SKILLPILLS

Skill Pill: gnuplot

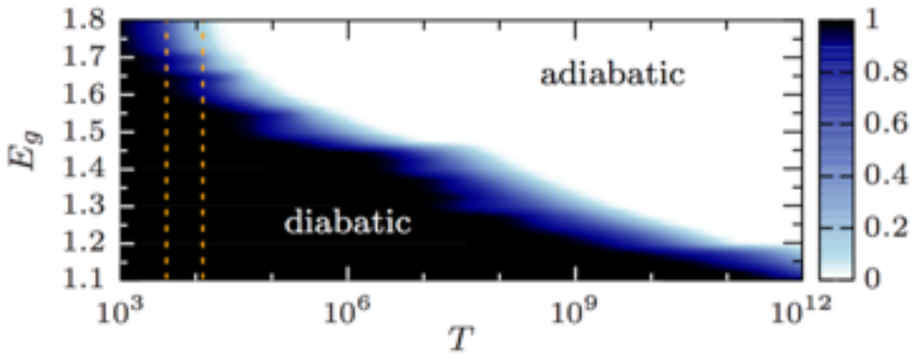
Day 2

OIST

Why do I use gnuplot?

SKILLPILLS

```
set term epslatex standalone color dashed \
    size 6.5 cm, 4 cm font 8 header '\usepackage{amsmath}'
set out 'niceplot.tex'

set xr[0.0:1.0]
set xtics 0.2

set yr[0:2]
set ytics 0.5

set key samplen 1

set key at graph 0.5, 0.75 reverse Left top

 plot \
 'pop' u ($1/TT):2 every 20  title '\small $P_1$' lw 2,  \
 '' u ($1/TT):3 every 20  title '\small $P_2$' lw 2, \
 '' u ($1/TT):4 every 20  title '\small $P_3$' lw 2
```



