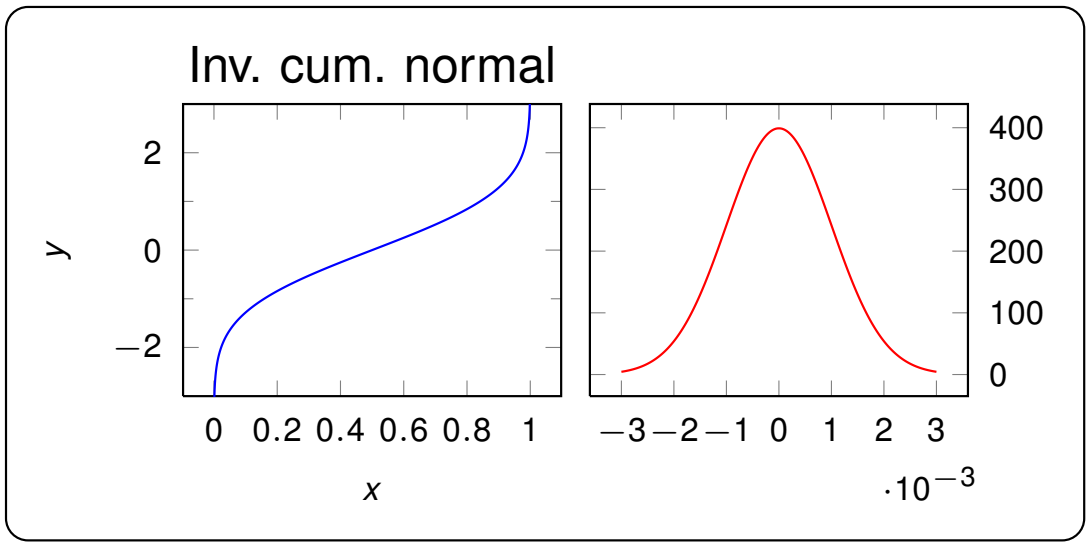
- Step 3: fix up vertical alignment + tick label positions

```
\begin{tikzpicture}[baseline]
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}%
%
\begin{tikzpicture}[baseline]
\begin{axis}[
   tiny,
   yticklabel pos=right,
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- Step 3: fix up vertical alignment + tick label positions

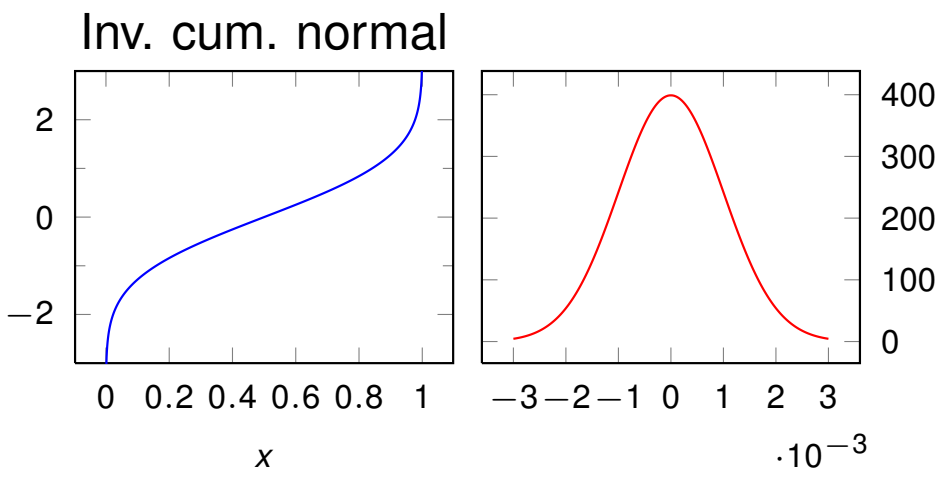
```
\begin{tikzpicture}[baseline]
\begin{axis}[
   title=Inv. cum. normal,
   xlabel={$x$},
   ylabel={$y$},
   tiny,
   ymin=-3, ymax=3,
   minor y tick num=1,
]
\addplot[blue] table {invcum.dat};
\end{axis}
\end{tikzpicture}%
%
\begin{tikzpicture}[baseline]
\begin{axis}[
   tiny,
   yticklabel pos=right,
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```



Use–case: Function visualization

- Step 3: fix up vertical alignment + tick label positions

- `\begin{tikzpicture}[baseline]` ⇝ same baseline (lower edge)
- similarly:
  1. `\begin{tikzpicture}[trim axis left]`
  2. `\begin{tikzpicture}[trim axis right]`
- `yticklabel pos=right` switches sides




```
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

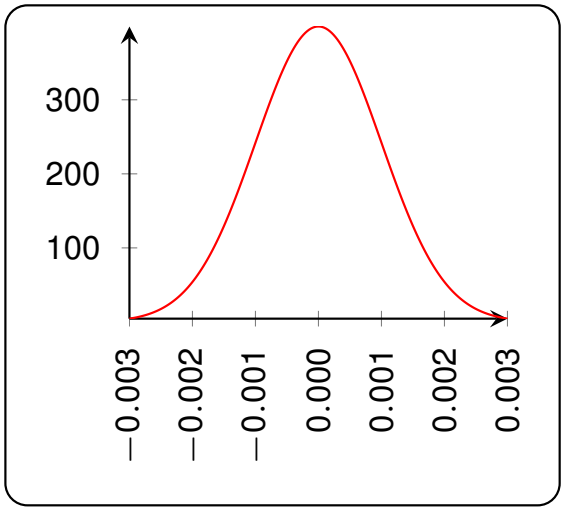
- On commonly asked questions... style adjustments.

```
\begin{tikzpicture}
\begin{axis}[
   tiny,
   axis lines=left,
   scaled ticks=false,
   xticklabel style={
      rotate=90,
      anchor=east,
      /pgf/number format/precision=3,
      /pgf/number format/fixed,
      /pgf/number format/fixed zerofill},
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```

Use–case: Function visualization

- On commonly asked questions... style adjustments.

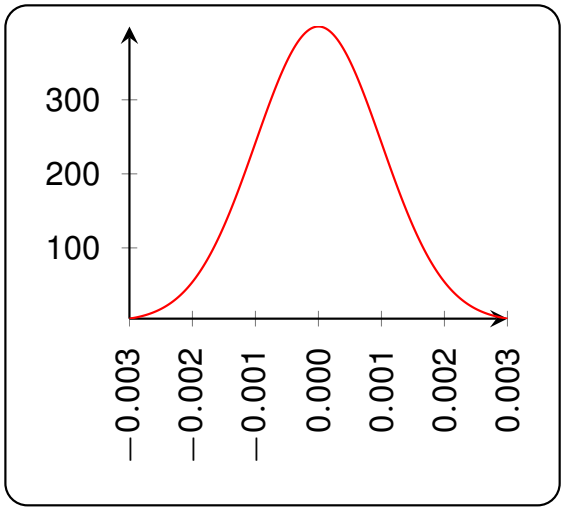
```
\begin{tikzpicture}
\begin{axis}[
   tiny,
   axis lines=left,
   scaled ticks=false,
   xticklabel style={
      rotate=90,
      anchor=east,
      /pgf/number format/precision=3,
      /pgf/number format/fixed,
      /pgf/number format/fixed zerofill},
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```



Use–case: Function visualization

- `scaled ticks` ⇝ factor out common tick factors
- `/pgf/number format/...` ⇝ number printing
- `axis lines=...` ⇝ non-boxed axes

estions... style adjustments.




```
        /pgf/number format/precision=3,
        /pgf/number format/fixed,
        /pgf/number format/fixed zerofill},
]
% density of Normal distribution:
\newcommand\MU{0}
\newcommand\SIGMA{1e-3}
\addplot[red,domain=-3*\SIGMA:3*\SIGMA,samples=201]
   {exp(-(x-\MU)^2 / 2 / \SIGMA^2) / (\SIGMA * sqrt(2*pi))};
\end{axis}
\end{tikzpicture}
```

Outline

Scatter plot Use–case A

- Step 1: simple scatter plot
- Input: data file with coords + meta data (table format)

