Introduction to GNUPLOT

Katharina Fierlinger[1,2]
with Christian Alig[1]**, Klaus Dolag**[1] **and Tadziu Hoffmann**[1]

[1] Universitätssternwarte München, Scheinerstr. 1, 81679 Munich, Germany
[2] Excellence Cluster Universe, Boltzmannstr. 2, 85748 Garching, Germany
katharina.fierlinger@universe-cluster.de

7th & 8th of November 2013

```
katharina@fierli:~$ gnuplot

        G N U P L O T
        Version 4.4 patchlevel 3
        last modified March 2011
        System: Linux 3.2.0-24-generic

        Copyright (C) 1986-1993, 1998, 2004, 2007-2010
        Thomas Williams, Colin Kelley and many others

        gnuplot home: http://www.gnuplot.info
        faq, bugs, etc: type "help seeking-assistance"
        immediate help: type "help"
        plot window: hit 'h'

Terminal type set to 'wxt'
gnuplot>
```

Basics

GNUPLOT is a freely distributed command-line based interactive plotting program

- installing GNUPLOT
  Windows get .exe from sourceforge.net/projects/gnuplot/files
  Mac get .tar.gz from sourceforge.net/projects/gnuplot/files
  open shell, go to download directory, type ”configure”, ”make”, ”sudo make install”
  Linux use package management system
- starting GNUPLOT
  Windows GUI
  Linux, Mac open shell, type ”gnuplot”
- closing GNUPLOT
  Windows GUI
  Linux, Mac type ”exit” or ”quit”

```
katharina@fierli:~$ gnuplot

        G N U P L O T
        Version 4.4 patchlevel 3
        last modified March 2011
        System: Linux 3.2.0-24-generic

        Copyright (C) 1986-1993, 1998, 2004, 2007-2010
        Thomas Williams, Colin Kelley and many others

        gnuplot home: http://www.gnuplot.info
        faq, bugs, etc: type "help seeking-assistance"
        immediate help: type "help"
        plot window: hit 'h'

Terminal type set to 'wxt'
gnuplot>
```

Checking gnuplot on your laptop

Assignment 0

- Go to a command line.
- Start gnuplot by typing `gnuplot` .
- Check which gnuplot version you have. The loop syntax changed between the versions 4.4 and 4.6.
- Check which terminals you have installed by typing `set terminal` . You want (at least) wxt, png, latex and tikz. If you don't have them, install the missing packages.
- You can check your current terminal with `show terminal`
- If you type `test`, gnuplot will show you the line types and point types available for your current terminal type.
- Check if you have ImageMagick installed by typing `!convert` .

GNUPLOT syntax basics

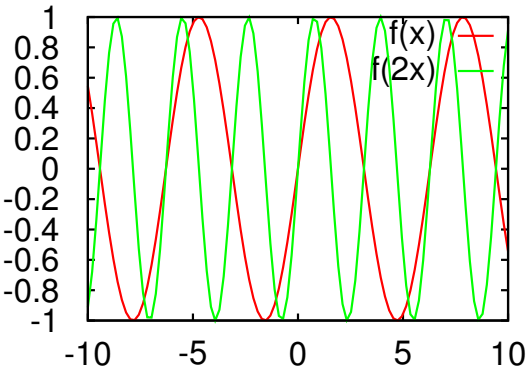
- GNUPLOT can display the manuals for its commands
  e.g. type `help plot` to get information on the `plot` command
- commands can be shortened
  e.g. `rep` instead of `replot`
  or `p` instead of `plot`
- `reset` restores the defaults
- If you want to use more than one GNUPLOT command in one line, you have to separate the commands by `;`
- GNUPLOT comments start with `#`
- shell commands (e.g. vi) in GNUPLOT start with `!`
- file names have to be enclosed in single or double quotes

Simple example

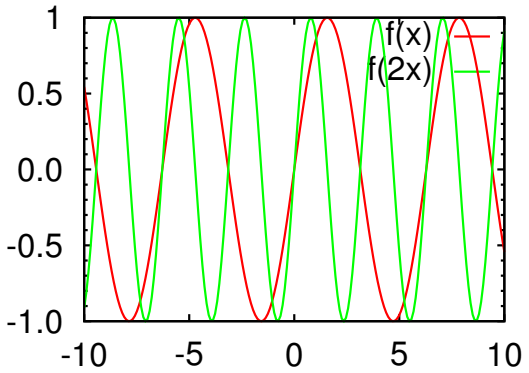
To plot a sine curve open GNUPLOT and type:

- `f(x) = sin(x) # define a function`
- `plot f(x) # plot this function`
- `replot f(2*x) # plot another function`

GNUPLOT will open a window with your plot.
The curves will not look smooth.
Fix this by increasing the sampling rate:

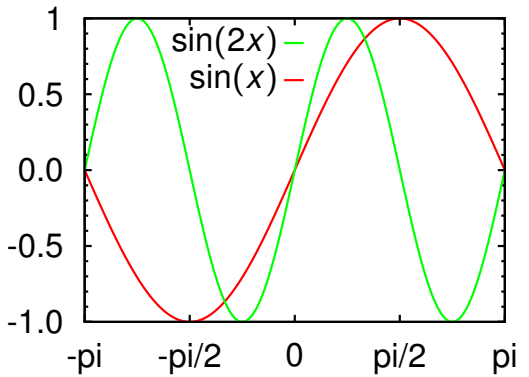


- `set samples 200 # sampling rate`
- `set ytics 0.5; set mytics 5`
- `rep # update plot`



Now let's set the x-range and x-tick-labels

- `set xrange [-pi:pi] # x range`
- `set xtics ("-pi" -pi, "-pi/2" -pi/2, 0, "pi/2" pi/2, "pi" pi)`



Scripting

For your scientific work you will prefer scripts.
Store the commands in a text file.