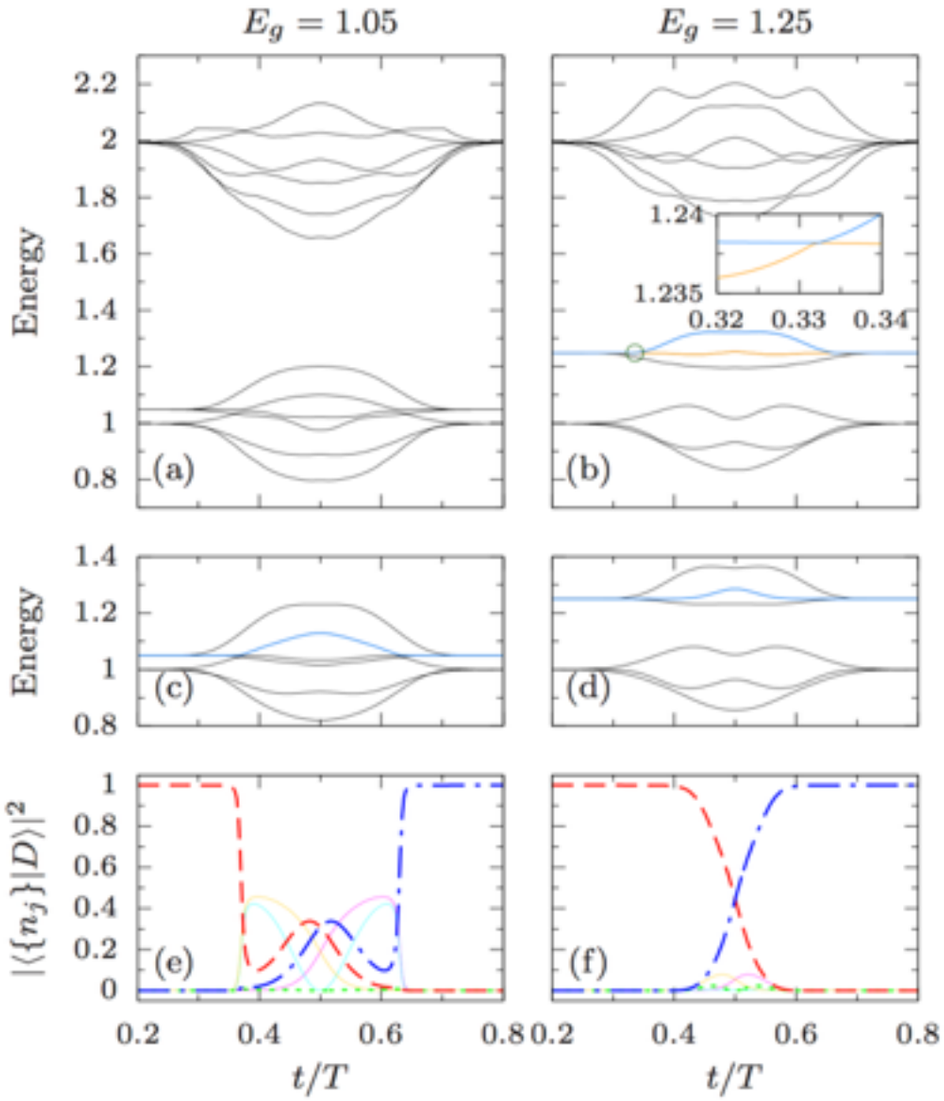


FIG. 5. Probability of transition $p_{i\to j}$ at the crossing between the eigenstates shown in the insets of Figs. 3 and 4 for different total times T and energies E_g. Dashed vertical lines indicate the total times used in Fig. 2.

Round I

Getting help

- *help*
- google is your friend
- display the current terminal styles, colors, …
  test
- list all colors defined in gnuplot
  show colornames

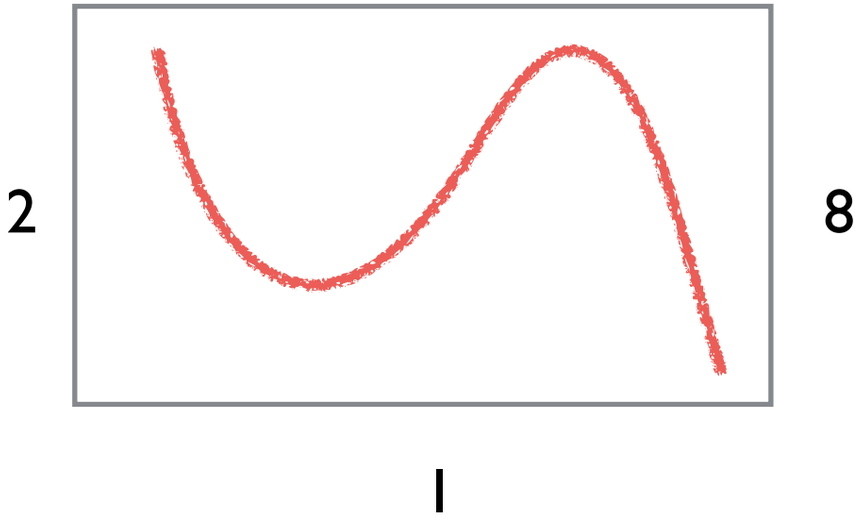
Scripting!

- run a script
  load 'script.gp'
- *# you can write comments*
- more advanced stuff
  call (like *load* but passing arguments)
  eval / macros (use strings as commands)
  word (returns *n*th word in a string)

Border

- *set border num lw ... ls ... lt ...*
- The *num* depends on which borders we want to draw

