A SmallTalk on Plotting and Gnuplot

Arne Tobias Elve (arne.t.elve@ntnu.no)

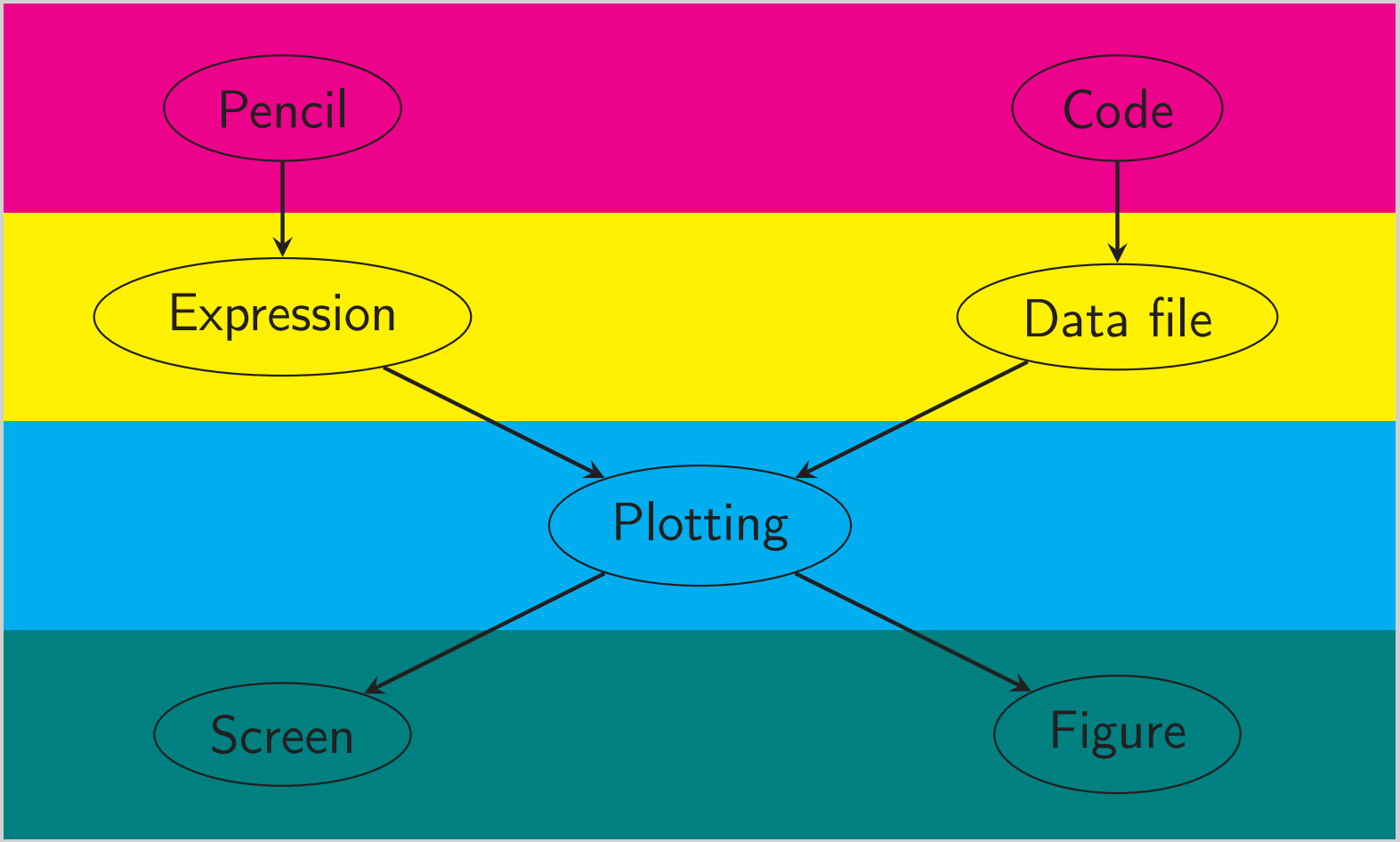
Department of Chemical Engineering
NTNU

6 April 2017

Disclaimer and Acknowledgements

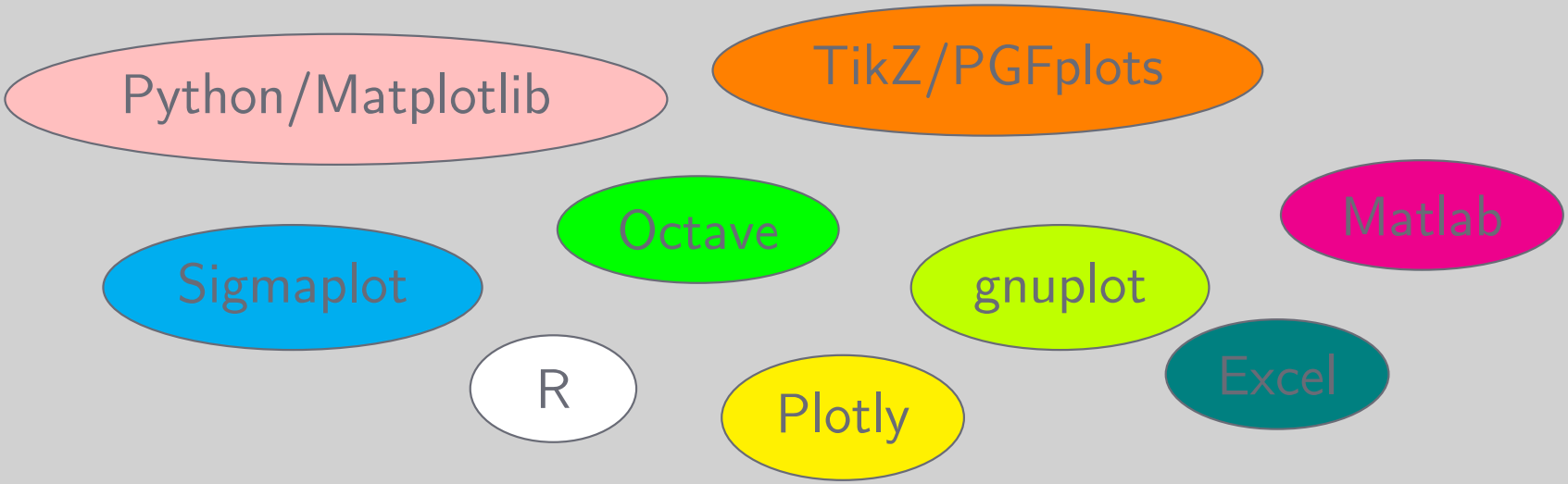
- I am no expert in gnuplot
- Based on my experience
- Achkowledgements to Tore Haug-Warberg & Sigve Karolius
- Contents of slides lifted with premission from:
  - *TKP4120: Different plotting methods*
  - *Different courses I have attended in some way at NTNU*
  - *Other presentations available on the net and tutorials*