```
# ordering=colwise,  basis=BASIS_HAT_HIER, number points=1657, (max)level(s): (7,8), do
# x_i : coordinates; (l_i,i_i): multiindex coords; scalaridx_i: linearized multiindex c
x_0             x_1             f(x)              l_0             i_0             scalaridx_0
0               0               0                 0               0               0
0               0.024543693     -0.00054758064    0               0               0
0               0.049087385     -0.0021903226     0               0               0
⋮
```

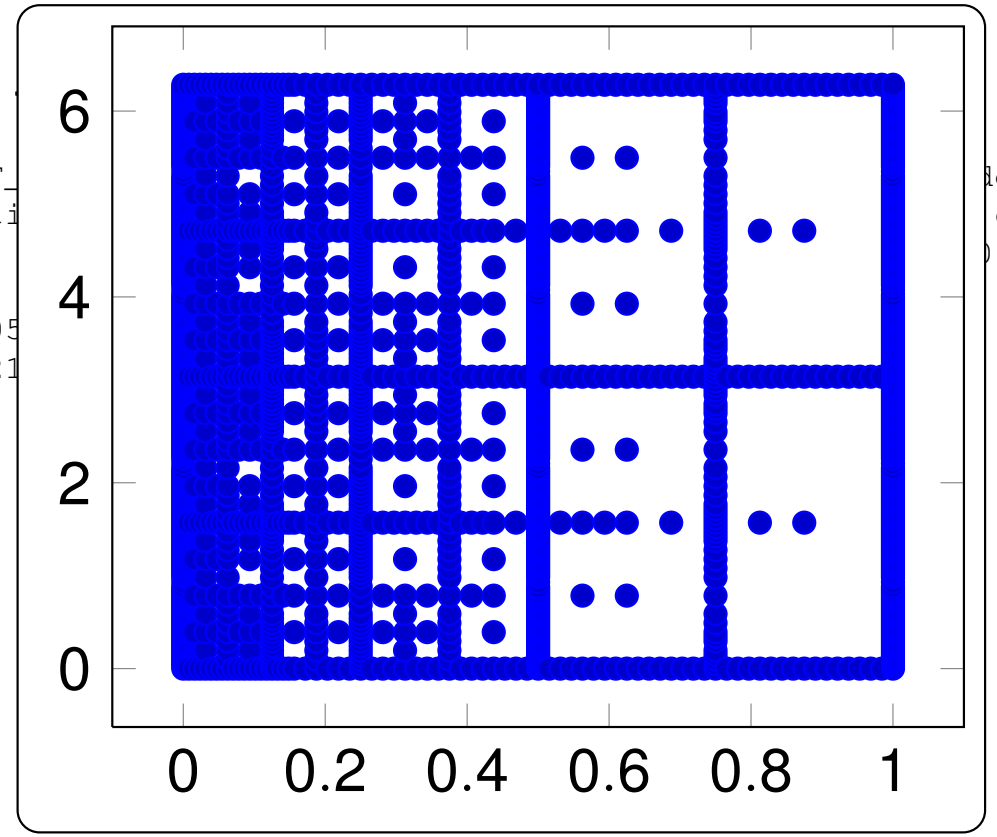
```
\begin{tikzpicture}
\begin{axis}
\addplot+[only marks] table
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```

Scatter plot Use–case A

- Step 1: simple scatter plot
- Input: data file with coords

```
# ordering=colwise,  basis=BASIS_HAT_
# x_i : coordinates; (l_i,i_i): multi
x_0             x_1             f(x)
0               0               0
0               0.024543693     -0.0005
0               0.049087385     -0.0021
⋮
```

```
\begin{tikzpicture}
\begin{axis}
\addplot+[only marks] table
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```

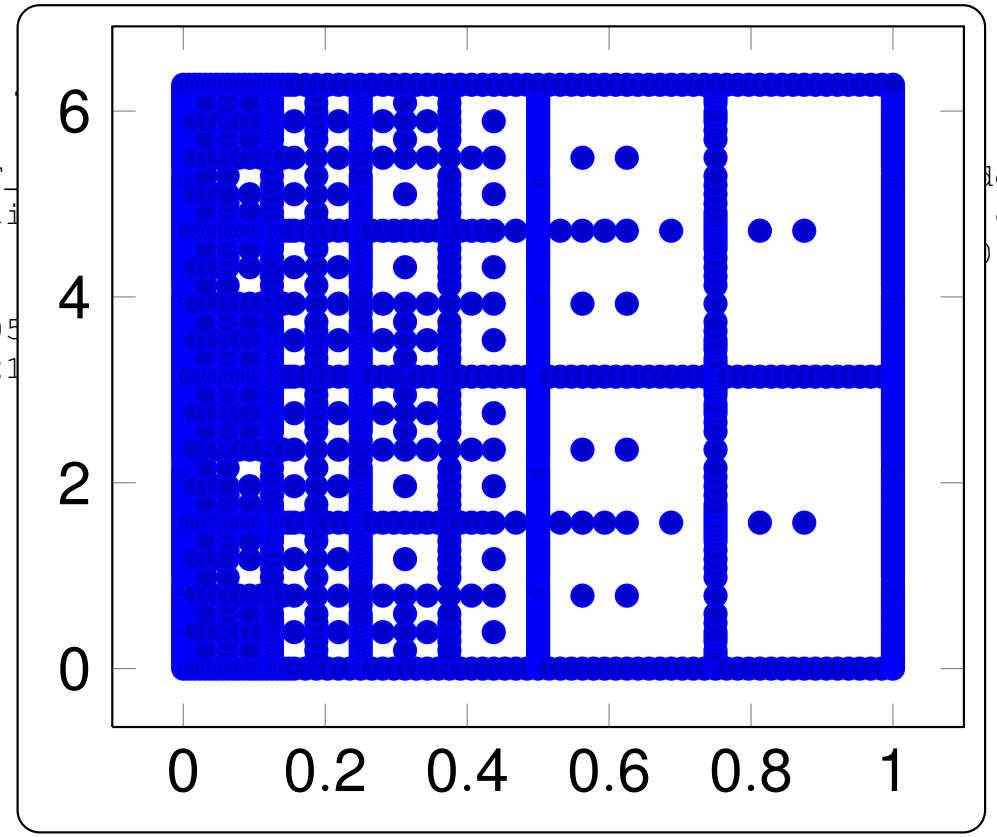


Scatter plot Use–case A

- `\addplot+` ⇝ options of `cycle list` + `only marks`
- `table` ⇝ plot first two cols

```
# ordering=colwise,  basis=BASIS_HAT_
# x_i : coordinates; (l_i,i_i): multi
x_0             x_1             f(x)
0               0               0
0               0.024543693     -0.0005
0               0.049087385     -0.0021
⋮
```

```
\begin{tikzpicture}
\begin{axis}
\addplot+[only marks] table
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



Scatter plot Use–case A

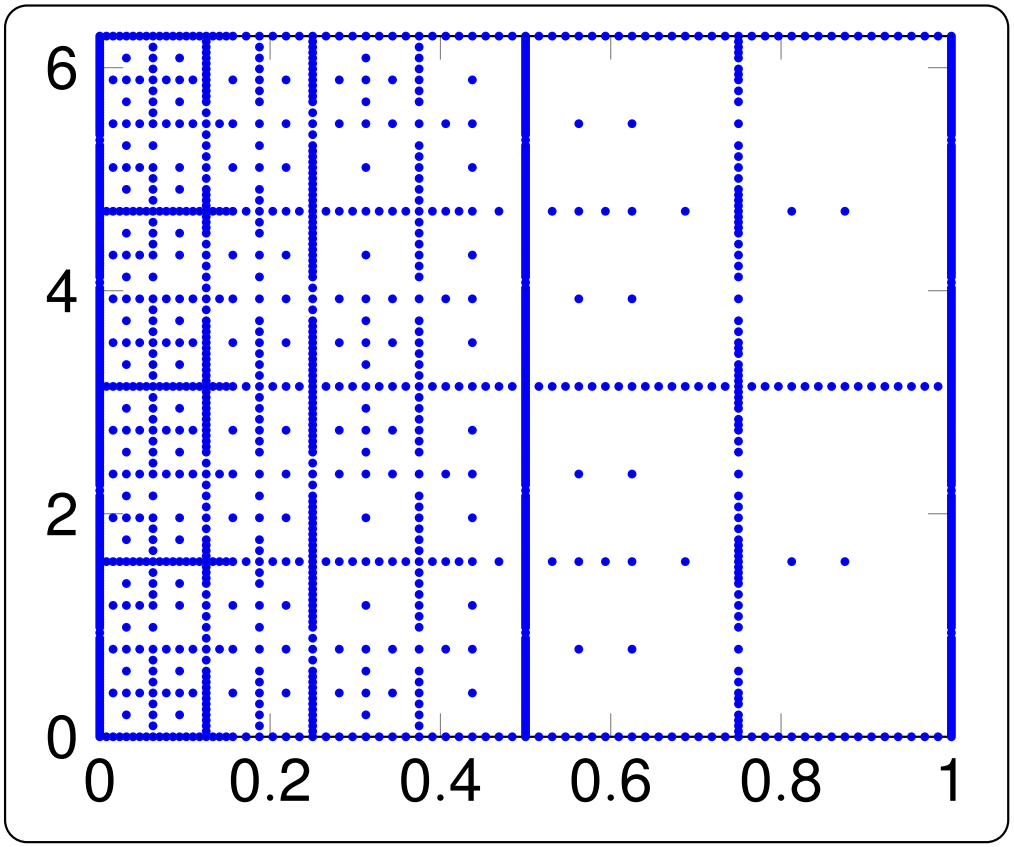
- Step 2: fine tuning

```
\begin{tikzpicture}
\begin{axis}[
   enlargelimits=false,
]
\addplot+[only marks,mark size=0.6pt]
   table {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```

Scatter plot Use–case A

- Step 2: fine tuning

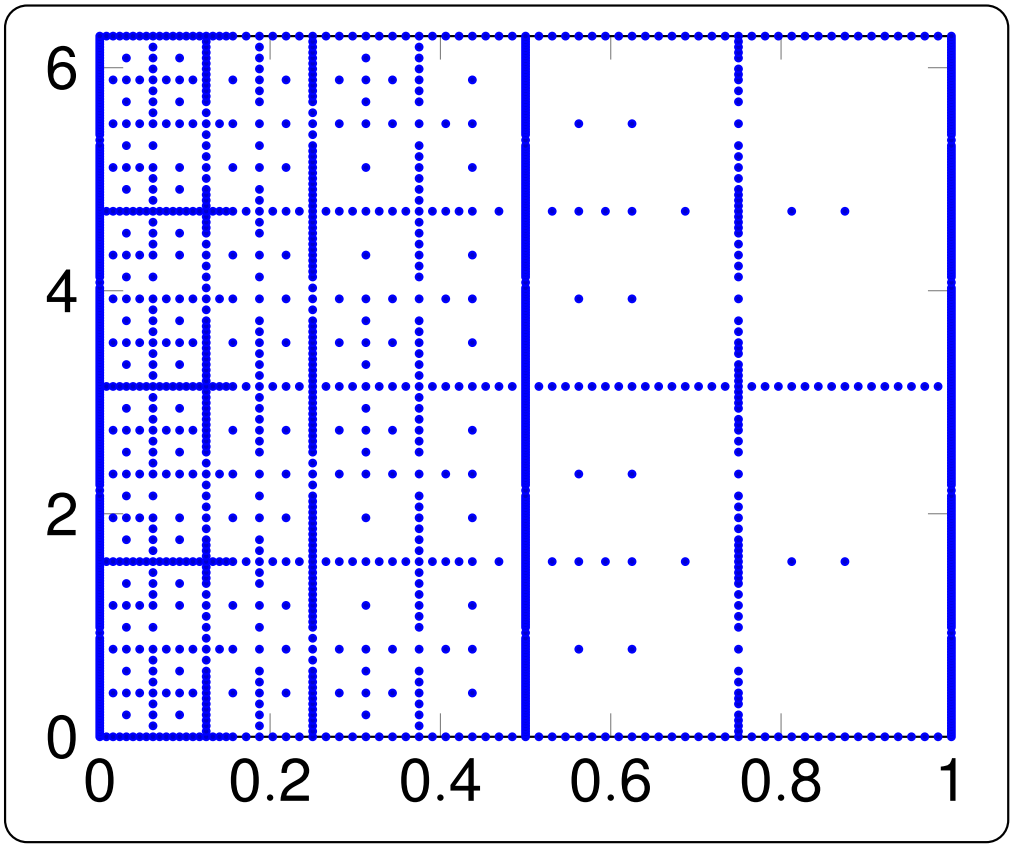
```
\begin{tikzpicture}
\begin{axis}[
   enlargelimits=false,
]
\addplot+[only marks,mark size=0.6pt]
   table {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