4

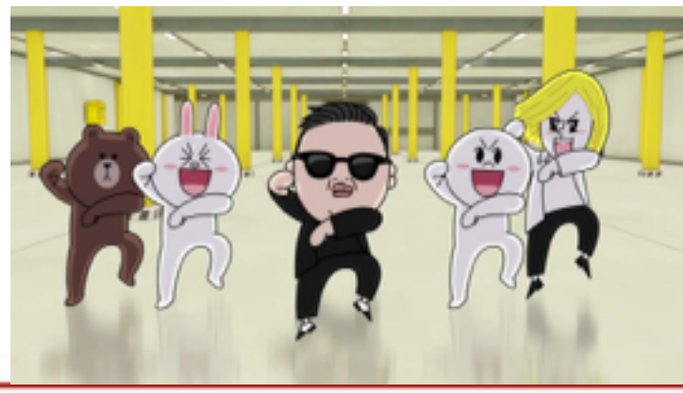


- *set ytics nomirror*

Defining line stlyes

- *plot 'data' lw 2 lc 'olive' dt 3*

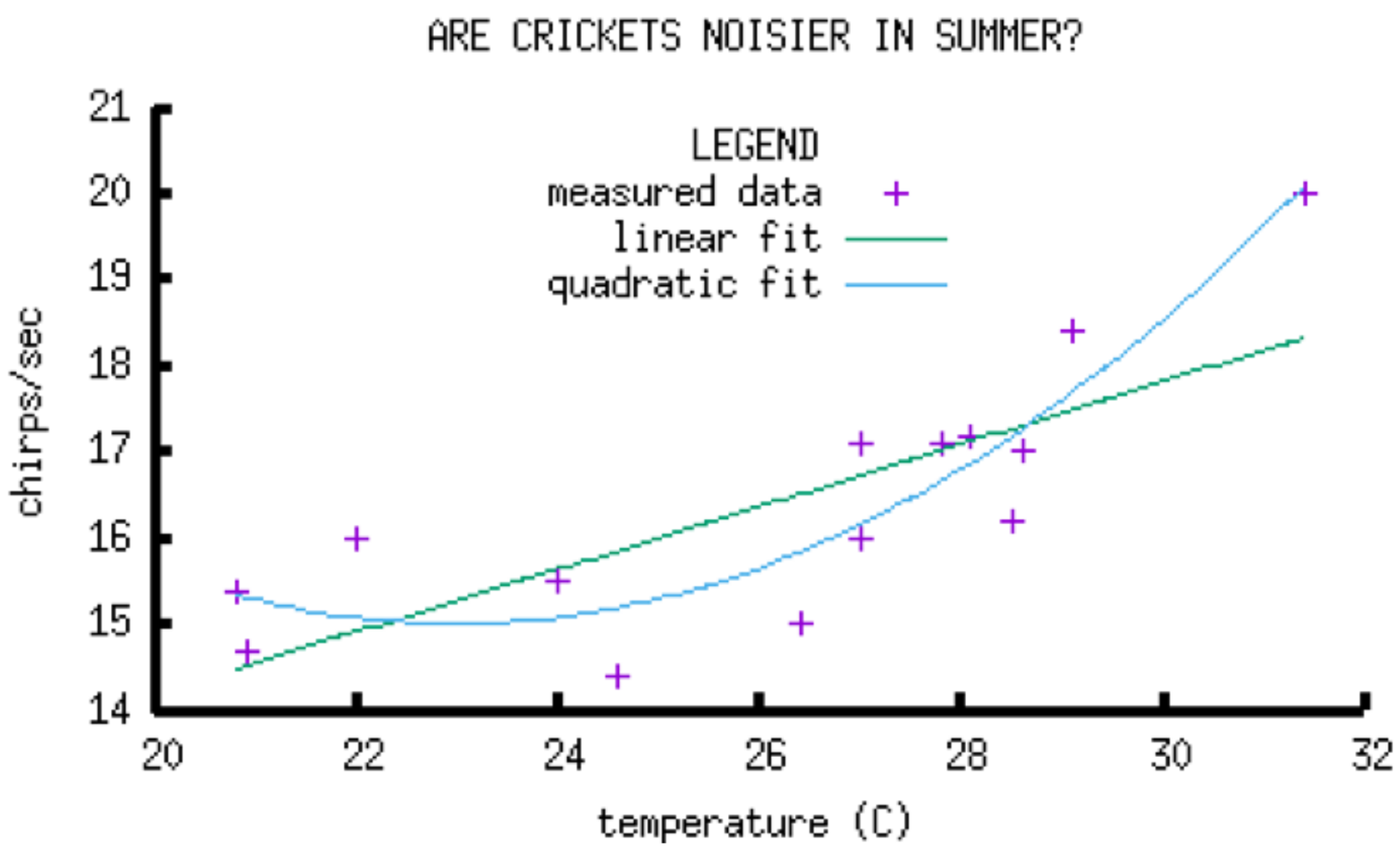
- we can define line types to reuse
  set linetype ... (default styles, not affected by *reset*)
  set linestyle …. / set style line