myscript.gnuplot

```
set term wxt                                          # select display type
set ytics 0.5                                         # define distance of labeled
                                                      #  tick marks on the y-axis
set mytics 5                                          # number of small
                                                      #  tick marks on the y-axis
set xrange [-pi:pi]                                   # set x range of the plot
set xtics ("-pi" -pi,"-pi/2" -pi/2,0,"pi/2" pi/2,"pi" pi) # custom labels
set samples 200                                       # increase sampling rate
set key at -pi/8,0.8 invert samplen 2                 # place and format
                                                      #  key of symbols
f(x) = sin(x)                                         # define a function
plot f(x) t "sin(x)", f(2*x) t "sin(2x)"              # plot two functions
pause -1  "hit ENTER to exit script"                  # don't close gnuplot
```

load this script in GNUPLOT by typing either

- `gnuplot myscript.gnuplot` on the command-line
- `load 'myscript.gnuplot'` in GNUPLOT

Tabulating a function

To tabulate the function f(x) and to store the result in a file called tablefile.txt use

- `set table "tablefile.txt"`
- `plot f(x)`
- `unset table`

Assignment 1

- Define a function.
- Choose a number of points.
- Write the tabulated function to a file.
- Plot this file. Use `plot` followed by the file name in quotes. For plot styles see next page. Try scatter plots and line plots.
- If you type `test`, gnuplot will show you the line types and point types available for your chosen terminal type.

Plotting data from files

GNUPLOT can also read data from files

- scatter plot:

```
plot 'data.txt' using 1:2
plot 'data.txt' using 1:2 with points
```

- example for the short format:

```
p 'data.txt' u 1:2 w p pt 1 lt 2 lw 2
notitle
```

- line plot:

```
plot 'data.txt' using 1:2 with lines
```

- multiple data series:
  use `replot` or separate by commas

```
plot 'data.txt' using 1:2, 'data.csv'
using 1:3
```

- set key:

```
plot 'data.txt' using 1:2 title "key"
```

```
1  2.1  3.2
2  2.2  4.3
3  2.1  3.6
4  2.1  4.8
5  2.0  3.2
6  2.2  4.5
7  2.2  2.5
8  2.0  6.3
9  2.1  1.1
```

data.txt

produced via FORTRAN IO or C++ IO

Parametric plot

If you want to draw a circle you can use:

- `set parametric`
- `set trange [-pi:pi]`
- `plot sin(t),cos(t)`
- `unset parametric`

In the parametric mode, the variable is “t” (instead of x).

Assignment 2

- Define two functions to plot an ellipse in parametric mode. You can use `a=1` if you want to define a variable `a` with a value 1. You can use such variables in your functions.
- Choose a number of points.
- Write the tabulated function to a file and plot this file.
- Use the commands from the next two pages. Try different line widths. For presentations you sometimes need thicker lines to make sure that the projector will display the lines. Add labels.

Labels, arrows, key

- place or hide key

  `set key top center`, `set nokey`
- set a title

  `set title "the title"`
- define axis labels

  `set xlabel "x [pc]"`, `set ylabel "y [pc]"`
- change the number format

  `set format x "%10.3f"`
- plot an arrow

  `set arrow from 0.5,0 to 0.5,1`
- define a label

  `set label "rarefaction wave" at 0.5,0`
- set border style

  `set border lw 3`

Other useful commands

- do math on columns
  $0 ... running index, $1 ... first column, $2 ... second column
  `plot './data.txt' u ($0*10):($2*10**($1)) w lp`
- blank lines in data files
  if blocks of data are separated by blank lines the `index` command can be used to plot individual blocks
- color, width and shape of lines/points
  `linetype / lt, pointtype / pt,`
  `linewidth / lw, pointsize / ps`
- logscale
  `[un]set logscale [xy],`
- select zoom
  `set xrange [0:10]` ... manually selected range of x-axis,
  `set yrange [*:*]` ... select zoom of y-axis automatically,
  `set autoscale` ... select zoom of any axis automatically

Polar plot

If you want to draw a rosette shaped curve you can use:

- `set polar`
- `set size square`
- `f(t)=a*sin(b*t)`
- `a=2`
- `b=2`
- `plot f(t)`
- `unset polar`

In the parametric mode, the variable is “t” (instead of x).

Assignment 3

- Draw some rosette shaped curves.

Multiple graphs

Sometimes you want to use several panels in a plot.

Assignment 4

- Open the file `2013_data/stars_3r.214.dat` in an editor and find out what is tabulated in the columns.
- Use the examples on the next pages and combine four plots. e.g. (1) x+z position (2) y+z position (3) x,age (4) y,age
- Label your plots.
- Now try a different number of panels e.g. 3 rows, 1 column.