1. My workflow



1. Plotting – We all do it

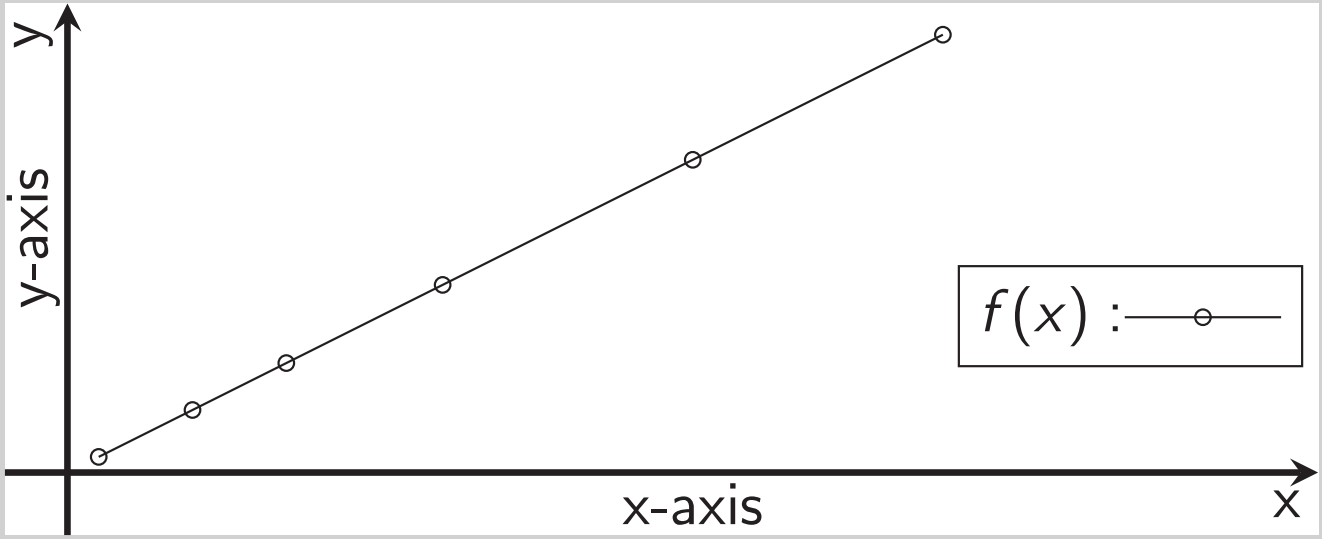
- Many tools for plotting exist



- All these tools does basically the same task

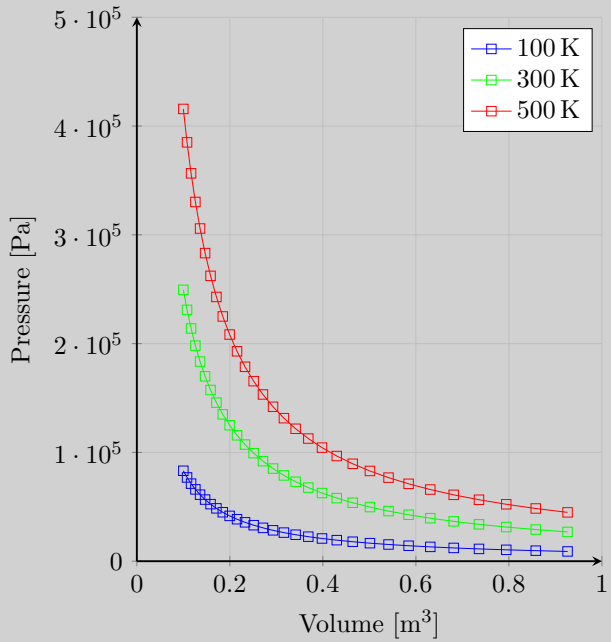
1. Plotting – Basics

- Making a plot by coding is pretty much the same you would do by hand.
  - *Start with a blank paper*
  - *Draw the axis*
  - *Include the data points*
  - *Draw lines between the points*
  - *Finish with axis labels, and legend*



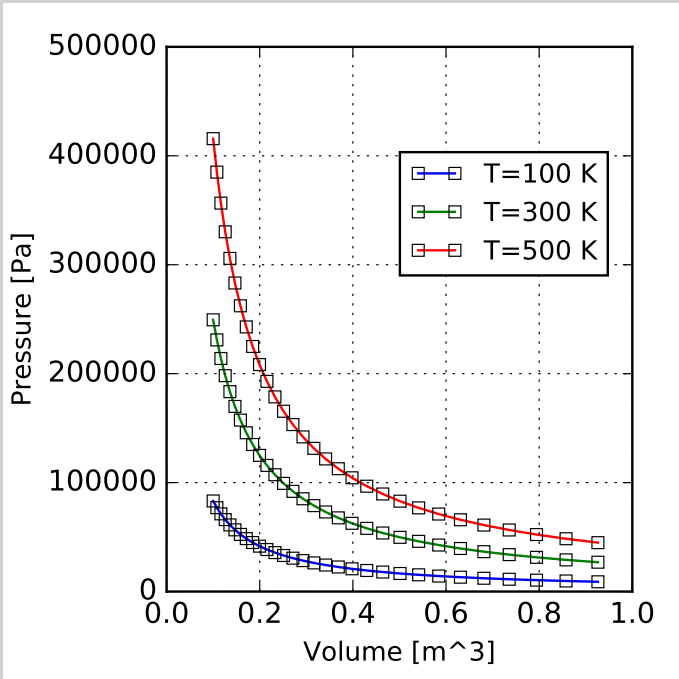
1. PGFplots

```
\begin{tikzpicture}
 \draw (0cm,0cm) circle (0pt);   % canvas lower left
 \draw (10cm,10cm) circle (0pt); % canvas upper right
  \begin{axis}[
  axis line style={->},
  axis line style = thick,
  grid=major,
  scale only axis,
  axis x line=bottom,
  scaled x ticks = false,
  axis y line=left,
  scaled y ticks = false,
  width=6.89655cm,height=8.06452cm,
  at={(2.41379cm,1.29032cm)},
  xmin=0.0,xmax=1.0,
  xlabel={Volume [$\mathrm{m}^3$]},
  ymin=0.0,ymax=5e5,
  ylabel={Pressure [$\mathrm{Pa}$]},
 ]{}%
  \addplot[mark=square,mark options={blue}, color=blue, solid]
          table[x index=0,y index=1,col sep=comma]{../pptable.txt};
  \addplot[mark=square,mark options={green},color=green,solid]
          table[x index=0,y index=2,col sep=comma]{../pptable.txt};
  \addplot[mark=square,mark options={red},
color=red,  solid]
          table[x index=0,y index=3,col sep=comma]{../pptable.txt};
  \addlegendentry{$100\,\mathrm{K}$};
  \addlegendentry{$300\,\mathrm{K}$};
  \addlegendentry{$500\,\mathrm{K}$};
  \end{axis}
\end{tikzpicture}
```