Fits

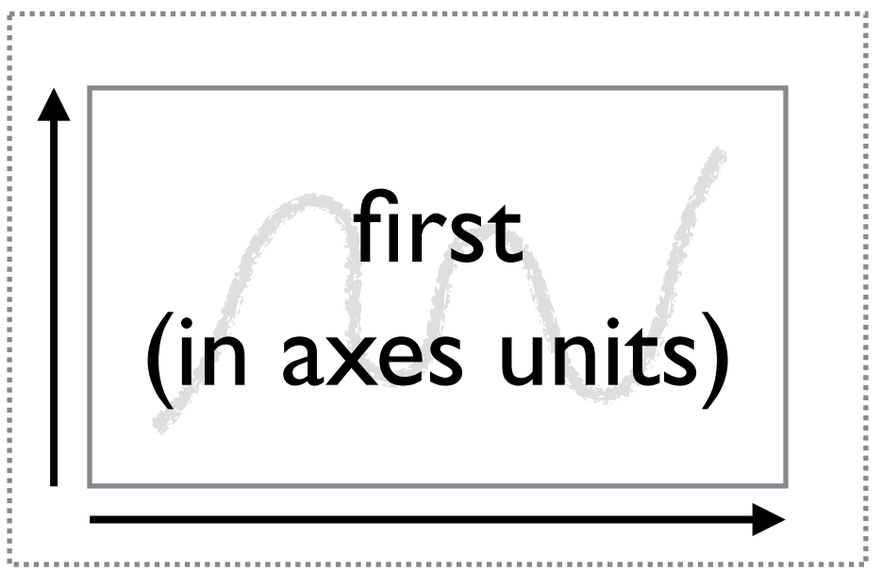
- *f(x) = a + b*x + c*x**2*
  fit f(x) 'measured.dat' using 1:2 via a,b,c

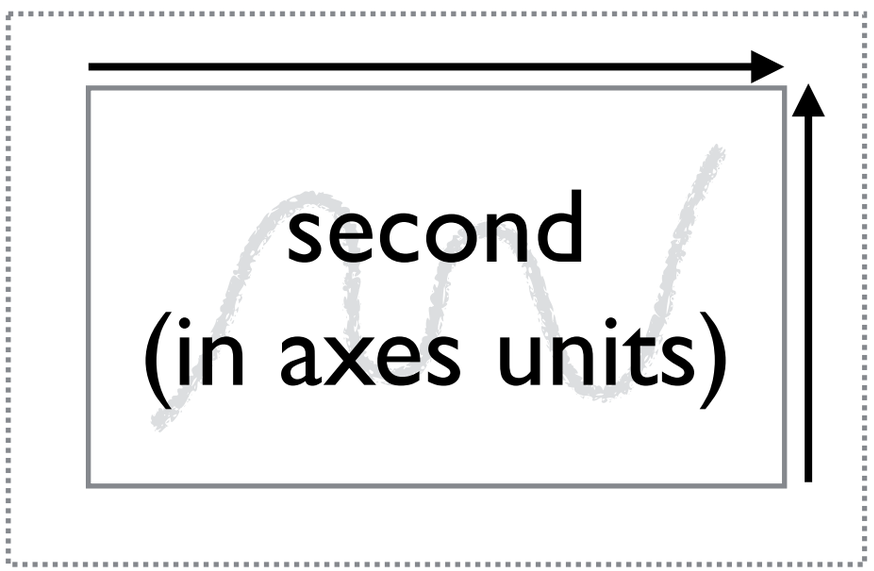
- Using can be used to operate on the values of columns
  *... using 1:(2*f($2))*