Scatter plot Use–case A

- modify marker size
- small limits

```
    enlargelimits=false,
]
\addplot+[only marks,mark size=0.6pt]
    table {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



Scatter plot Use–case A

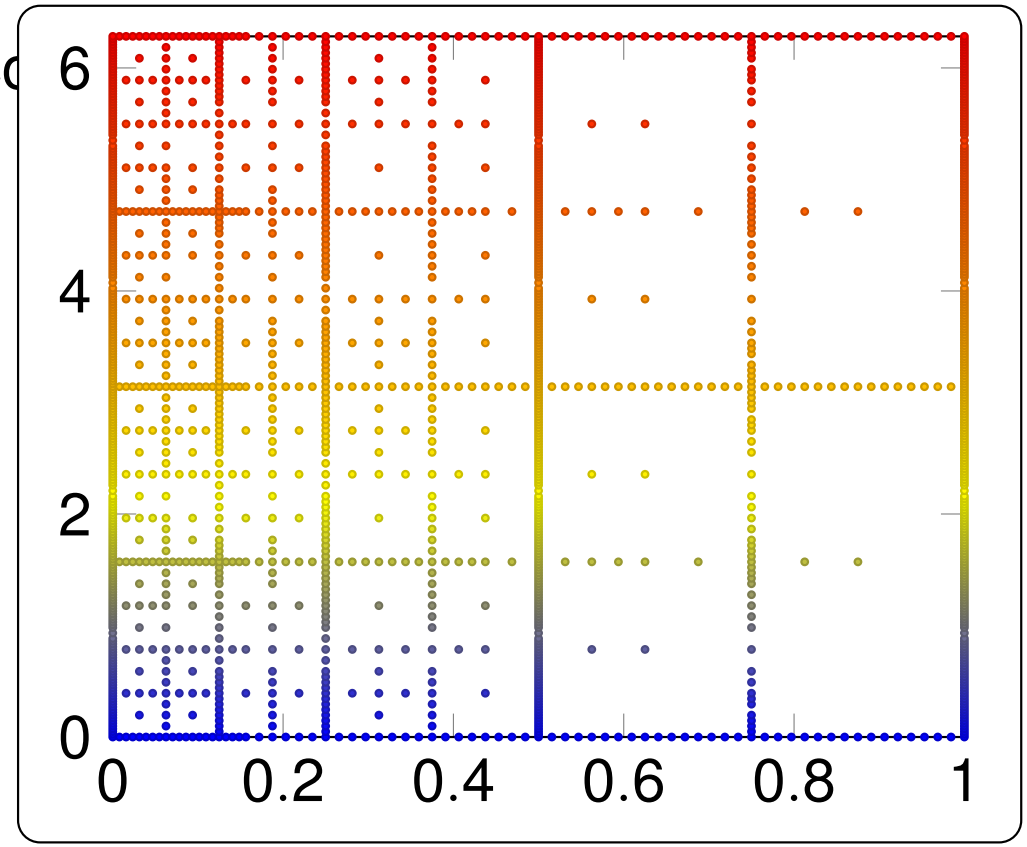
- Step 3: color coding according to meta data

```
\begin{tikzpicture}
\begin{axis}[
   enlargelimits=false,
]
\addplot+[only marks,scatter,
   mark size=0.6pt
]
table[meta=f(x)]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```

Scatter plot Use–case A

- Step 3: color coding acc

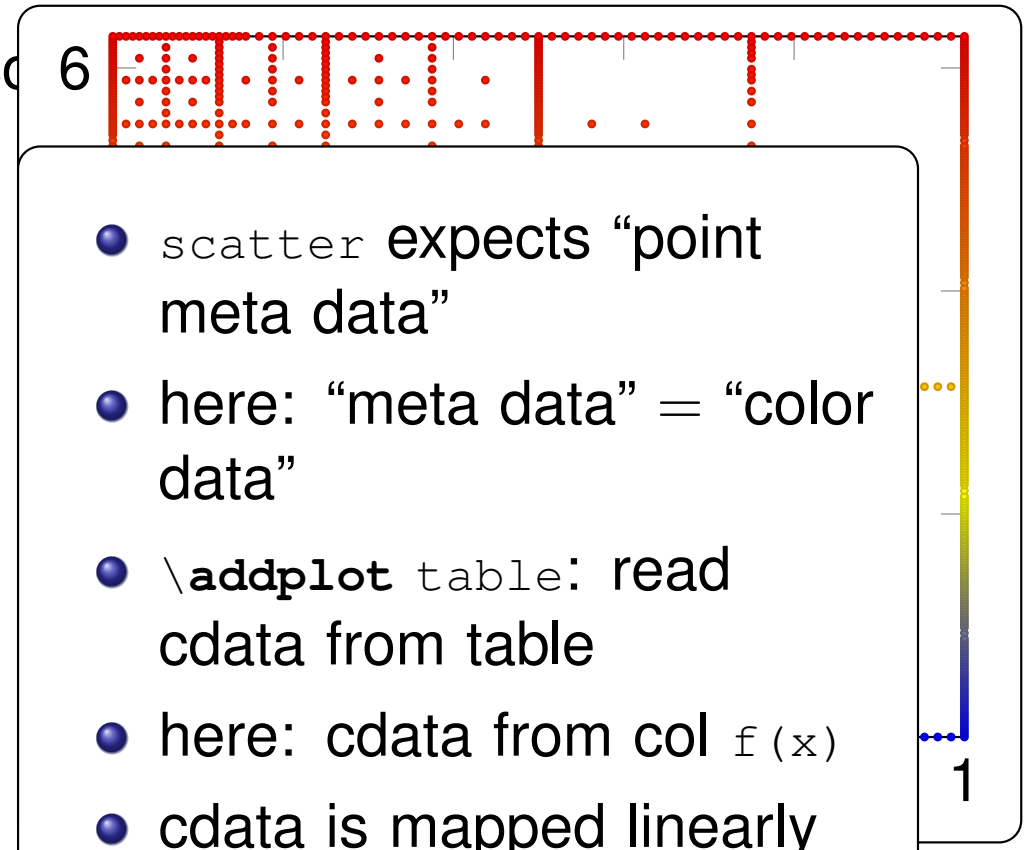
```
\begin{tikzpicture}
\begin{axis}[
   enlargelimits=false,
]
\addplot+[only marks,scatter,
   mark size=0.6pt
]
table[meta=f(x)]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



Scatter plot Use–case A

- Step 3: color coding acc

```
\begin{tikzpicture}
\begin{axis}[
   enlargelimits=false,
]
\addplot+[only marks,scatter,
   mark size=0.6pt
]
table[meta=f(x)]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



- `scatter` expects “point meta data”
- here: “meta data” = “color data”
- `\addplot table`: read cdata from table
- here: cdata from col `f(x)`
- cdata is mapped linearly into a `colormap`