Multiple graphs - panels

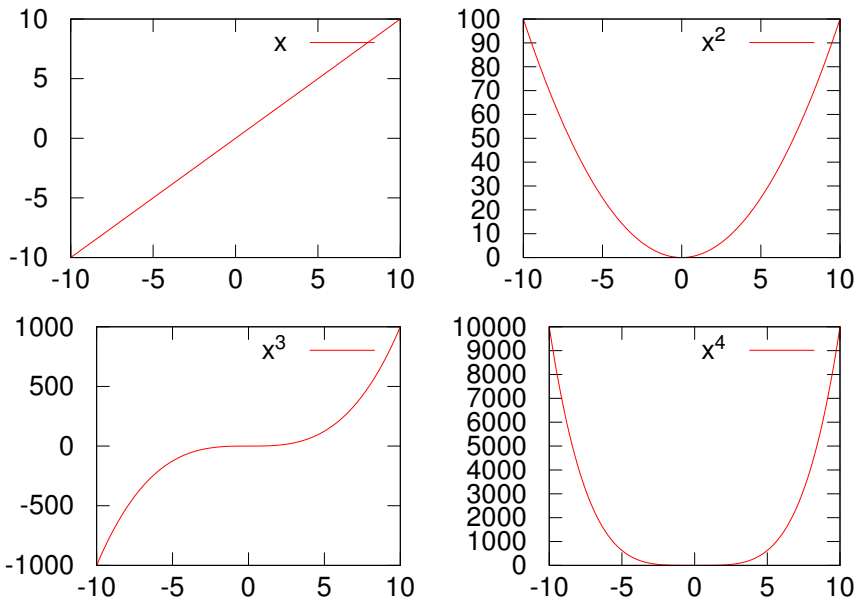
- stack several `plot` commands

  `set multiplot`
- scale the plot

  `set size`
- place the plot

  `set origin`
- leave multiplot mode

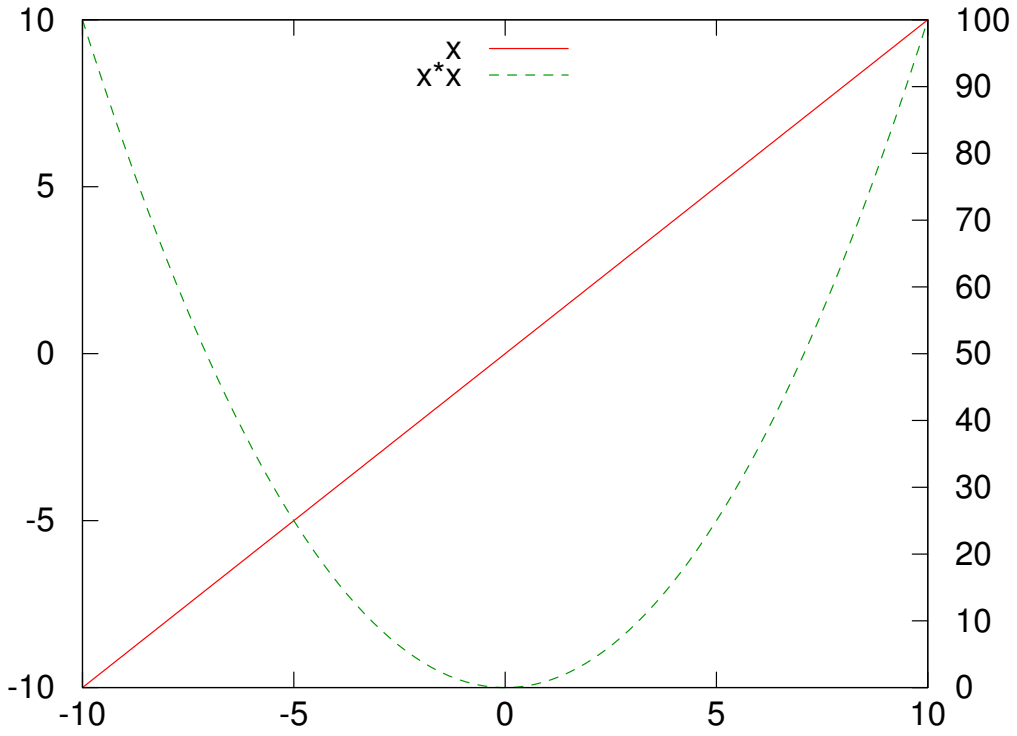
  `unset multiplot`




```
set multiplot
set origin 0,0
set size 0.5,0.5
plot x*x*x t "x^3"
set origin 0,0.5
plot x t "x"
set origin 0.5,0.5
plot x*x t "x^2"
set origin 0.5,0
plot x*x*x*x t "x^4"
unset multiplot
```

Multiple graphs - y1 and y2 axis

```
set ytics nomirror
set y2tics 0, 10
set key top center
plot x axis x1y1, x*x axis x1y2
```



Fitting data

Gnuplot can also fit your data with functions.

Assignment 5

- Use your ellipse from assignment 2. You can add scatter with `awk` and the `rand()` function. e.g.

```
awk '{if ($3=="i"){$1=$1+(rand()-0.5)*0.01; $2=$2
    +(rand()-0.5)*0.1; print $0}}' ellipse.txt >
    ellipse.scatter
```

- Note: if you use German language settings in your shell, you might need to say `LANG=C` or `LANG=en_US.UTF-8` on the shell to avoid conflicts with decimal points or commas.
- Plot it.
- Try to fit it.
- Also plot the fit.
- A possible solution can be found in `ellipse.solution`
- Now try to tilt your ellipse by 45 degrees and try to fit it again.

awk

awk commands can have three blocks:

```
awk 'BEGIN{i=0}{i++}END{print "number of lines:",i}'
    2013_data/gas_3r.214.dat
```

- The “BEGIN” block contains commands that should be executed directly after awk starts. Typically initializations.
- The main part loops over all lines in the input file
- The “END” block is read directly before awk exits.

All blocks are enclosed in curly brackets and commands in a block are separated by ; . You can also use loops in awk:

```
awk 'BEGIN{srand();i=0;j=0;k=0;}
{while(j<10){
    while(k<10){
    T=rand()*1e4;print j,k,T;k=k+1;}
    k=0;print "";j=j+1;}
}' 1line
```