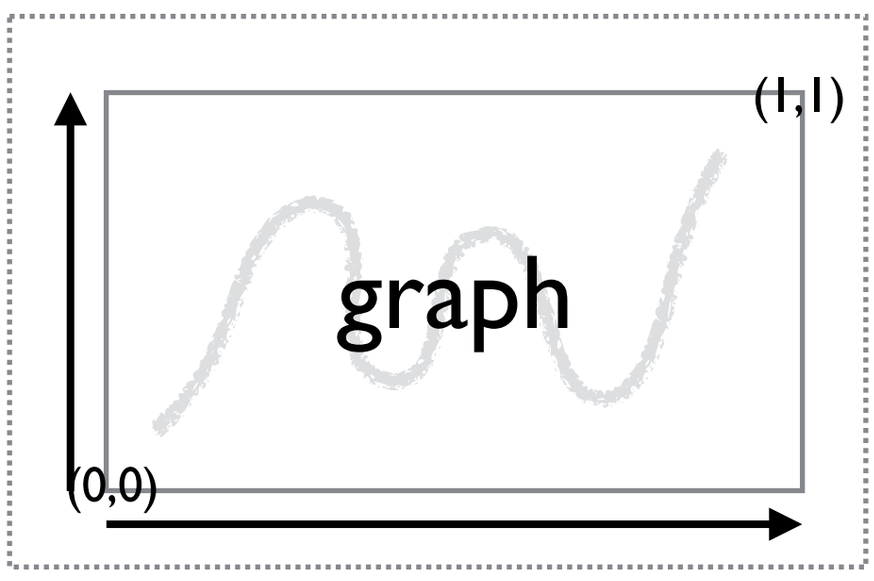

ARE CRICKETS NOISIER IN SUMMER?
LEGEND
measured data
linear fit
quadratic fit
chirps/sec
temperature (C)
14
15
16
17
18
19
20
21
20
22
24
26
28
30
32

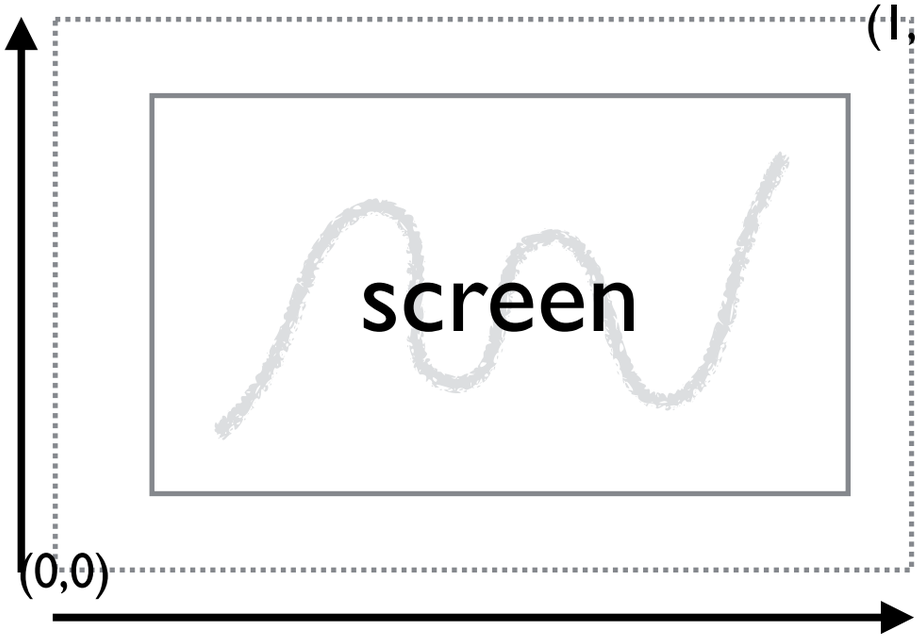
Round 2

Coordinates









Adding labels (text)

SKILLPILLS

Label

Arbitrary labels can be placed on the plot using the **set label** command.