Scatter plot Use–case B

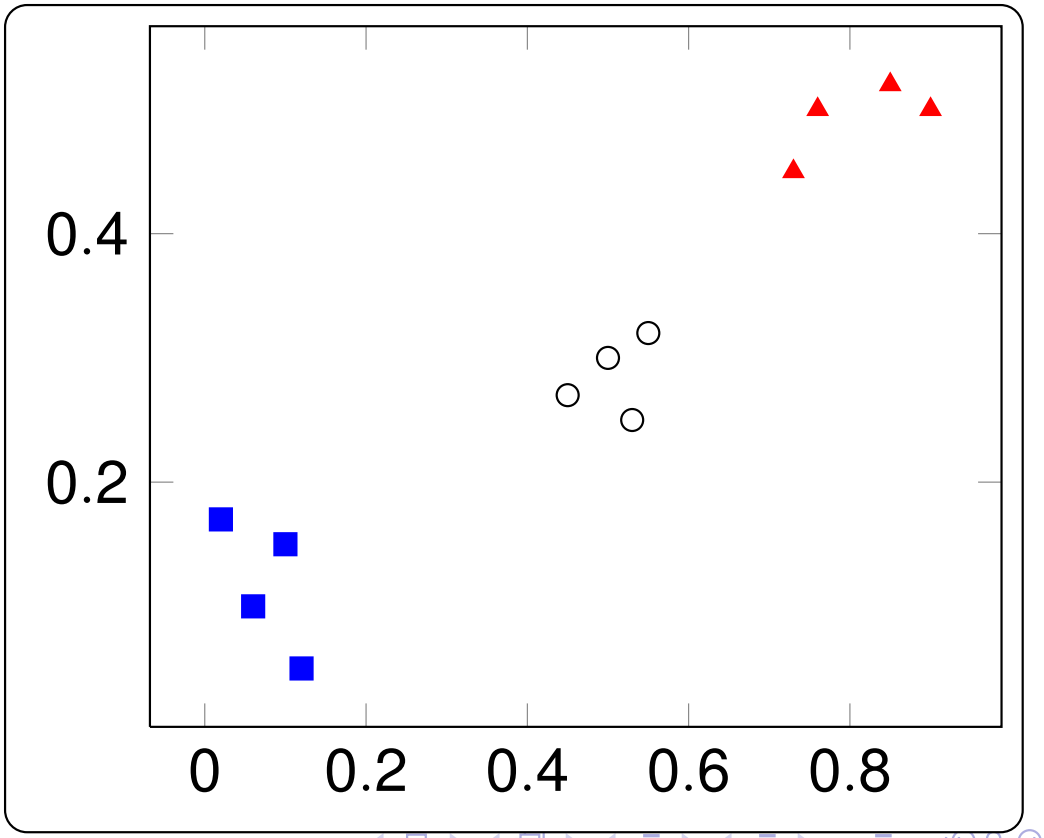
- scatter plot with class labels per point

```
\begin{tikzpicture}
\begin{axis}[
    scatter/classes={
      a={mark=square*,blue},%
      b={mark=triangle*,red},%
      c={mark=o,draw=black}}]
\addplot[scatter,only marks,
    scatter src=explicit symbolic]
table {
    x       y         label
    0.1     0.15      a
    0.45    0.27      c
    0.02    0.17      a
    0.06    0.1       a
    0.9     0.5       b
    0.5     0.3       c
    0.85    0.52      b
    0.12    0.05      a
    0.73    0.45      b
    0.53    0.25      c
    0.76    0.5       b
    0.55    0.32      c
};
\end{axis}
\end{tikzpicture}
```

Scatter plot Use–case B

- scatter plot with class labels per point

```
\begin{tikzpicture}
\begin{axis}[
    scatter/classes={
      a={mark=square*,blue},%
      b={mark=triangle*,red},%
      c={mark=o,draw=black}}]
\addplot[scatter,only marks,
    scatter src=explicit symbolic]
table {
    x       y         label
    0.1     0.15      a
    0.45    0.27      c
    0.02    0.17      a
    0.06    0.1       a
    0.9     0.5       b
    0.5     0.3       c
    0.85    0.52      b
    0.12    0.05      a
    0.73    0.45      b
    0.53    0.25      c
    0.76    0.5       b
    0.55    0.32      c
};
\end{axis}
\end{tikzpicture}
```

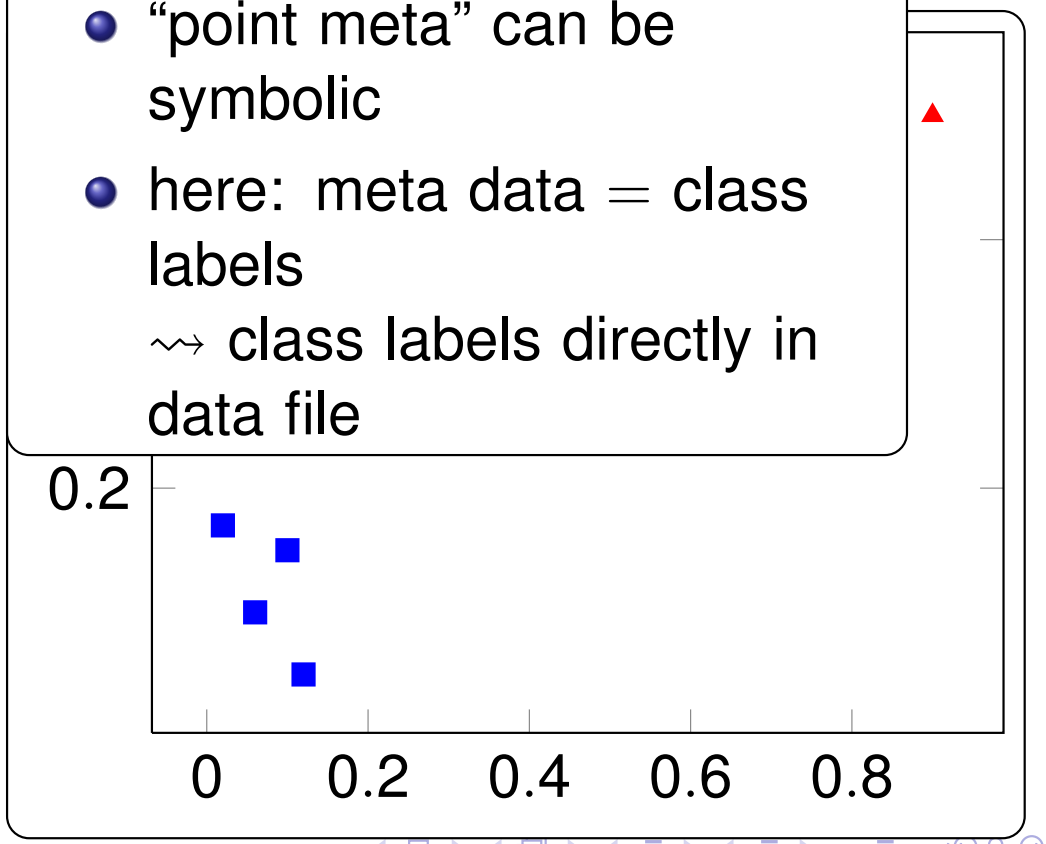


Scatter plot Use–case B

- scatter plot with class labels per point

```
\begin{tikzpicture}
\begin{axis}[
    scatter/classes={
      a={mark=square*,blue},%
      b={mark=triangle*,red},%
      c={mark=o,draw=black}}]
\addplot[scatter,only marks,
    scatter src=explicit symbolic]
table {
    x       y         label
    0.1     0.15      a
    0.45    0.27      c
    0.02    0.17      a
    0.06    0.1       a
    0.9     0.5       b
    0.5     0.3       c
    0.85    0.52      b
    0.12    0.05      a
    0.73    0.45      b
    0.53    0.25      c
    0.76    0.5       b
    0.55    0.32      c
};
\end{axis}
\end{tikzpicture}
```

- “point meta” can be symbolic
- here: meta data = class labels
  ⇝ class labels directly in data file



Scatter plot Use–case C

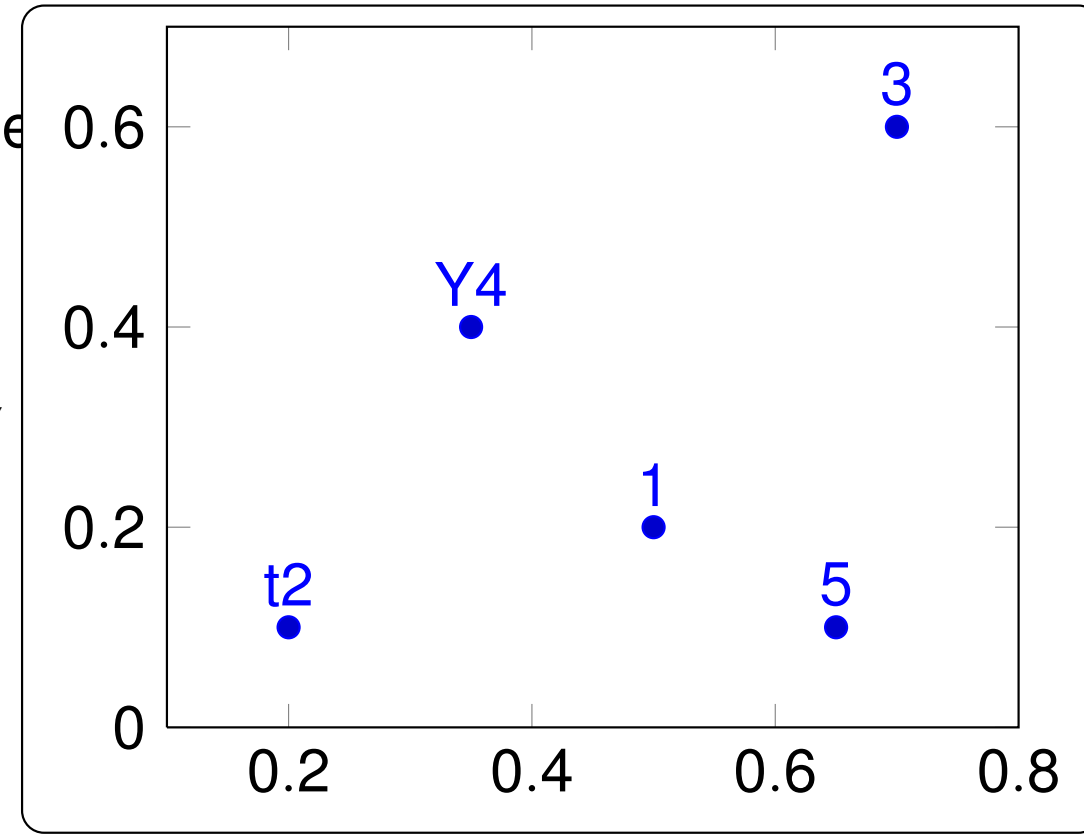
- scattered data + text labels per point

```
\begin{tikzpicture}
\begin{axis}[
    enlargelimits=0.2,
]
\addplot+[nodes near coords,only marks,
   point meta=explicit symbolic]
table[meta=label] {
    x    y   label
    0.5  0.2 1
    0.2  0.1 t2
    0.7  0.6 3
    0.35 0.4 Y4
    0.65 0.1 5
};
\end{axis}
\end{tikzpicture}
```