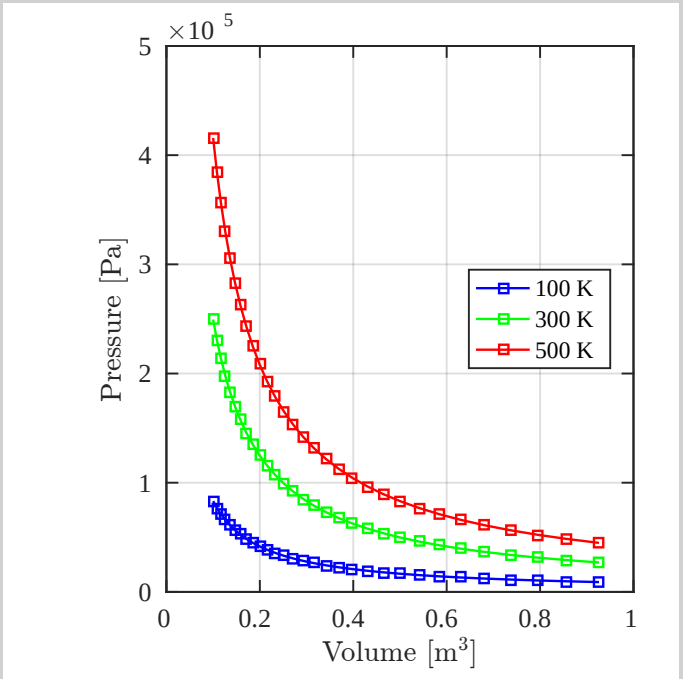
1. Python

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3  #-------------------------------
4
5  w = 100.0/25.4
6  h = 100.0/25.4
7
8  plt.figure(figsize=(w, h), dpi=300)
9  plt.gca().set_color_cycle([ 'b', 'g', 'r', 'c', 'y', 'm'
10
11 f = '../pptable.txt'
12 t = np.loadtxt(f, comments=('#','%'), delimiter=',')
13
14 v = [ ti[0 ] for ti in t ]
15 p = [ ti[1:] for ti in t ]
16
17 plt.plot(v, p, marker='s', markersize=5, markerfacecolor
18 plt.xlabel('Volume␣[m^3]', fontsize=11)
19 plt.ylabel('Pressure␣[Pa]', fontsize=11)
20 plt.ylim([ 0.0, 5e5 ])
21 plt.xlim([ 0.0, 1.0 ])
22
23 plt.subplots_adjust(left=xl, right=xr, bottom=yb, top=yt)
24
25 plt.legend(['T=100␣K', 'T=300␣K', 'T=500␣K'], fontsize=11, loc='center', \
26             bbox_to_anchor=(xc, yc))
27
28 plt.grid(True)
29
30 plt.savefig('myfig.pdf')
```



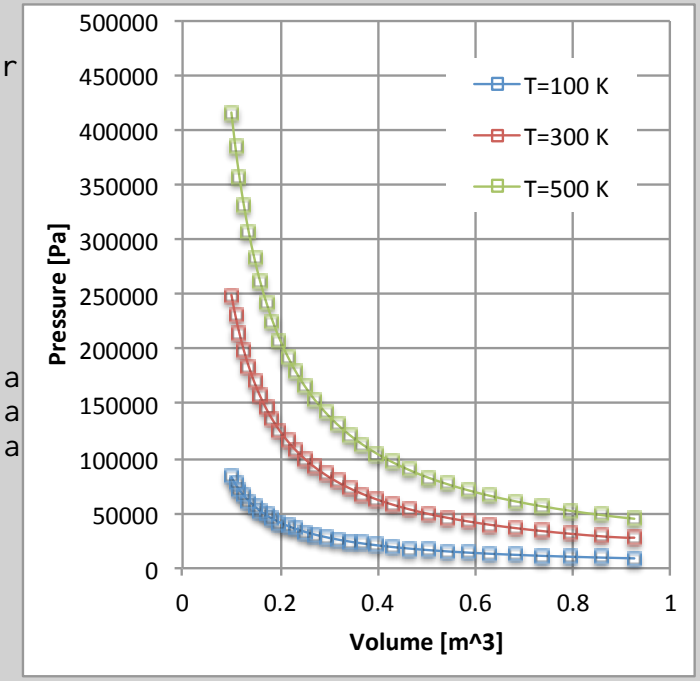
1. Matlab

```
1  set(gcf,'PaperSize', [10, 10]);
2
3  figure()
4  table = load('-ascii', '../pptable.txt');
5
6  x = table(:,1);                    % extract volume
7  y = table(:,2:end);                % extract pressure
8
9  % Plot. Note that 'subplot' position is relative to gcf
10 subplot('position', [xl, yb, xr-xl, yt-yb]);
11 plot(x, y, '-s');
12
13 axis([0, 1, 0, 5e5]);
14 set(gca, 'xaxislocation', 'origin');
15 set(gca, 'yaxislocation', 'origin');
16 set(gca, 'XTick', [0:0.2:1]);
   % Only works after the plot command
17 set(gca, 'GridLineStyle', '-');
18
19 % Legend. Postitioning in Octave does not work and we want
   to be  portable.
20 leg = legend('100␣K', '300␣K', '500␣K', 'Location', 'east');
21 xlabel(gca, 'Volume␣[$\mathrm{m^{3}}$]', 'Interpreter', 'latex');
22 ylabel(gca, 'Pressure␣[$\mathrm{Pa}$]',
   'Interpreter', 'latex');
23
24 % Save.
25 print('myfig','-dpdf');
```



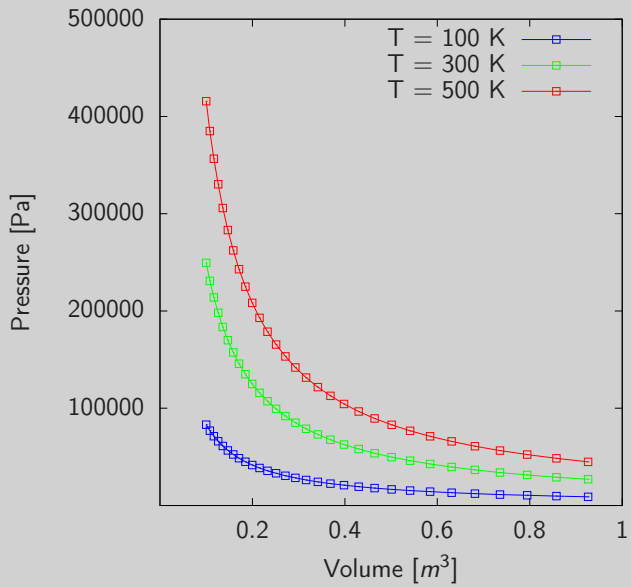
1. Excel

```
            Click: <standard>
              Input: Existing sheet <=$A$1> (or whatever
1
2
3
4   Select: All columns
5
6   Menu: Scatter diagram
7     Click: Smoothed curve
8
9   Right-click: A curve or legend
10    Click: Choose data
11      Input: Name of "Serie1" changed to <T=100 K> (or wha
12      Input: Name of "Serie2" changed to <T=300 K> (or wha
13      Input: Name of "Serie3" changed to <T=500 K> (or wha
14      ... etc.
15
16  Right-click: Curve 1
17    Click: Format data series
18      Click: Marker line
19        Click: Thickness and arrows
20          Input: <1>pt (or whatever)
21        Click: Indicator style
22          Click: <square symbol>
23        Click: Marker fill
24          Click: <no fill>
25  ... etc.
26
27  Right-click: Horisontal axis
28    Click: Add grid lines
29
30  Right-click: Vertical axis
31    Click: Add grid lines
```



1. Gnuplot

```
1   set terminal postscript enhanced eps \
2       font "Helvetica" 18 \
3       size 10.cm, 10.0cm
4
5   set datafile commentschars "#!%"
6   set datafile separator ",;\t"
7   set datafile missing "NaN"
8   set output OUTPUTFILE
9   set xrange [0.0:1.0]
10  set yrange [0.0:5.0e5]
11  set ytics 1e5, 1e5, 5e5
12  set mytics 1
13
14  set xtics 0.2, 0.2, 1.0
15  set mxtics 1
16
17  set xlabel "Volume [m^3]"
18  set ylabel "Pressure [Pa]"
19
20  set style line 1 \
21      linetype -1 linewidth 1.0 linecolor 3 pointtype 4 pointsize 1.0
22
23  set style line 2 \
24      linetype -1 linewidth 1.0 linecolor 2 pointtype 4 pointsize 1.0
25
26  set style line 3 \
27      linetype -1 linewidth 1.0 linecolor 1 pointtype 4 pointsize 1.0
28
29  # Column 1 along x-axis and columns 2, 3 and 4 along y-axis:
30  plot INPUTFILE using 1:2 title "T = 100 K" with linespoints linestyle 1, \
31       INPUTFILE using 1:3 title "T = 300 K" with linespoints linestyle 2, \
```



2. An introduction to Gnuplot

- Gnuplot is a command-based graphing utility for:
  - *plotting data files*
  - *plotting analytical functions*
  - *curve fitting*
- The source code is copyrighted but freely distributed.
- Command-line program.

2. Installation

- Go to: `http://www.gnuplot.info/`
  - *Windows: Download tar-files*
  - *Linux:* `apt install gnuplot`
  - *Mac:* `brew install gnuplot`
- On the webpage you also find:
  - *manual:* `http://www.gnuplot.info/docs_5.0/gnuplot.pdf`
  - *examples*
  - *tutorials*

2. Features

- You can
  - *work with it*
  - *produce figures for presentations*
  - *scripts for producing figures*
- 2D plots
- 3D plots
- 4D plots (Gif)
- Contour plots
- Any output format (pdf, eps, tex, ps, png, jpeg, HTML5, tikz and many more)

2. Terminals

- The output from the program is determined using terminals
- To list all available terminal type: `set terminal`

Table: Frequently used terminals in gnuplot

name	output	description
pngcairo	png	Produce png-figure
postscript	eps	Produce eps-figure
dumb	plain text	ASCII-art
wxt,aqua,x11	window	Interactive window
epslatex	tex	Plain eps for plot and tex overlays

2. Datafile

- Data files for gnuplot is arranged in columns

```
1  ##########################################
2  # Purpose: Data file Kalman filter data
3  # Author:  Arne Tobias Elve
4  # Date:    Wed Apr  5 14:21:47 2017
5  # Why:     Exercise in TKP4135
6  ##########################################
7  # step   measured kalman
8  700      1214.84  1100
9  701.059  1214.37  1102.25176471
10 702.119  1209.15  1105.47588459
11 703.178  1196.02  1109.34400826
12 704.237  1208.36  1113.25944403
13 705.297  1192.28  1118.23209307
14 706.356  1193.63  1122.57396541
15 707.415  1189.69  1127.1377283
16 708.475  1197.59  1131.46000466
17 709.534  1198.39  1136.3073763
18 710.593  1194.26  1141.08123379
19 711.653  1194.75  1145.33277605
20 712.712  1190.04  1149.41093923
21 713.771  1168.74  1152.85178401
22 714.831  1171.45  1154.22609483
```

- Does not matter if the data is neatly lined up. Gnuplot looks for spaces

2. Preparing Figures Gnuplot (6)