Syntax:

```
set label {<tag>} {"<label text>"} {at <position>}
          {left | center | right}
          {norotate | rotate {by <degrees>}}
          {font "<name>{,<size>}"}
          {noenhanced}
          {front | back}
          {textcolor <colorspec>}
          {point <pointstyle> | nopoint}
          {offset <offset>}
          {boxed}
          {hypertext}
unset label {<tag>}
show label
```

Adding arrows (and lines)

SKILLPILLS

Arrow

Arbitrary arrows can be placed on a plot using the **set arrow** command.

Syntax:

```
set arrow {<tag>} from <position> to <position>
set arrow {<tag>} from <position> rto <position>
set arrow {<tag>} from <position> length <coord> angle <ang>
set arrow <tag> arrowstyle | as <arrow_style>
set arrow <tag> {nohead | head | backhead | heads}
                {size <headlength>,<headangle>{,<backangle>}}
                {filled | empty | nofilled | noborder}
                {front | back}
                {linestyle <line_style>}
                {linetype <line_type>} {linewidth <line_width>}
                {linecolor <colorspec>} {dashtype <dashtype>}

unset arrow {<tag>}
show arrow {<tag>}
```

Adding objects

SKILLPILLS

Object

The **set object** command defines a single object which will appear in all subsequent 2D plots. You may define as many objects as you like. Currently the supported object types are **rectangle**, **circle**, **ellipse**, and **polygon**. Rectangles inherit a default set of style properties (fill, color, border) from those set by the command **set style rectangle**, but each object can also be given individual style properties. Circles, ellipses, and polygons inherit the fill style from **set style fill**.

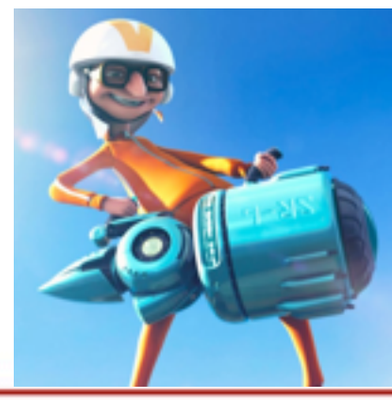
Syntax:

```
set object <index>
    <object-type> <object-properties>
    {front|back|behind} {clip|noclip}
    {fc|fillcolor <colorspec>} {fs <fillstyle>}
    {default} {lw|linewidth <width>} {dt|dashtype <dashtype>}
unset object <index>
```