Fitting data

define a power law

```
f(x)=a*x**b+c
```

fit your data - you might need to set initial values for a,b,c

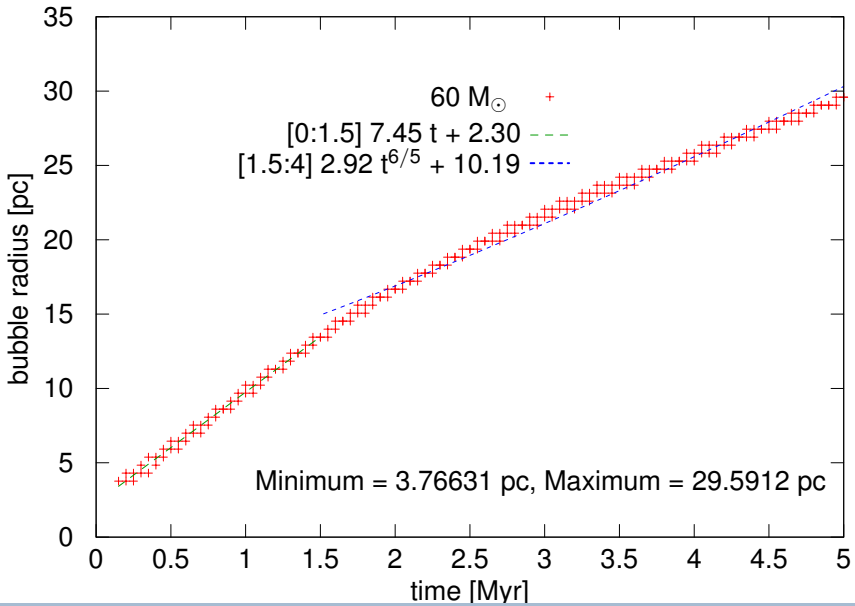
```
fit [1.5:*] f(x) 'data.txt' using 1:2 via a,b,c
```

clip the fit to the fitted area

```
fcut(x) = x <1.5 ?  1/0 :  f(x)
```

display the fit

```
rep fcut(x) t sprintf("[1.5:4] %g x^{%g} + %g", a,b,c)
```



Drawing error bars

```
plot 'data.txt' u 0:1:2:3 w yerrorbars
```

uses

- column 1 as y values,
- column 2 as lower end of the vertical error bar and
- column 3 as the upper end of the vertical error bar.

horizontal error bars are created with `xerrorbars`

Output formats

`set term` latex, tikz, eps, png ...

Example:

```
set terminal png
set output "plot.png"
set multiplot
...
unset multiplot
set term wxt
```

You can customize the terminal:

```
set terminal png nocrop enhanced font FreeSansBold 18
size 800,600
```

File with coordinates and temperature

Assignment 6

The file `gas_3r.214.dat` contains the (x,y,z) coordinates of a point and the temperature. Use the example on the next two pages and plot color coded temperatures at the xy position of the point. The example uses a scale running from red to blue, but feel free to test other palettes.

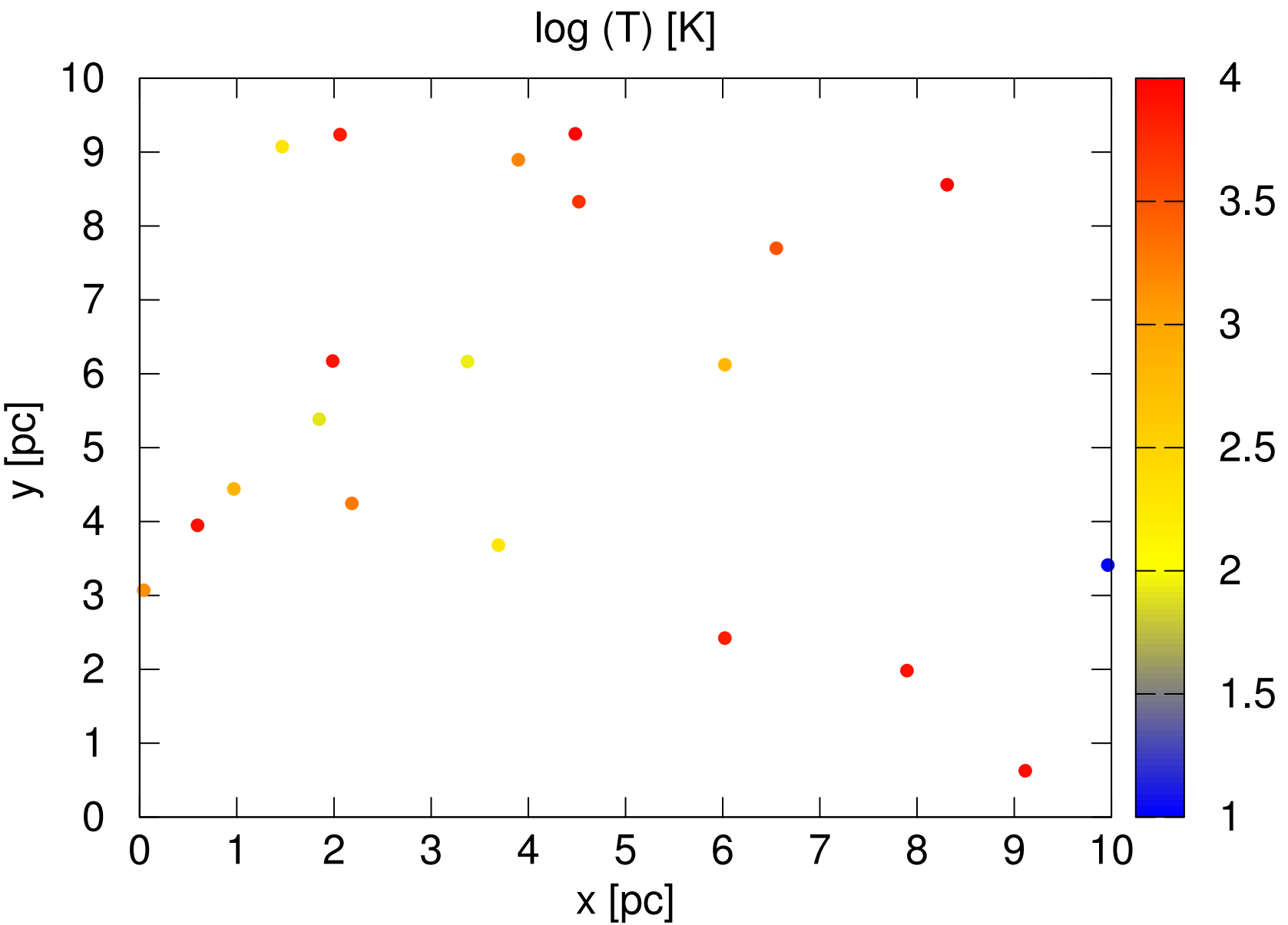
```
#awk 'BEGIN{srand();ind=10;}
# {while (ind > 0) {
#   x=rand()*10;y=rand()*10.;
#   z=rand()*10;T=rand()*1e4;
# print x, y, z, T; ind=ind-1}}'
# x y z T
0.59582 3.94872 6.1967 7998.84
6.5519 7.6995 5.57703 3189.28
2.18446 4.24577 8.66912 1837.28
9.1135 0.627003 8.04655 8363.9
1.98704 6.17223 6.72384 7623.37
6.02263 2.42297 2.81232 6649.25
3.89771 8.89551 6.84724 1554.17
0.970098 4.44185 4.21257 690.311
2.06219 9.2371 7.89356 7030.08
4.48321 9.24652 6.26523 9672.72
9.96331 3.41045 9.51404 10.3592
1.4666 9.07461 7.048 203.987
3.69213 3.68113 8.76336 195.501
8.30955 8.55802 4.67305 9369.63
1.84836 5.38461 9.17682 78.5009
7.89623 1.98271 3.33848 8015.49
3.37396 6.16493 3.95816 85.7048
0.041832 3.0696 0.683514 1341.57
4.52064 8.32836 4.73786 4954.87
6.02226 6.12297 8.839 650.453
```