```
gnuplot
gnuplot> set terminal                                    # List all available terminals
gnuplot> set terminal wxt                                  # select wxt terminal
gnuplot> plot 'data.txt' using 1:2
gnuplot> plot 'data.txt' using 1:2, 'data.txt' using 1:3
gnuplot> plot 'data.txt' using 1:2, 'data.txt' using 1:3, x
gnuplot> plot 'data.txt' using 1:2, 'data.txt' using 1:3, 1200
gnuplot> set terminal pngcairo                             # select png terminal
gnuplot> plot 'data.txt' using 1:2, 'data.txt' using 1:3, 1200  # dump to screen
gnuplot> set output 'data.png'
gnuplot> plot 'data.txt' using 1:2, 'data.txt' using 1:3, 1200   # print to file
```

3. Scripting commands

- Instead of writing all commands every time use scripts

```
1 set terminal pngcairo
2 set output 'data.png'
3 plot 'data.txt' using 1:2, 'data.txt' using 1:3, 1200          # print to file
```

- save this script as `main.gp`
- run script:

```
1 gnuplot main.gp
```

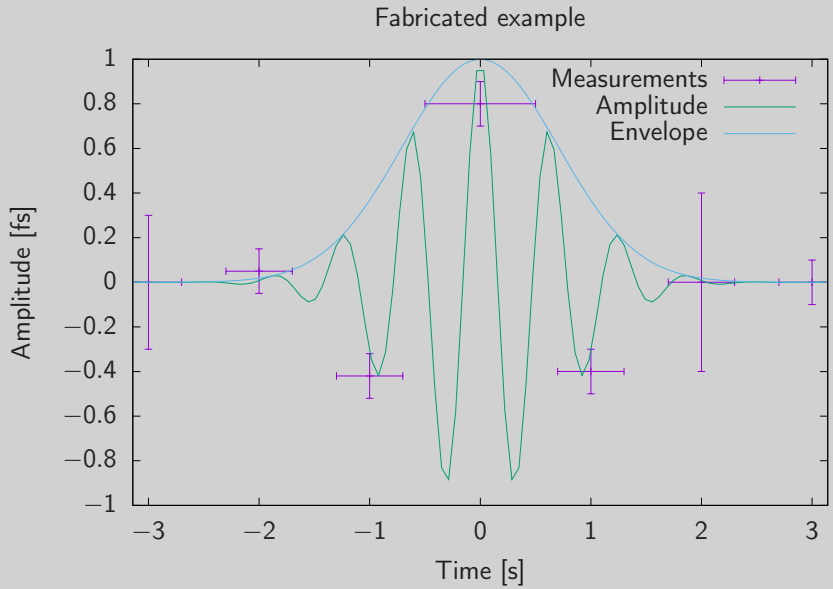
3. Basic commands: Example

gnuplot script:

```
1  set terminal epslatex
2  set output 'basic.tex'
3  set title 'Fabricated example'
4  set xlabel 'Time [s]'
5  set ylabel 'Amplitude [fs]'
6  set xrange [-pi:pi]
7  plot 'Data/data.txt'  using 1:2:3:4 \
8  with xyerrorbars title 'Measurements',\
9  cos(x*10)*exp(-x*x) t 'Amplitude', \
10 exp(-x*x) title 'Envelope'
```

Datafile:

```
1  ########################################
2  # Author: Arne Tobias Elveo
3  # Date:   2017-04-05
4  # Why:    Gnuplot presentation
5  # How:    Lifted example from uni-graz
6  ########################################
7  # x    y    xe   ye
8  -3   0    0.3    0.3
9  -2   0.05    0.3    0.1
10 -1   -0.42    0.3    0.1
11 0    0.8    0.5    0.1
12 1   -0.4    0.3    0.1
13 2   0    0.3    0.4
14 3   0    0.3    0.1
```



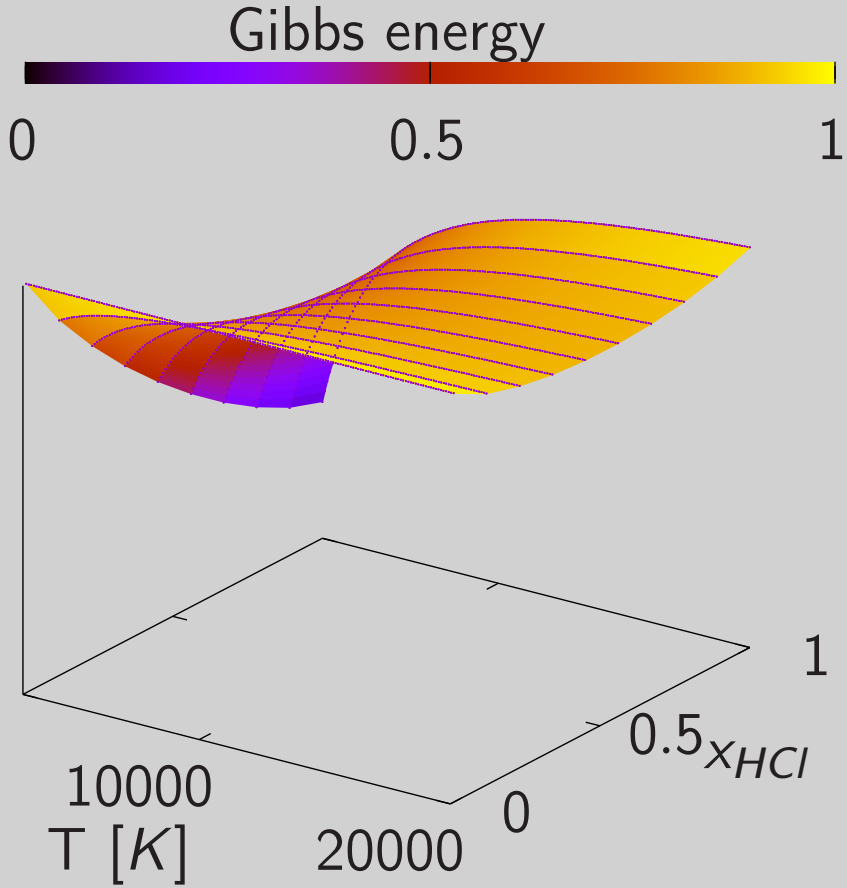
3. 3D plots

gnuplot script:

```
1  set term epslatex size 7cm,7cm color
2  set output "treDD.tex"
3  set title "Gibbs energy"
4  set xtics 10000 offset -1, -0.5, 0
5  set ytics 0.5 offset 1, 0, 0
6  unset ztics
7  set surface
8  set colorbox user size 0.74,\
9    .02 noborder horizontal
10 set colorbox user origin 0.17,0.85
11 set cbtics 0.5 offset -0.1
12 set xrange [3000:20000]
13 set view 65,35
14 set key off
15 set pm3d
16 set xlabel "T [$K$]" offset -1.2,-1.1,0
17 set ylabel "$x_{HCl}$" offset -1,-0.5,0
18 splot "Data/treData.csv" \
19   using 1:2:3 ps 0.1
```

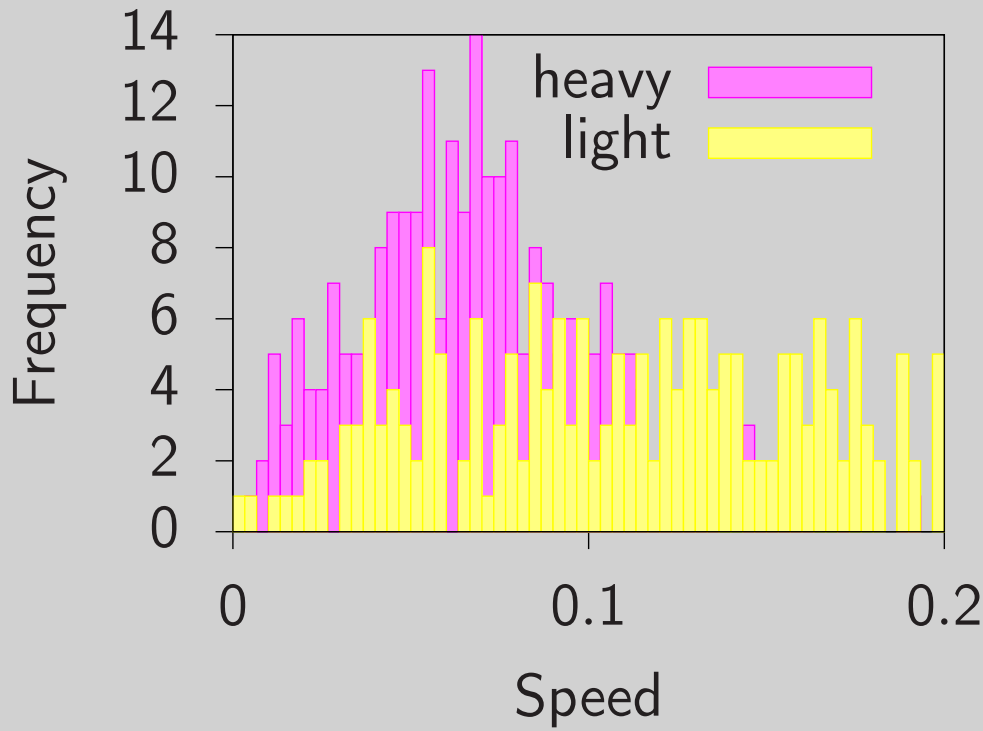
Datafile:

```
1  # T      x_HCl           Energy
2  3000   0.010989010989   1.0
3  3000   0.120879120879   0.830420582482
4  3000   0.230769230769   0.695437208432
5  3000   0.340659340659   0.577965805164
6  3000   0.450549450549   0.473814781249
7  3000   0.56043956044    0.381377112941
8  3000   0.67032967033    0.300264439273
```



3. Histograms

```
1  set term epslatex color size 6.8cm,5cm
2  set output 'histogram_both_7s.tex'
3  n=60 #number of intervals
4  max=0.2 #max value
5  min=0. #min value
6  wid=(max-min)/n #interval wid
7  # Define function for histograms
8  hist(x,wid)=wid*floor(x/wid)+wid/2.0
9  set xrange [min:max]
10 set yrange [0:]
11 set xtics min,(max-min)/2,max
12 set boxwidth wid
13 set style fill solid 0.5 #fillstyle
14 set tics out nomirror
15 set xlabel "Speed"
16 set ylabel "Frequency"
17
18 plot 'Data/problem3_speed_heavy.txt'\
19 using (hist($8,wid)):(1.0) smooth freq \
20 with boxes linecolor rgb "magenta"\
21 title "heavy",\
22 "./Data/problem3_speed_small.txt" \
23 u (hist($8,wid)):(1.0) smooth freq \
24 w boxes lc rgb"yellow" t "light"
```

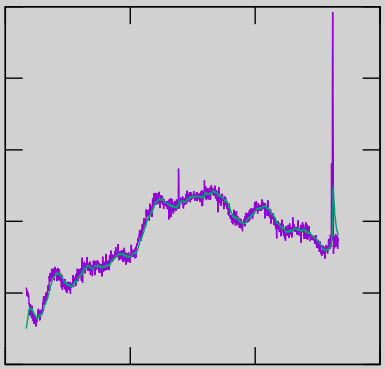


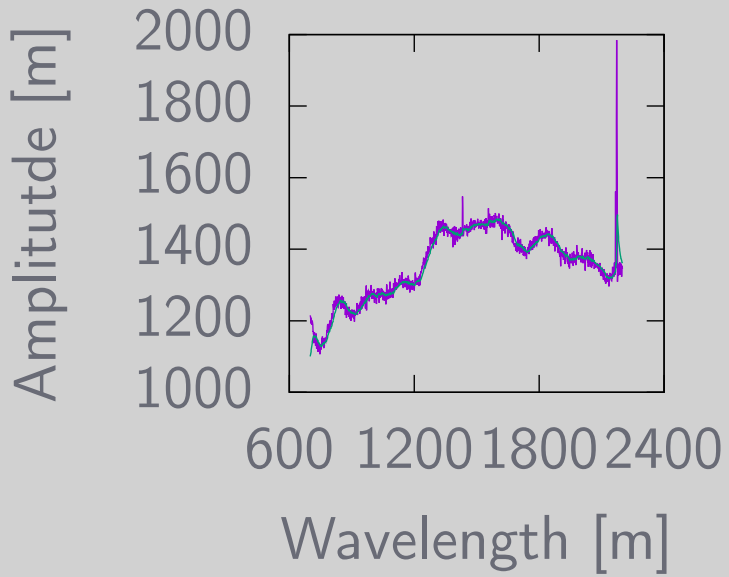
Datafile: 'Data/problem_3_speed_small.txt'

```
1  #0 sec   1 sec   2 sec   3 sec   4 sec   5 sec   6 sec   7 sec   8 sec   9 sec
2  0.1000   0.1525   0.2397   0.2694   0.1241   0.2782   0.1681   0.1574   0.0994   0.1321
3  0.1000   0.0980   0.1466   0.0817   0.1444   0.2015   0.2015   0.2422   0.1802   0.1802
```

4. LaTeX and gnuplot – Basics

1. Select one of the latex terminals:
   - `latex, epslatex, tikz, etc...`
   - *My personal favorite is* `epslatex`
   - *epslatex produces the plot as a plain eps and tex file.*





2. set name of output file TeX files
3. run your gnuplot script
4. transclude the tex file in your LaTeX document using:

```
1 | {{\graphicspath{Figures/Path/To/Folder} \input{Figures/Path/To/Folder/file}}}
```

4. Simple example: code

- Gnuplot code `simple.gp`

```
set terminal epslatex
set output 'eq1.tex'
plot [-3.14:3.14] sin(x)
```

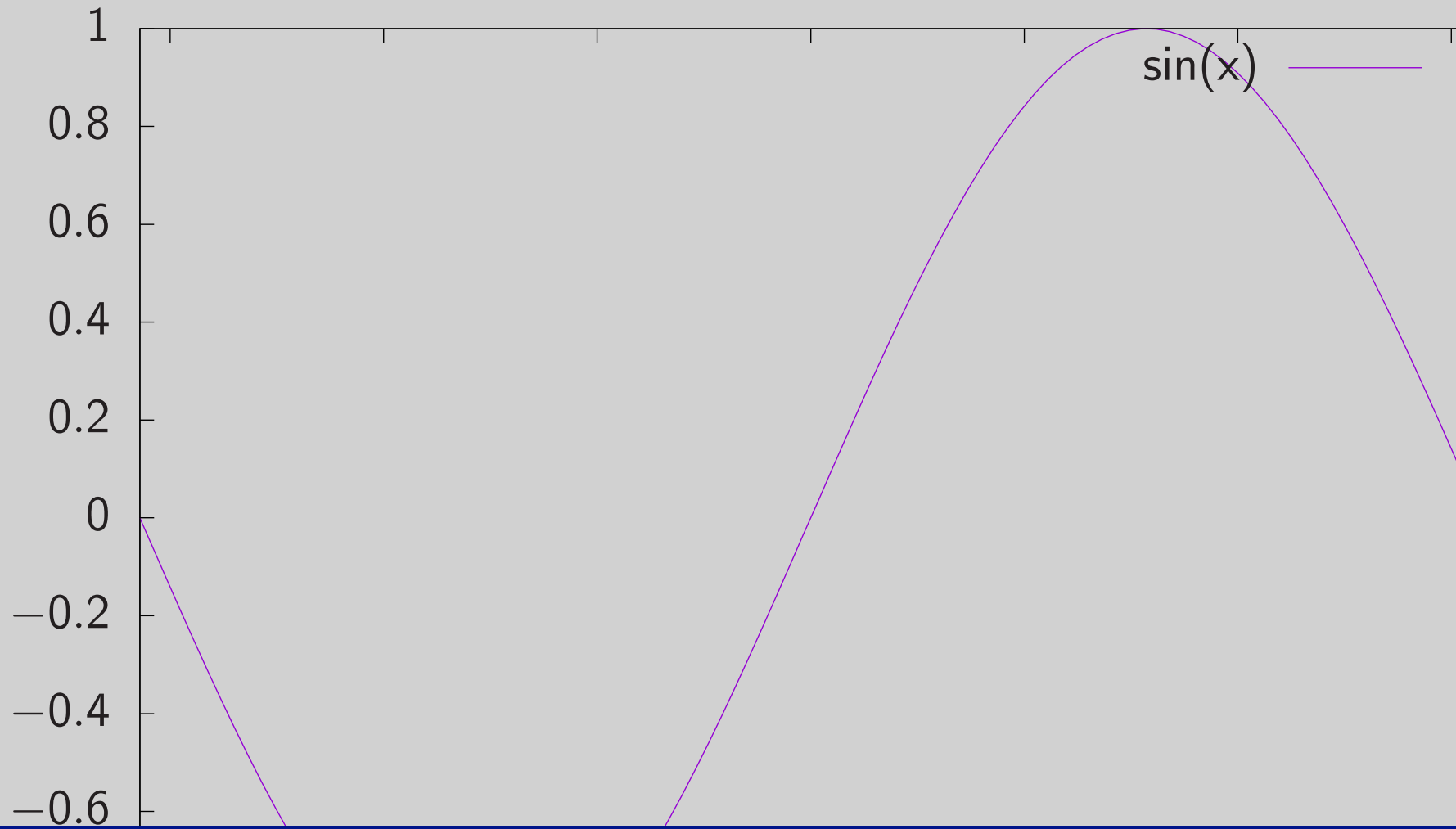
- run gnuplot script