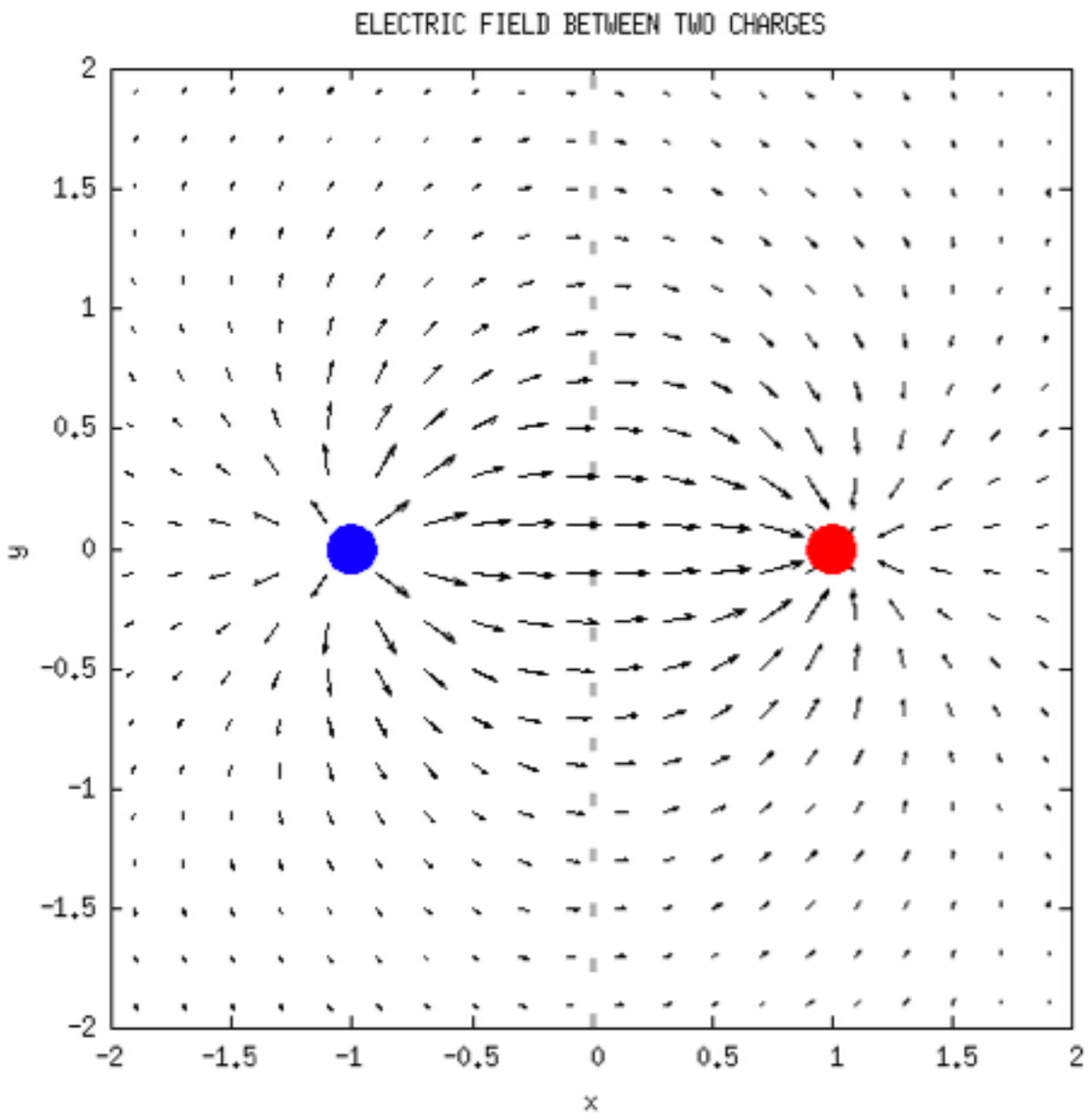
<object-type> is either **rectangle**, **ellipse**, **circle**, or **polygon**. Each object type has its own set of characteristic properties.

Vector plot

- *plot 'vec.dat' with vectors*
- needs four columns: x, y, deltax, deltay
- also works in 3d (six columns)



ROUND 2: FIGHT!

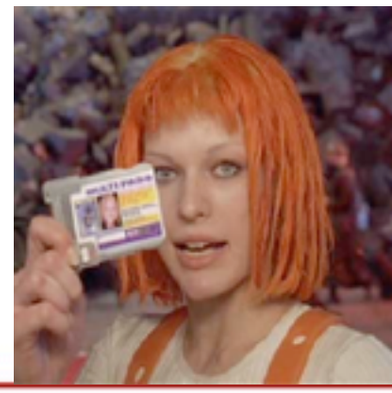


Round 3

Multiplot

- *set multiplot layout <rows>,<cols>*
  [more options available… prev/next]