data file with x[pc] y[pc] z[pc] T [K]

Script for colored dots

```
set title "log␣(T)␣[K]"
# data x y z T
set palette defined ( 1 "blue", 2 "yellow", 3 "orange
    ", 4 "red" )
set cbrange [1:4] # 10 to 1e4 Kelvin
#2d
set xlabel "x␣[pc]"
set ylabel "y␣[pc]"
plot 'xyzT.txt' u 1:2:(log($4)/log(10.)) with points
     palette pt 7 notitle
pause -1 '2d,␣press␣ENTER␣to␣proceed␣to␣3d'
#3d:
set ticslevel 0      # zero-point of z axis
set zlabel "z␣[pc]"
splot 'xyzT.txt' u 1:2:3:(log($4)/log(10.)) with
    points palette pt 7 notitle
```

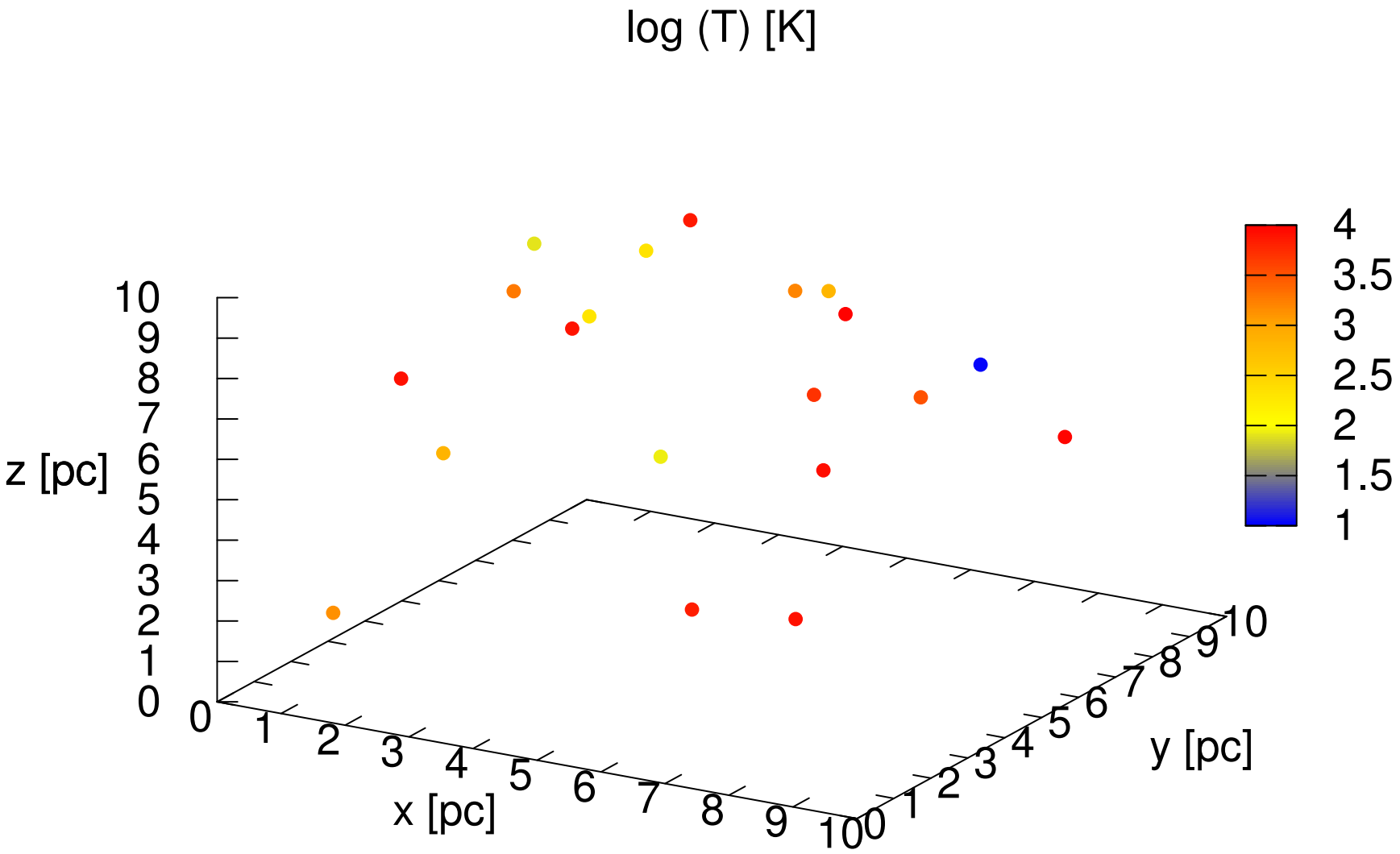
2d, `plot`



3d plots

- Use `splot` to produce a plot with the positions of the stars or the gas data in the 2013_data folder. Click on the plot and change the viewing angle.
- Use awk and plot only a certain temperature range. An example is on the page after the next page.