```
gnuplot simple.gp
```

- LaTeX-code to produce next frame:

```
\frame{
  \frametitle{Simple example: figure}
  \begin{figure}
    \centering
    {\graphicspath{{./Figures/LaTeX/}} \input{./Figures/LaTeX/eq1}}
    \caption{$f\left(x\right) = \sin\left(x\right)$}
    \label{fig:simpleexample}
  \end{figure}
}
```

4. Simple example: figure



4. A more fancy example: code

- Gnuplot code simple2.gp

```
set terminal epslatex size 10cm,6cm
set output 'eq2.tex'
set xrange [-3.14:3.14]
set yrange [-1.2:1.2]
set xlabel 'This is the $x$-axis'
set ylabel '$y$'
plot sin(x)
```

- run gnuplot script

```
gnuplot simple2.gp
```

- LaTeX-code to produce next frame:

```
\frame{
  \frametitle{A more fancy example: figure}
  \begin{figure}
    \centering
    {\graphicspath{./Figures/LaTeX/} \input{./Figures/LaTeX/eq2}}
    \caption{$f\left(x\right) = \sin\left(x\right)$}
    \label{fig:simpleexample2}
  \end{figure}
}
```

4. A more fancy example: figure

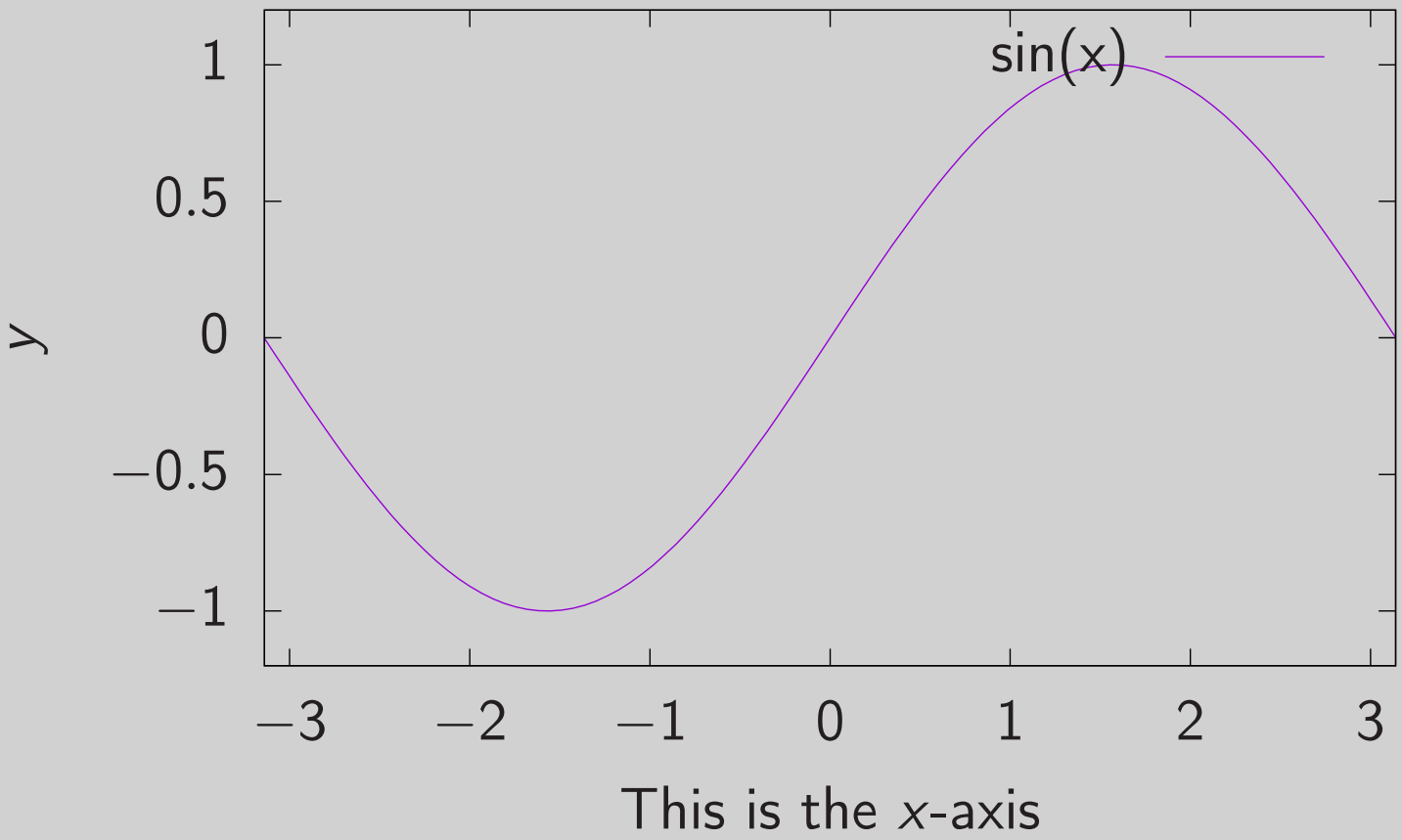


Figure: $f(x) = \sin(x)$

4. A better example: code

- Gnuplot code simple3.gp

```
set terminal epslatex size 10cm,6cm
set output 'eq3.tex'
set format x '$%.2f$'
set xrange [-pi:pi]
set yrange [-1.2:1.2]
set xlabel 'This is the $x$-axis'
set ylabel '$y$'
set xtics -pi,pi/2
set key at 3,-0.5
plot sin(x) linewidth 3 linecolor rgb '#001489'
```

- run gnuplot script

```
gnuplot simple3.gp
```

- LaTeX-code to produce next frame:

```
\frame{
  \frametitle{A better example: figure}
  \begin{figure}
    \centering
    {\graphicspath{{./Figures/LaTeX/}} \input{./Figures/LaTeX/eq3}}
    \caption{$f\left(x\right) = \sin\left(x\right)$}
    \label{fig:simpleexample3}
  \end{figure}
}
```

4. A better example: figure

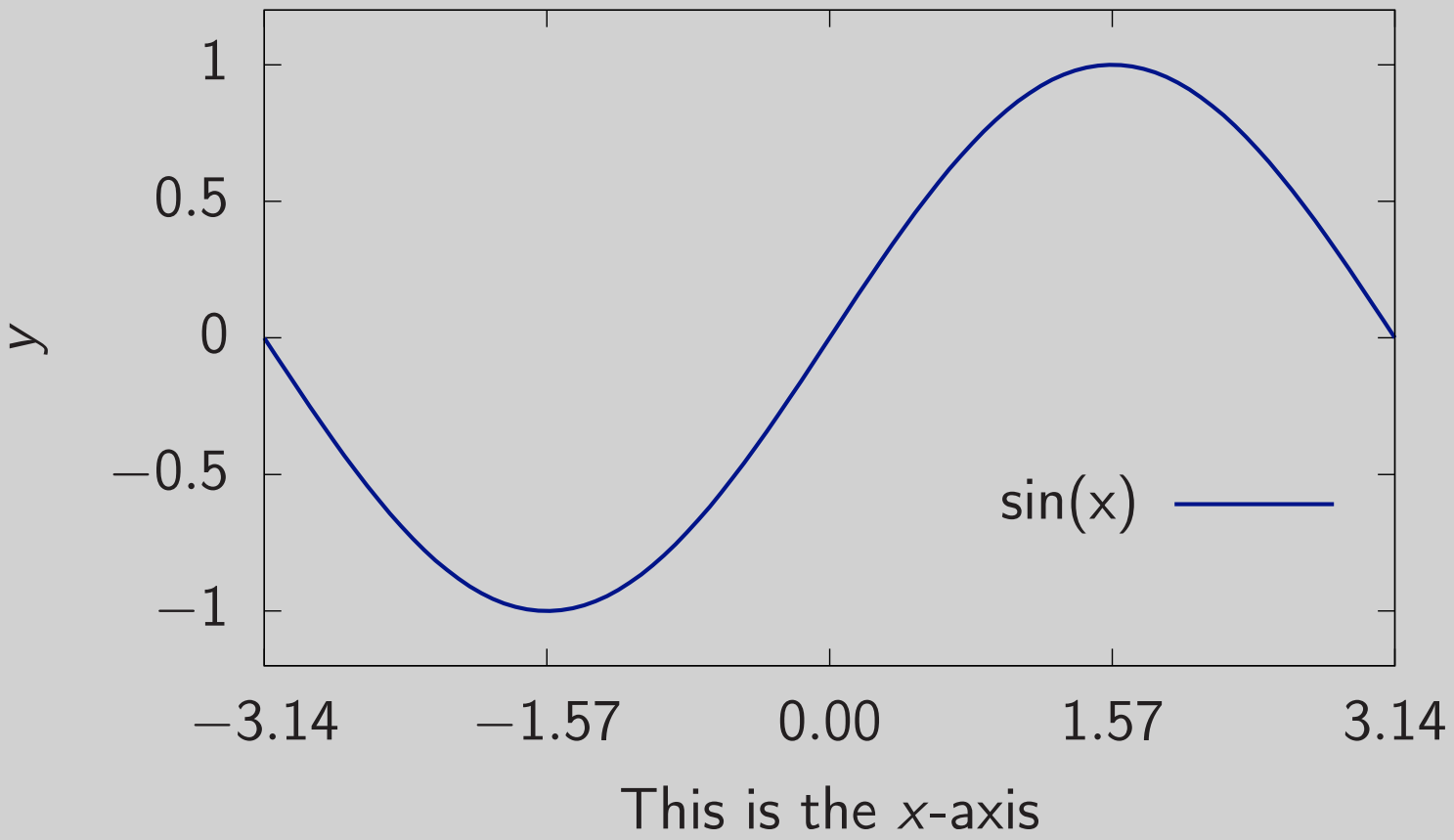


Figure: $f(x) = \sin(x)$

5. My personal experience

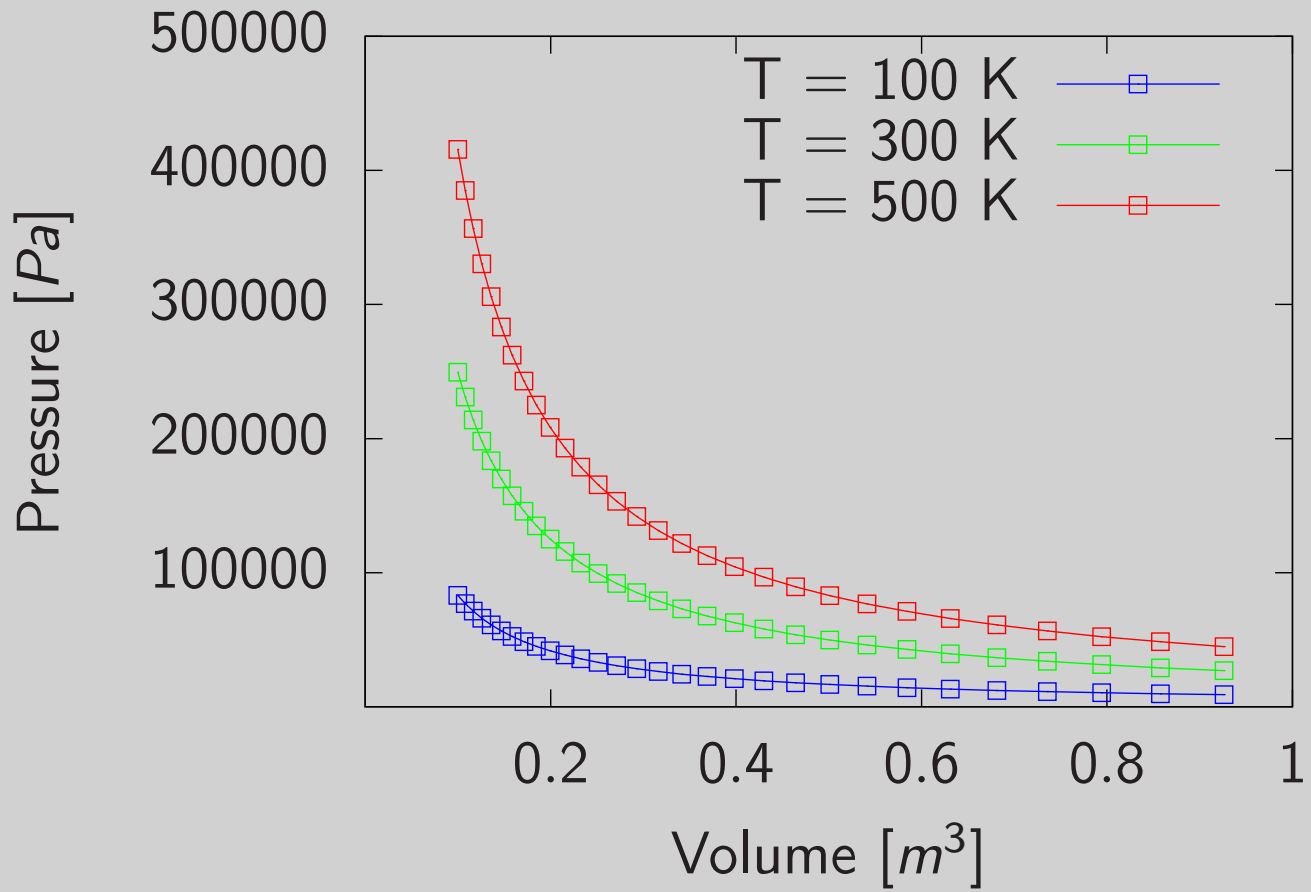
- I do not use gnuplot for simple stuff
- gnuplot is for presenting data
- almost all my plots are some sort of x,y plots
  - *automate regular plots*
  - *the rest is handled individually*

5. Gnuplot script with inputs

```
#-------------------------------------------------------------------------
# Purpose     : Import comma separated text file to GNUplot for plotting
#               x, y1, y2, ....
# Author      : Tore Haug-Warberg and Arne Tobias Elve
# Organization: Department of Chemical Engineering, NTNU, Norway
# Usage       : $ gnuplot -e \
#                 "OUTPUTFILE='file.end'; \
#                 INPUTFILE ='Data/data.txt'; \
#                 TERM = epslatex size 3cm,3cm color etc." main.gp
#-------------------------------------------------------------------------
  set terminal @TERM

  set datafile commentschars "#!%"
  set datafile separator ",;\t"
  set datafile missing "NaN"
  set output OUTPUTFILE

  set xrange [0.0:1.0]                                  # x-axis [xmin:xmax]
  set yrange [0.0:5.0e5]                                # y-axis [ymin:ymax]

  set ytics 1e5, 1e5, 5e5                          # start, increment, end
  set xtics 0.2, 0.2, 1.0

  set xlabel "Volume [m^3]"
  set ylabel "Pressure [Pa]"