Scatter plot Use–case C

- scattered data + text labe

```
\begin{tikzpicture}
\begin{axis}[
    enlargelimits=0.2,
]
\addplot+[nodes near coords,only marks,
   point meta=explicit symbolic]
table[meta=label] {
    x    y   label
    0.5  0.2 1
    0.2  0.1 t2
    0.7  0.6 3
    0.35 0.4 Y4
    0.65 0.1 5
};
\end{axis}
\end{tikzpicture}
```

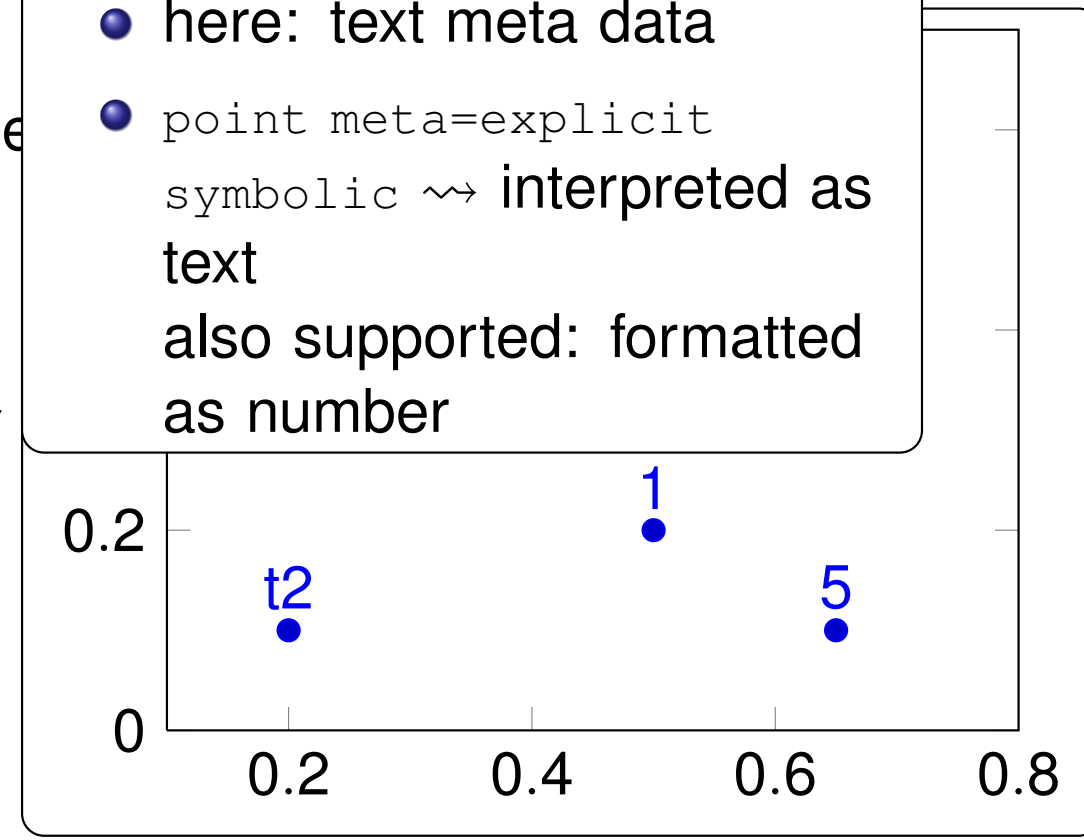


Scatter plot Use–case C

- scattered data + text labe

```
\begin{tikzpicture}
\begin{axis}[
    enlargelimits=0.2,
]
\addplot+[nodes near coords,only marks,
   point meta=explicit symbolic]
table[meta=label] {
    x    y   label
    0.5  0.2 1
    0.2  0.1 t2
    0.7  0.6 3
    0.35 0.4 Y4
    0.65 0.1 5
};
\end{axis}
\end{tikzpicture}
```

- here: text meta data
- `point meta=explicit symbolic` ⇝ interpreted as text
  also supported: formatted as number



Outline

Use–case: Functions of two variables

- Requirement: surface plot from data matrix + contour plot from expression
  Here: data file uses “scanlines”: empty lines indicate end-of-scanline

```
x_0             x_1             f(x)             l_0             i_0             scalaridx_0
0               0               0                0               0               0
0               0.19634954      0.038239818      0               0               0
0               0.39269908      0.14662768       0               0               0
⋮
0               5.8904862       0.14662768       0               0               0
0               6.0868358       0.038239818      0               0               0
0               6.2831853       6.9388939e-18    0               0               0

0.03125         0               3.0517578e-05    5               1               17
0.03125         0.19634954      0.030093496      5               1               17
0.03125         0.39269908      0.1146012        5               1               17
0.03125         0.58904862      0.24129102       5               1               17
0.03125         0.78539816      0.38946059       5               1               17
0.03125         0.9817477       0.53949733       5               1               17
```

Use–case: Functions of two variables

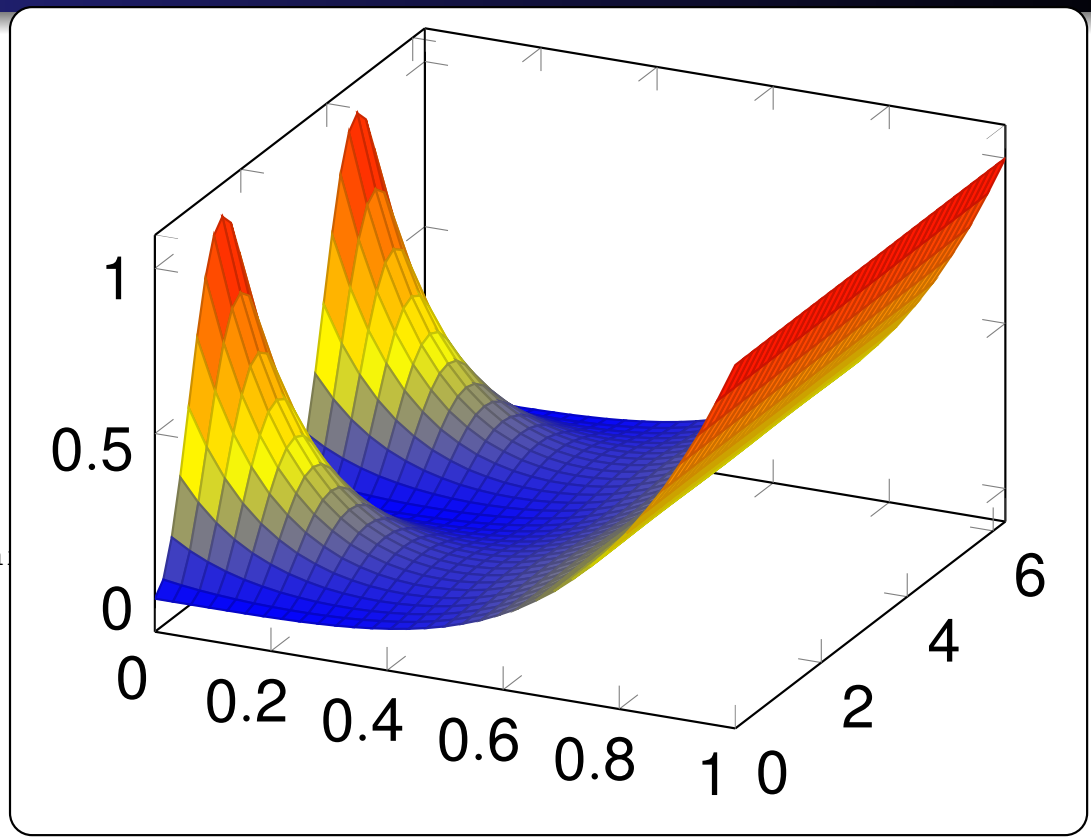
- Step 1: surface plot from data file

```
\begin{tikzpicture}
\begin{axis}
\addplot3[surf,mesh/ordering=y varies]
   table {concat_VV_together.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Functions of two variables

- Step 1: surface plot

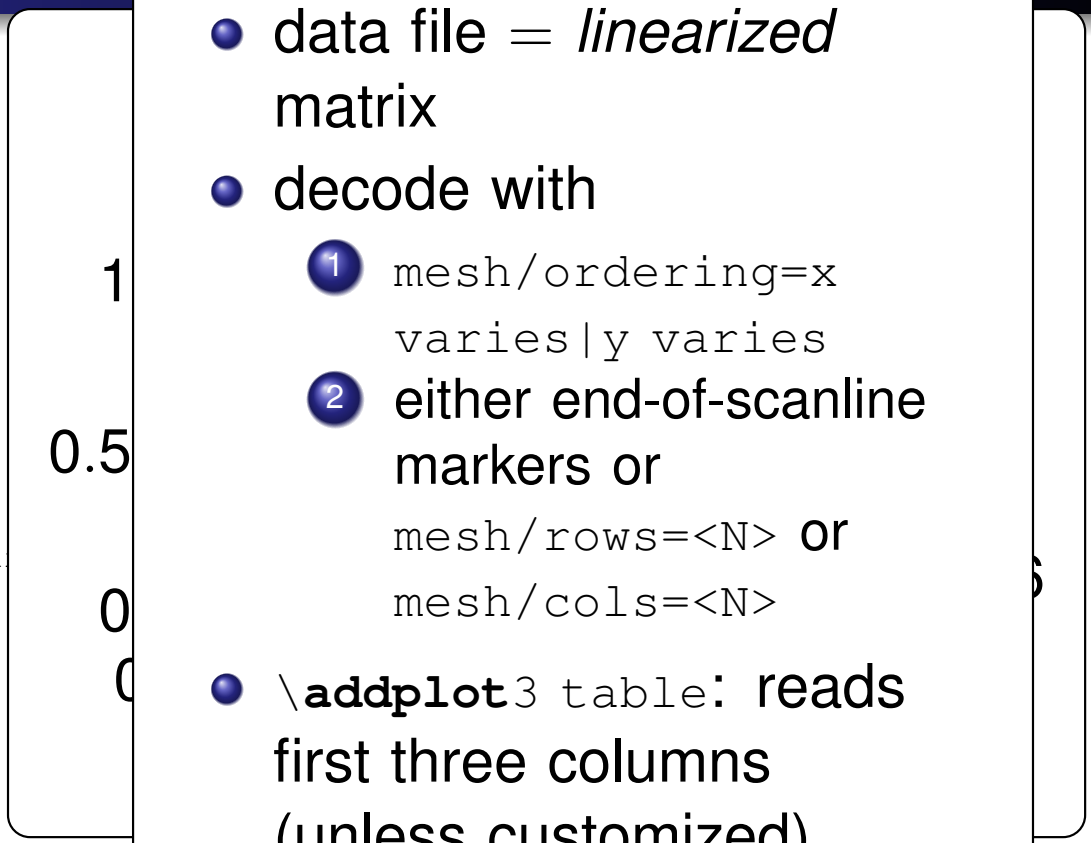
```
\begin{tikzpicture}
\begin{axis}
\addplot3[surf,mesh/ordering=y va
   table {concat_VV_together.dat}
\end{axis}
\end{tikzpicture}
```



Use–case: Functions of t

- Step 1: surface plot