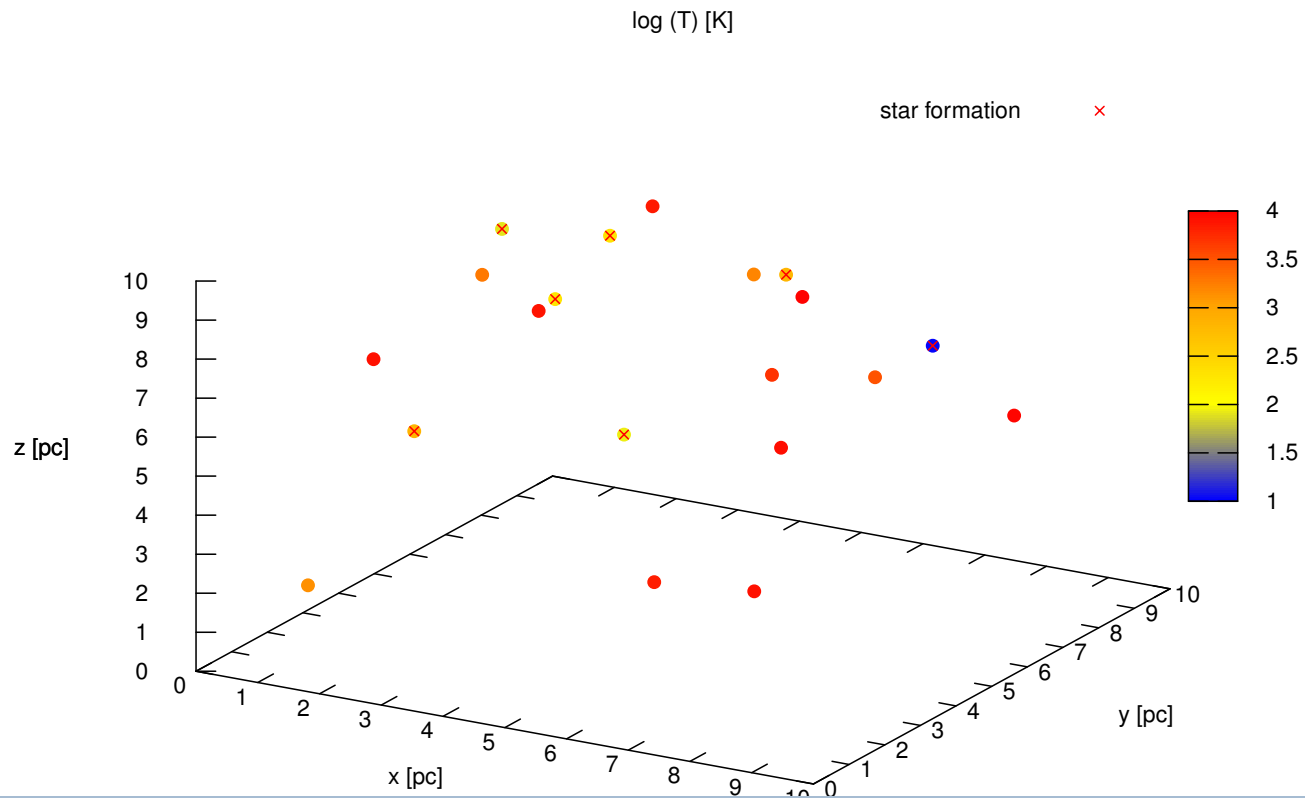
3d , splot



3d , splot and awk

```
rep "<awk 'BEGIN{Tmax=1000}{if ($4 < Tmax){print $1,$2,$3,$4}}' xyzT.txt" u 1:2:3 lt 1 t "star formation"
```

```
rep "<awk 'BEGIN{MAXcoldfraction=0.6;rhoSFR=1;Tmax=1000}{if (NF==6){if (($5 > MAXcoldfraction) || (($6 > rhoSFR) && ($4 < Tmax))){print $1,$2,$3,$4}}else{if ($4 < Tmax){print $1,$2,$3,$4}}}' xyzT.txt" u 1:2:3 lt 1 t "star formation"
```



Typical grid code data

A file contains the (x,y) coordinates of a cell and a temperature.
The data is arranged in blocks
The temperature will be plotted color coded on a scale running from red to blue.