# Column 1 along x-axis and columns 2, 3 and 4 along y-axis:
  plot INPUTFILE using 1:2 title "T = 100 K" with linespoints linestyle 1, \
       INPUTFILE using 1:3 title "T = 300 K" with linespoints linestyle 2, \
       INPUTFILE using 1:4 title "T = 500 K" with linespoints linestyle 3
```

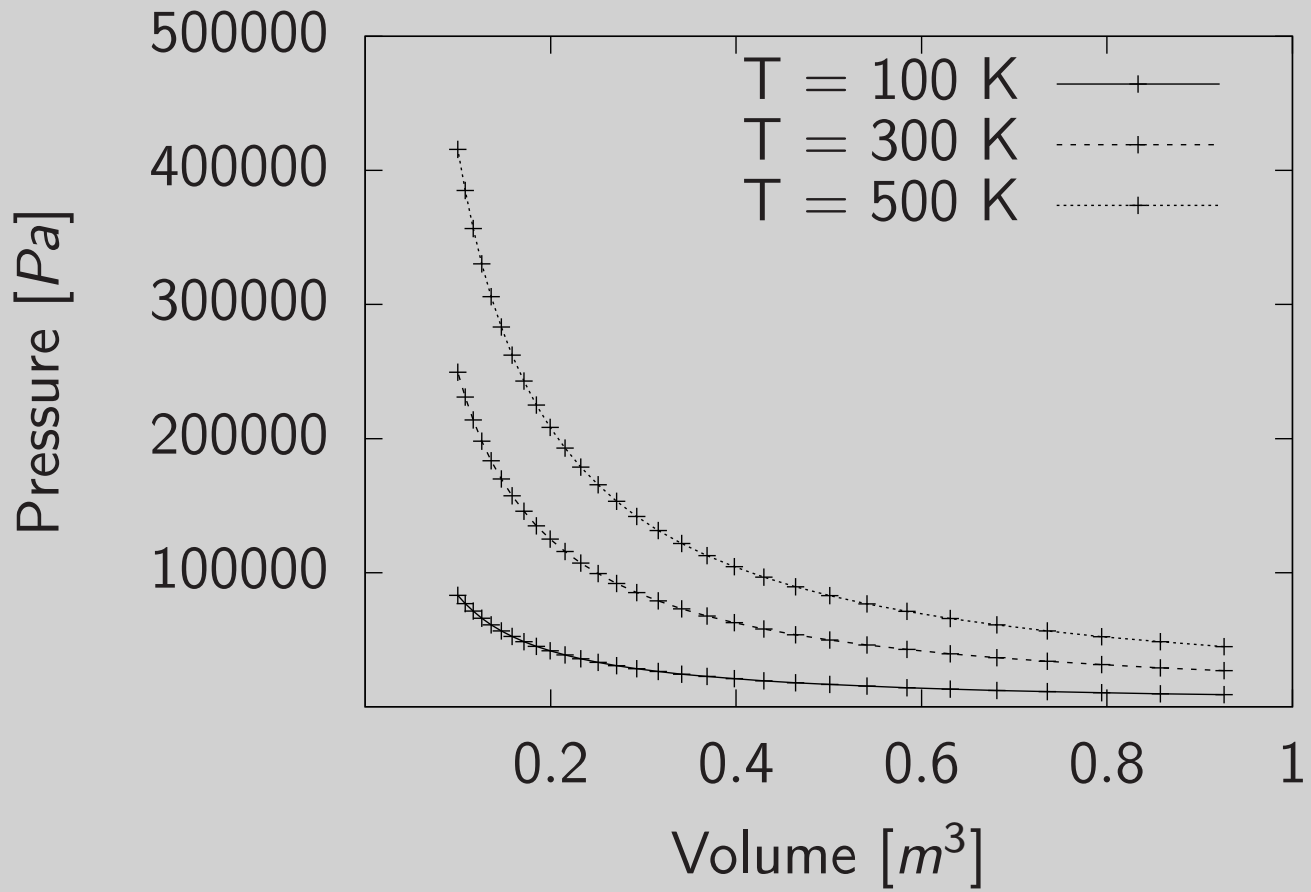
5. Normal

```
1 gnuplot -e "OUTPUTFILE='normal.tex'; \
2 INPUTFILE ='./data.txt'; \
3 TERM ='epslatex size 8cm,6cm colour solid'" main.gp
```



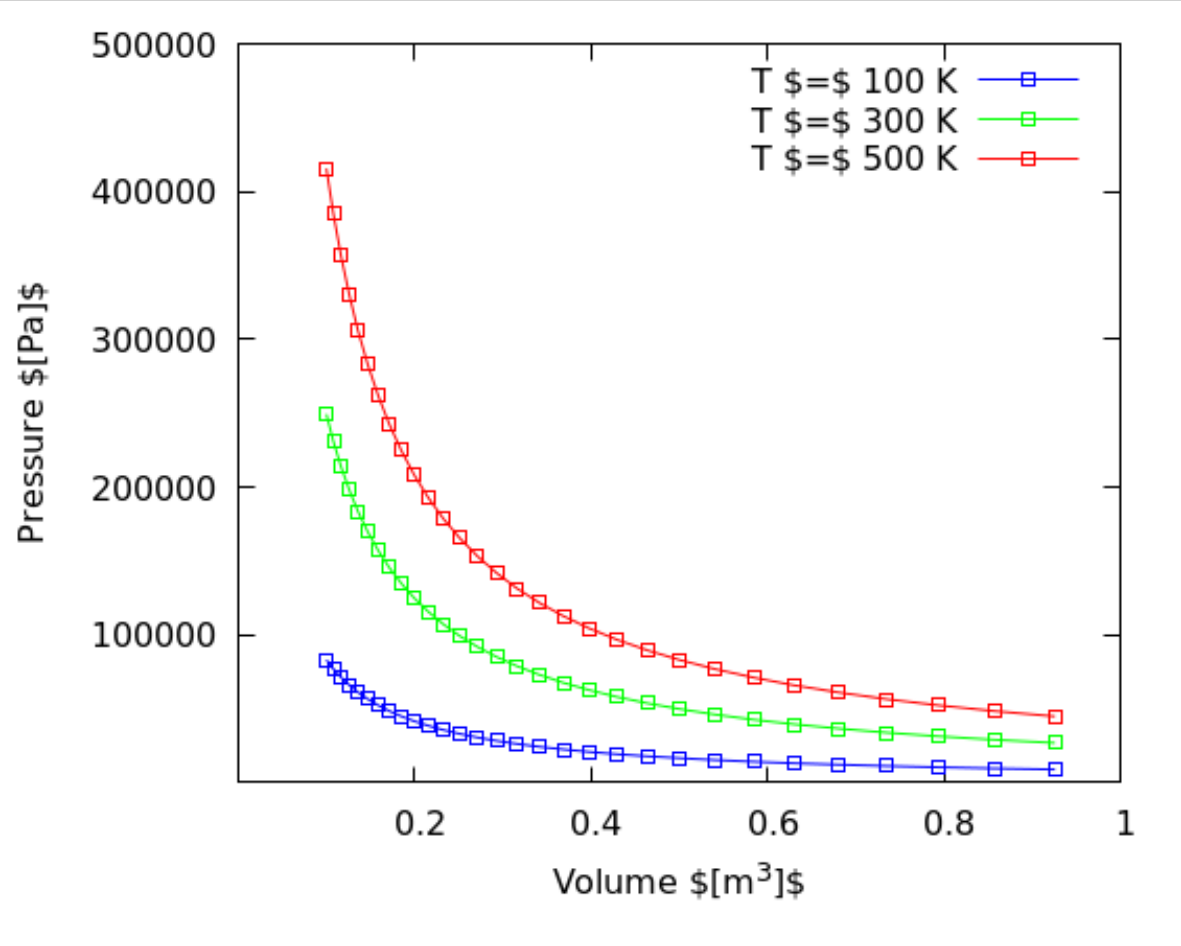
5. Gray-scale

```
1 gnuplot -e "OUTPUTFILE='blackWhite.tex'; \
2 INPUTFILE ='./data.txt'; \
3 TERM ='epslatex size 8cm,6cm monochrome'" main.gp
```



5. PNG

```
1 gnuplot -e "OUTPUTFILE='normalPNG.png'; \
2 INPUTFILE ='./data.txt'; \
3 TERM ='pngcairo size 20cm,16cm'" main.gp
```



6. Conclusions

- gnuplot is a domain specific language (dsl) for plotting
- Very flexible
- produce good looking figures
- Difficult to get stated
- Once you can use gnuplot you do not have to worry about plotting again.