```
\begin{tikzpicture}
\begin{axis}
\addplot3[surf,mesh/ordering=y va
   table {concat_VV_together.dat}
\end{axis}
\end{tikzpicture}
```



- key for 2d data files: matrix form.
- data file = *linearized* matrix
- decode with
  1. `mesh/ordering=x varies|y varies`
  2. either end-of-scanline markers or `mesh/rows=<N>` or `mesh/cols=<N>`
- `\addplot3 table`: reads first three columns (unless customized)

Use–case: Functions of two variables

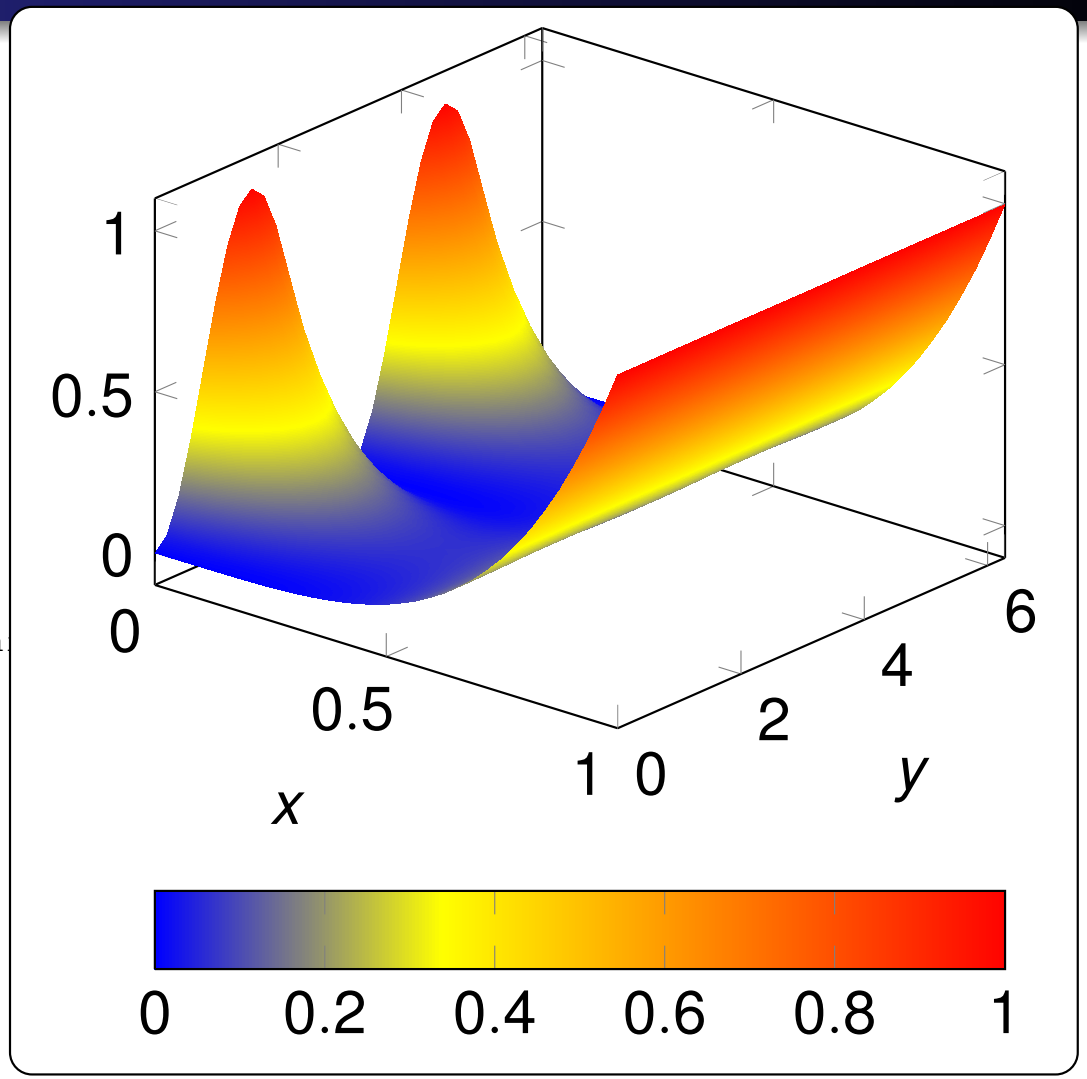
- Step 2: fine–tuning

```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y varies,
   shader=interp]
   table {concat_VV_together.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Functions of two variables

- Step 2: fine–tuning

```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y va
   shader=interp]
   table {concat_VV_together.dat}
\end{axis}
\end{tikzpicture}
```

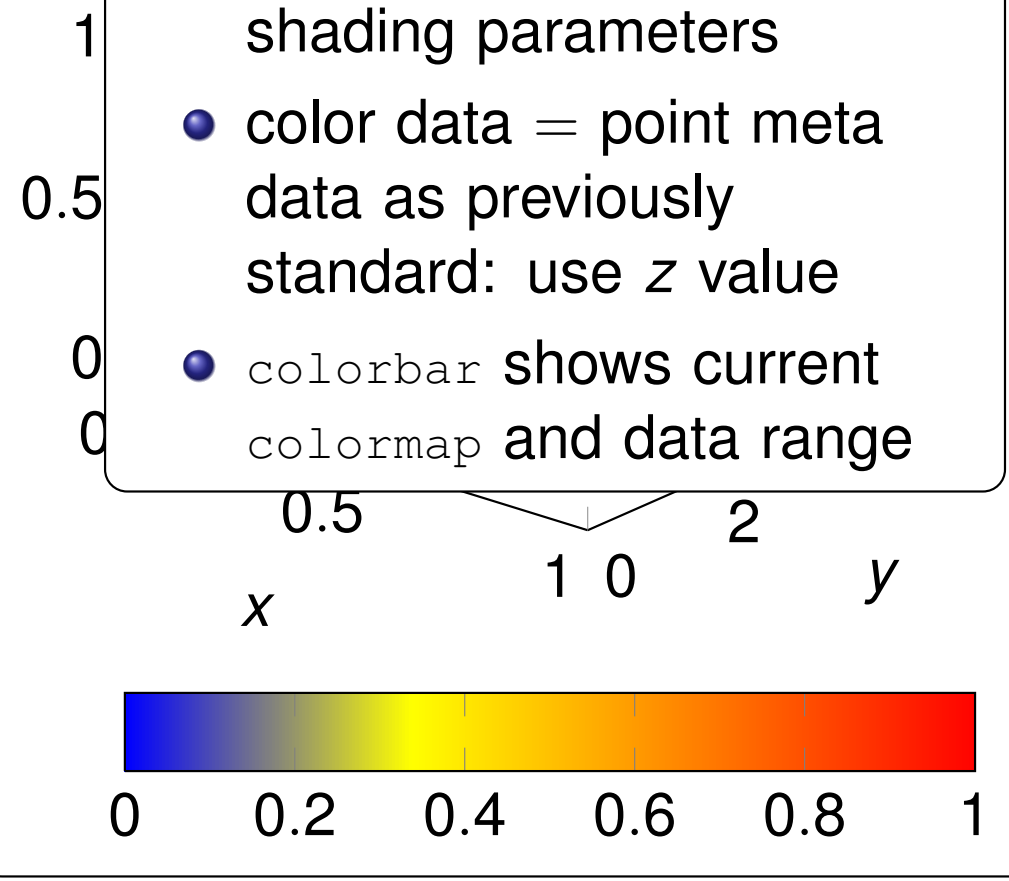


Use–case: Functions of t

- Step 2: fine–tuning

```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y va
   shader=interp]
   table {concat_VV_together.dat};
\end{axis}
\end{tikzpicture}
```