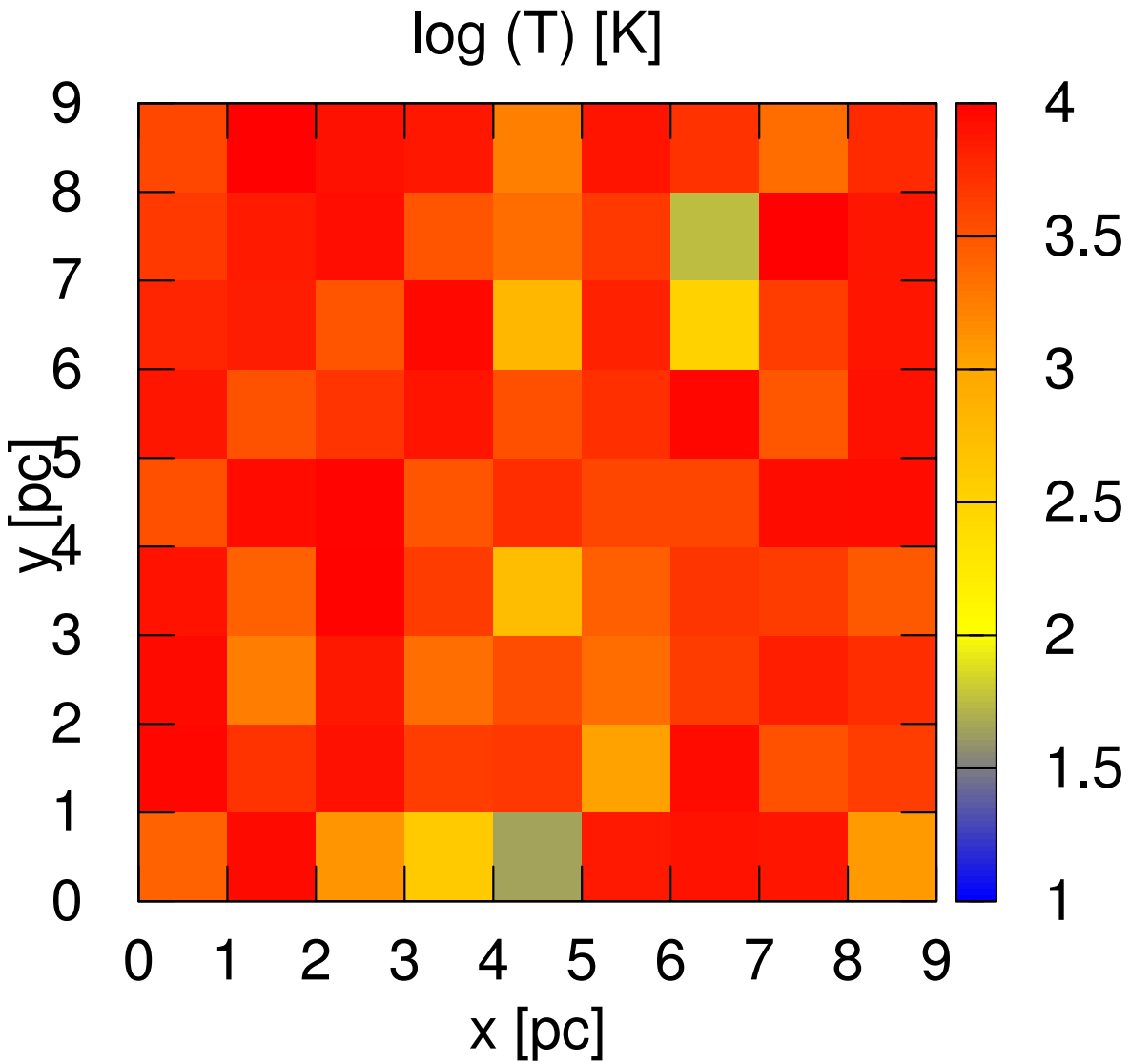
```
# x y T
#awk 'BEGIN{srand(); i=0; j=0; k=0;}
# {while (j < 10) {
#   while (k < 10) {
#   T=rand()*1e4;
# print j, k, T; k=k+1;}
# k=0; print " "; j=j+1;}}'
0 0 2539.97
0 1 9259.75
0 2 8698.62
0 3 7780.29
0 4 3295.49
0 5 7360.98
0 6 5922.8
0 7 4545.35
0 8 3687.71
0 9 9291.88

1 0 8663.94
1 1 4920.67
1 2 1738.12
1 3 2604.78
1 4 8554.96
1 5 3204.42
1 6 6657.76
1 7 7011.96
1 8 9935.54
1 9 6665.75
```

data file with x[pc] y[pc] T [K]