- `view/h=<angle>` ⇝ horizontal view angle
- `shader=...` various shading parameters
- color data = point meta data as previously standard: use z value
- `colorbar` shows current `colormap` and data range



Use–case: Functions of two variables

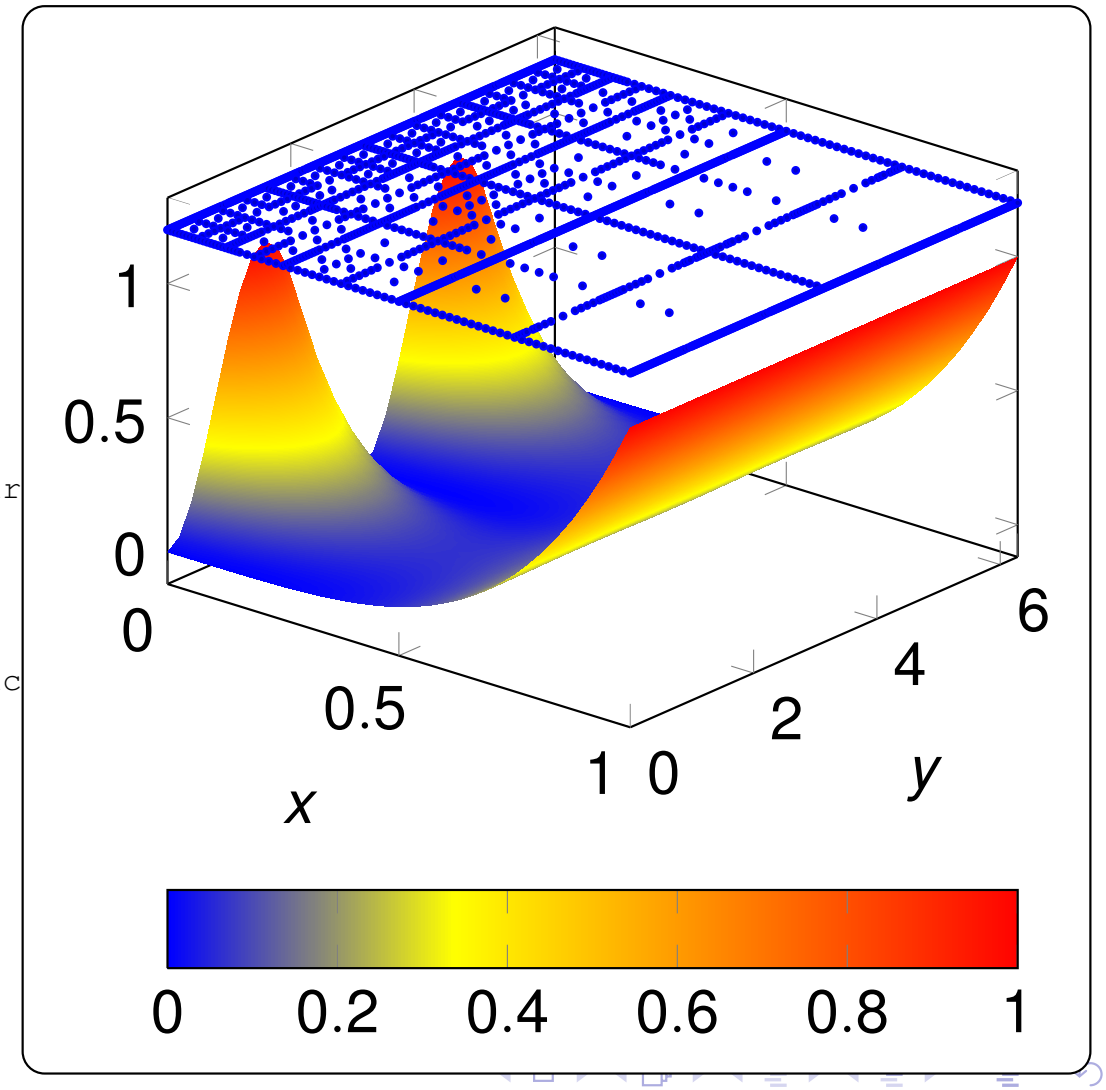
- Step 3: overlay grid

```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y varies,
   shader=interp]
table {concat_VV_together.dat};

\addplot3[blue,mark=*,
   mark options={fill=blue!80!black},
   only marks,mark size=0.6pt]
table[z expr=1.2]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```

Use–case: Functions of two variables

- Step 3: overlay grid

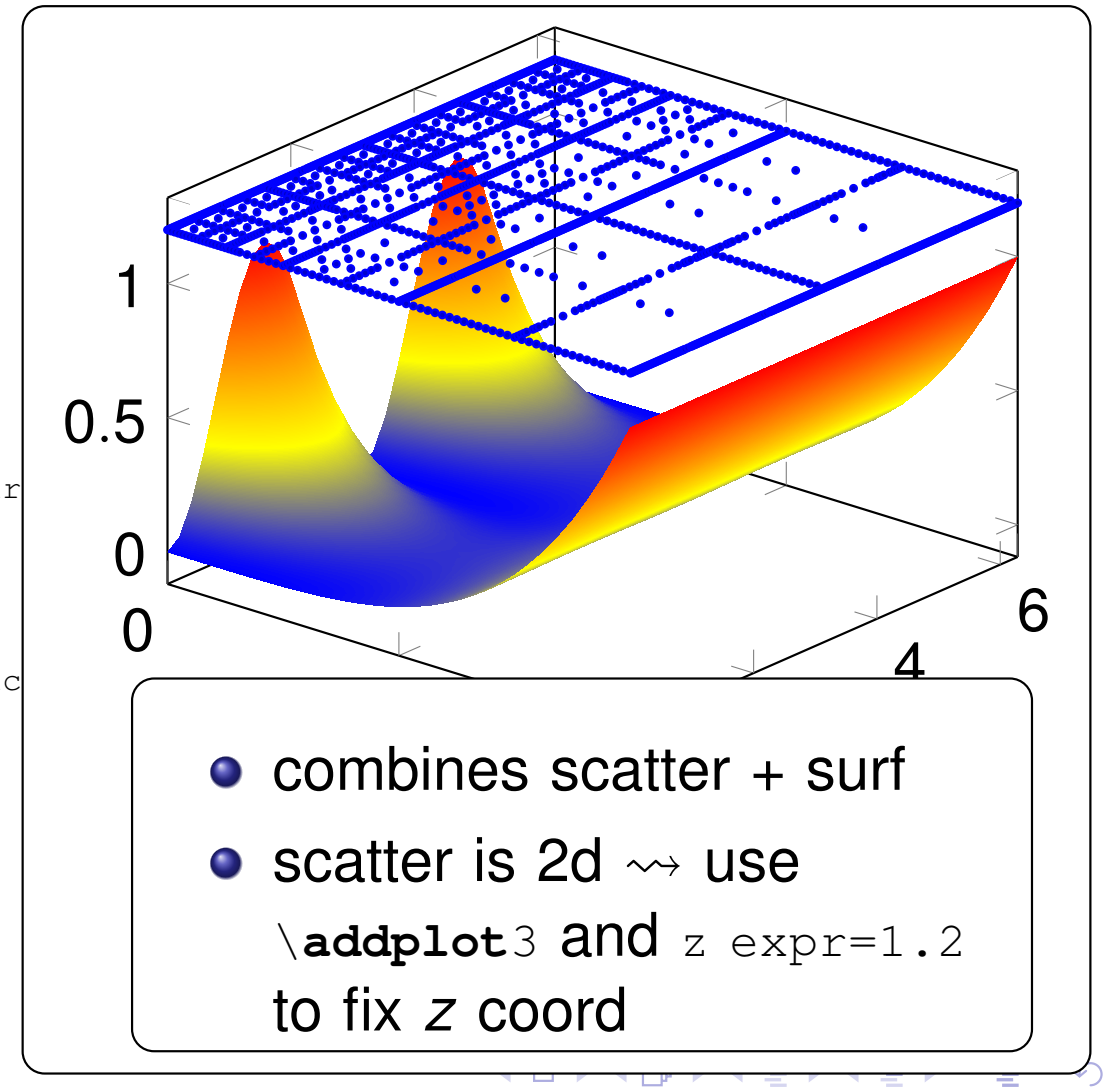
```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y var
   shader=interp]
table {concat_VV_together.dat};

\addplot3[blue,mark=*,
   mark options={fill=blue!80!blac
   only marks,mark size=0.6pt]
table[z expr=1.2]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



Use–case: Functions of two variables

- Step 3: overlay grid

```
\begin{tikzpicture}
\begin{axis}[
   view/h=40,
   colorbar horizontal,
   xlabel=$x$, ylabel=$y$,
]
\addplot3[surf,mesh/ordering=y var
   shader=interp]
table {concat_VV_together.dat};

\addplot3[blue,mark=*,
   mark options={fill=blue!80!blac
   only marks,mark size=0.6pt]
table[z expr=1.2]
   {concat_VV_together_grid.dat};
\end{axis}
\end{tikzpicture}
```



- combines scatter + surf
- scatter is 2d ⇝ use `\addplot3` and `z expr=1.2` to fix z coord

Use–case: Functions of two variables

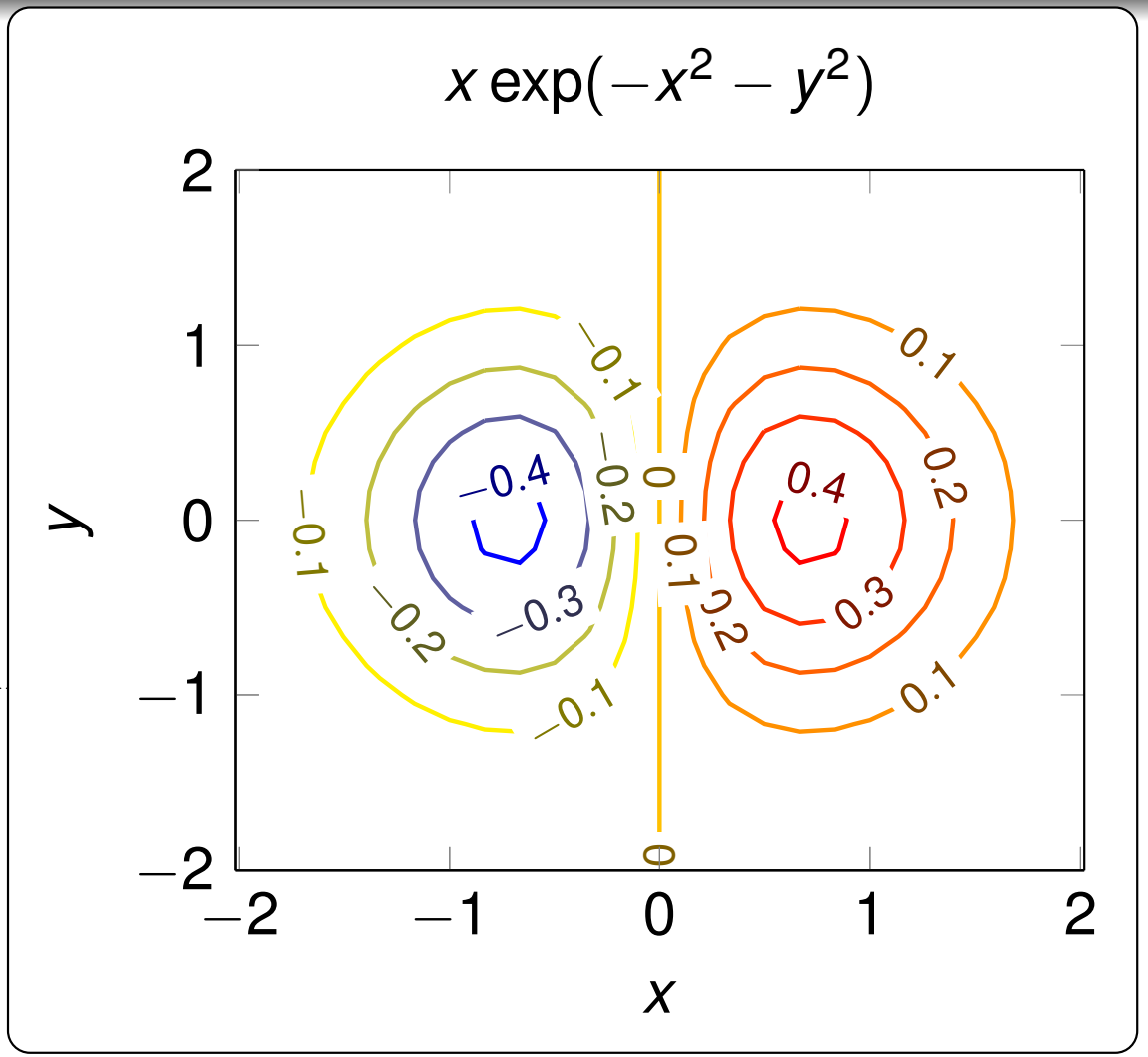
- Step 4: contour plot of function expression

```
\begin{tikzpicture}
\begin{axis}[
   title={$x \exp(-x^2-y^2)$},
   enlarge x limits,
   view={0}{90},
   xlabel=$x$, ylabel=$y$,
]
\addplot3[domain=-2:2,
   contour gnuplot={number=14},thick]
   {exp(-x^2-y^2)*x};
\end{axis}
\end{tikzpicture}
```

Use–case: Functions of two variables

- Step 4: contour plot

```
\begin{tikzpicture}
\begin{axis}[
   title={$x \exp(-x^2-y^2)$},
   enlarge x limits,
   view={0}{90},
   xlabel=$x$, ylabel=$y$,
]
\addplot3[domain=-2:2,
   contour gnuplot={number=14},th
   {exp(-x^2-y^2)*x};
\end{axis}
\end{tikzpicture}
```

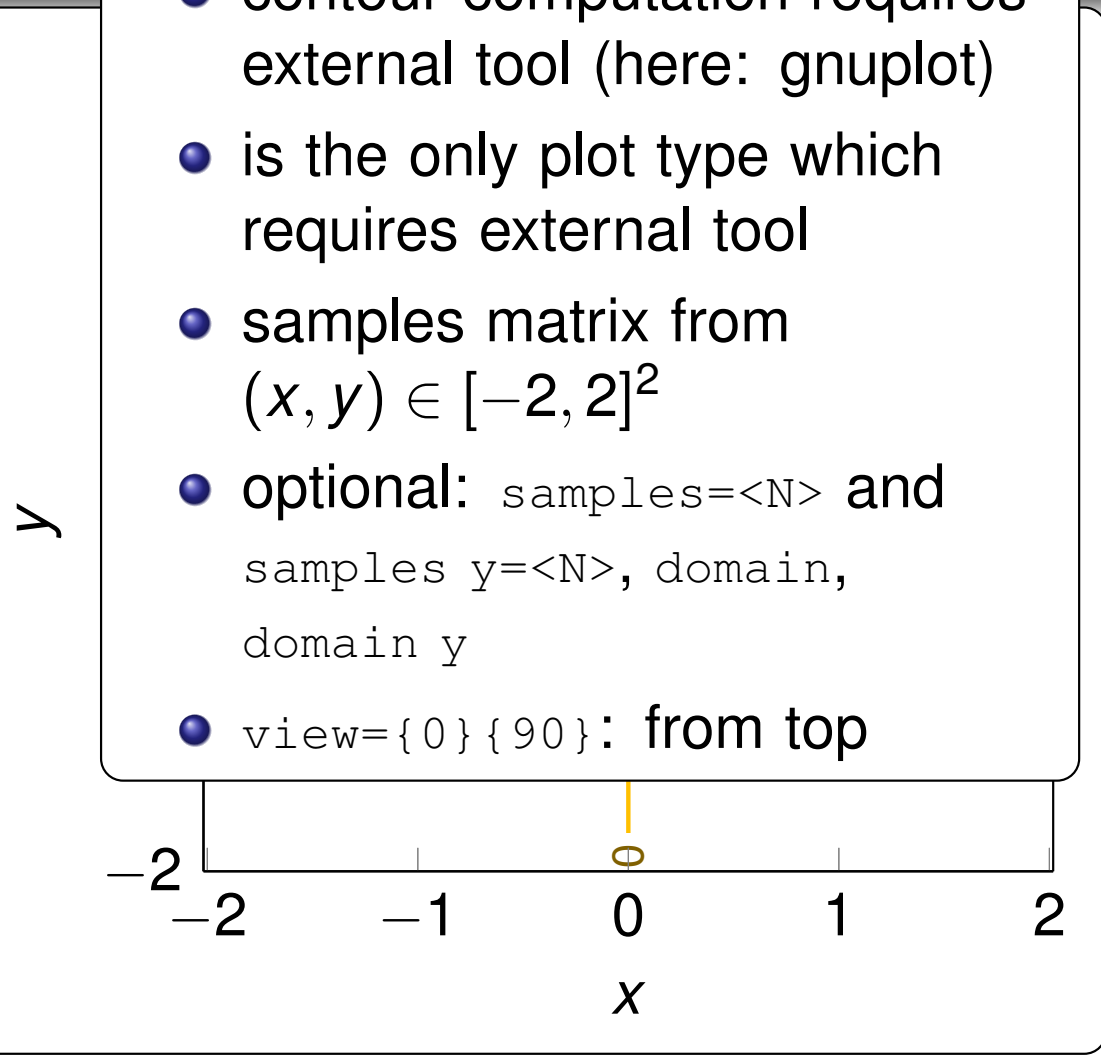


Use–case: Functions of t

- Step 4: contour plot

```
\begin{tikzpicture}
\begin{axis}[
   title={$x \exp(-x^2-y^2)$},
   enlarge x limits,
   view={0}{90},
   xlabel=$x$, ylabel=$y$,
]
\addplot3[domain=-2:2,
   contour gnuplot={number=14},th
   {exp(-x^2-y^2)*x};
\end{axis}
\end{tikzpicture}
```

- contour computation requires external tool (here: gnuplot)
- is the only plot type which requires external tool
- samples matrix from $(x, y) \in [-2, 2]^2$
- optional: `samples=<N>` and `samples y=<N>`, `domain`, `domain y`
- `view={0}{90}`: from top



Summary and Outlook

- user provides data + labels, pgfplots does the rest
- font+style consistency text/figure; figure/figure
- high-quality; powerful; flexible
- embedded solution (no 3rd party tools)
- supports pdftex, luatex, dvips, dvipdfm (partially)

Outlook on other features of PGFPlots:

- automatic image externalization
- plot types: bar plots; patch plots; quiver plots, . . .
- axis types: polar, smithchart, ternary
- table package companion
- full IEEE double range + 4-5 digits precision
- limitations: CPU intensive; 3d z buffer limited

Further reading

- `http://pgfplots.sourceforge.net`