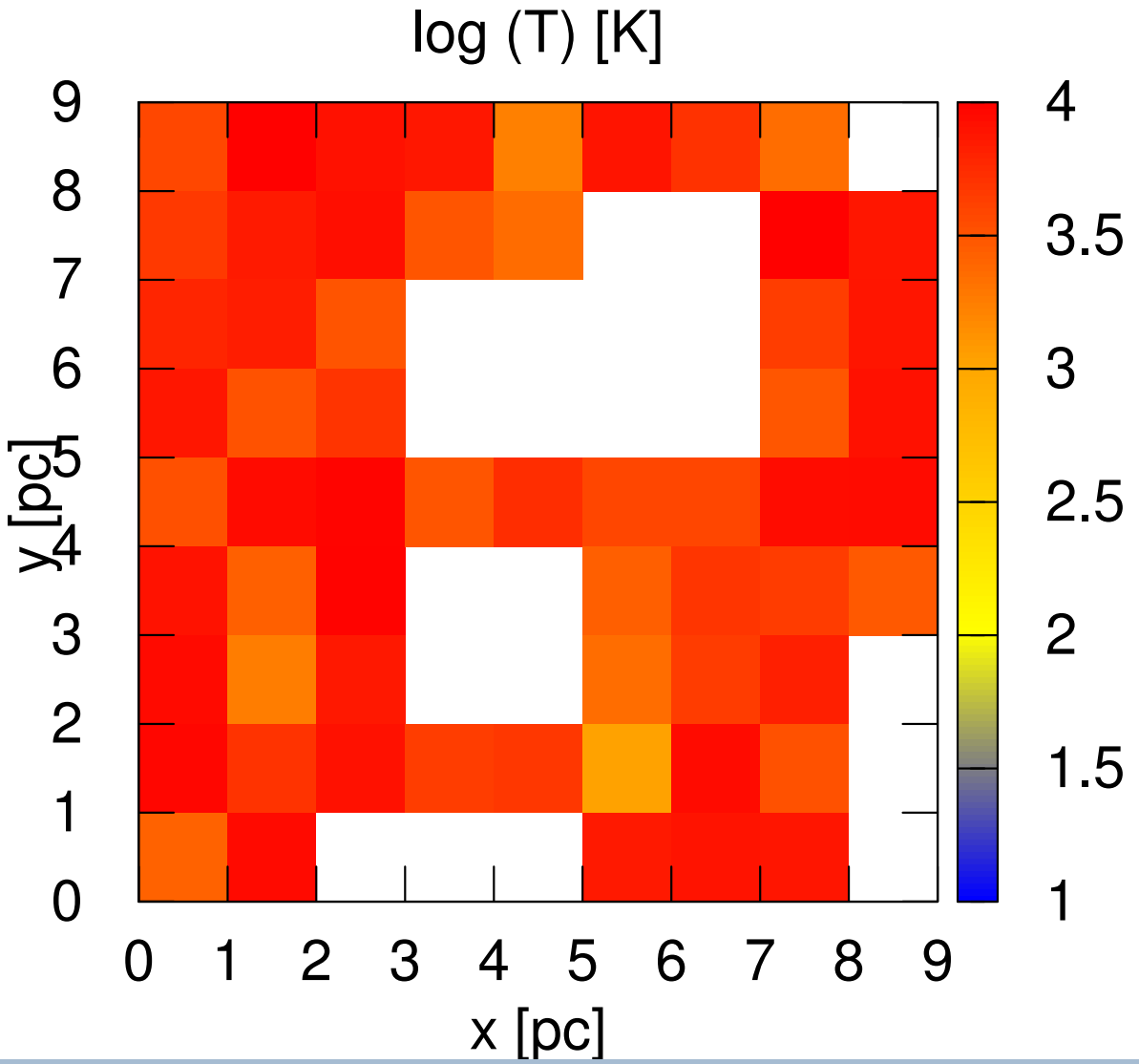
Grid code data: color maps with pm3d

```
set title "log␣(T)␣[K]"
# 2d data x y T
set palette defined ( 1 "blue", 2 "yellow", 3 "orange
    ", 4 "red" )
set cbrange [1:4] # 10 to 1e4 Kelvin
set xlabel "x␣[pc]"
set ylabel "y␣[pc]"
set size square
set view 0,0
set pm3d at b corners2color c1 clip4in map
unset surface
set term tikz
set output "Txy.tikz"
splot 'gridxyzT.txt' u ($1):($2):(log($3)/log(10.))
    notitle
set output "TxyCLIP.tikz"
splot [:][:][3:4] 'gridxyzT.txt' u ($1):($2):(log($3)
    /log(10.)) notitle # clip to [1e3:1e4] K
```

Grid code data: color maps with pm3d



Grid code data: color maps with pm3d zrange [1.e3:1.e4]



Movies

- produce png images for all snapshots
- combine with libmagick

  `!convert -delay 20 -loop 1 frame_????.png movie.gif`

  (inside GNUPLOT , otherwise no exclamation mark)

Movies

Assignment 9: Make a movie from your 3d positions.

- If you use gnuplot 4.6 you can use a do loop:
  To rotate the view, try e.g.
  `do for [t=0:18]{set view t*5,30,1,1;rep;pause -1}`.
  If you want to store the snapshots use:
  `do for [t=0:18]{outfile=sprintf('view%03.0f.png',t);`
  `set view t*5,30,1,1;set output outfile; rep}`
- In earlier versions you have to place your code in two files and to use “reread”:
  File 1: `t=0; tmax=18;load "file2.gnuplot"`
  File 2: `t=t+1;outfile = sprintf('view%03.0f.png',t);`
  `set view t*5,30,1,1;set output outfile; plot`
  `data.txt; if(t<tmax) reread;`

Loops in Gnuplot 4.6

Gnuplot 4.6 can handle loops:

```
#simple loop example - (slowly) rotates a 3d plot
set term x11 #package gnuplot-x11 has to be installed
splot sin(x)*sin(y) #any 3d plot
#set term png        #for movies: store as png
do for [t=0:45] {   #Gnuplot 4.6 or higher
 #t: loop variable, running from 0 to 45. step size:1
 # outfile = sprintf('franz%03.0f.png',t)  #numbered
     output files franz001.png, franz002.png ...
 # set output outfile    #use these filenamens
 set view 60, 2*t, 1, 1 #define viewing angle
 pause 0.2;             #slow down rotation in x11
 rep                    #update plot
}
#libmagick can convert these files into an animated
    gif: "convert -delay 20 -loop 1 franz???.png movie
    .gif"
```

Optimizing plots

1. Focus on the purpose of the figure.
   What do you want to show?
2. Keep it simple and efficient.
   Which variable is independent (plot on x axis) or dependent (y axis)? Choose good units for the axes. Scale the axes to make good use of the figure's area.
3. Explain what you plot: label the axes, find a good title, add a key to symbols and write a clear and complete caption [not just re-stating what's on the axes]. The plot should be self explanatory – the intended audience should understand it even without reading the text of your paper.
4. Show you figure to a colleague – optimally somebody not directly working with you – to check if it is clear. Try to make your plot as simple as possible.

Checklist for good plots

- enough information in well chosen title / caption ?
- content of labels (e.g. units)?
- content of key of symbols?
- too much information for a single graph?
- plot type suited for purpose?
  scatter vs. line graph; error bars; fits
- is *x* the independent variable and *y* the dependent variable?
- large enough font size of labels?
- sufficient line width?
  (e.g. a few pixels for presentations with data projectors)
- optimized number of tick marks, minor tick marks?
- plot looks ”empty”? zoom ... make good use of the plot area
- plot format (eps, png, tikz) suited for the purpose?

Manuals

- GNUPLOT homepage www.gnuplot.info
- Introduction to GNUPLOT and Not so FAQ and Solutions http://t16web.lanl.gov/Kawano/gnuplot/index-e.html
- Download e.g. Windows version http://sourceforge.net/projects/gnuplot/files/gnuplot/4.6.0/

`test` shows linetypes