- other options for more precise placing:
  set size … / set origin … (in “screen” coordinates)
  set {b,l,r,t}margin at screen …



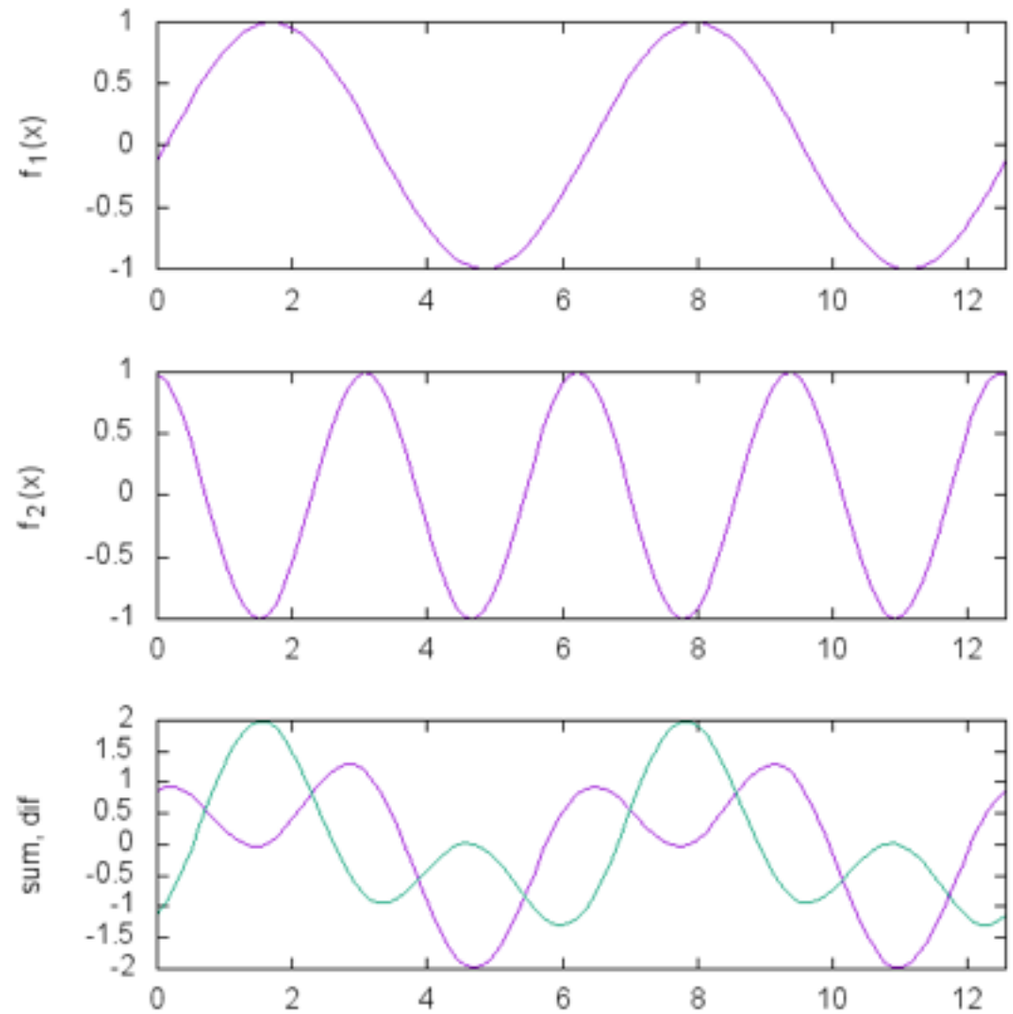
Animated gif

- *set terminal gif animate*
- a *plot* / *splot* creates a new frame
- ... or when we call *unset multiplot*

ROUND 3: FIGHT!

- f1(x) = sin(x - φ)
- f2(x) = cos(2x + φ)
- f1(x)+f2(x); f1(x)-f2(x)

  animate: φ from 0 to 2π



Round 4

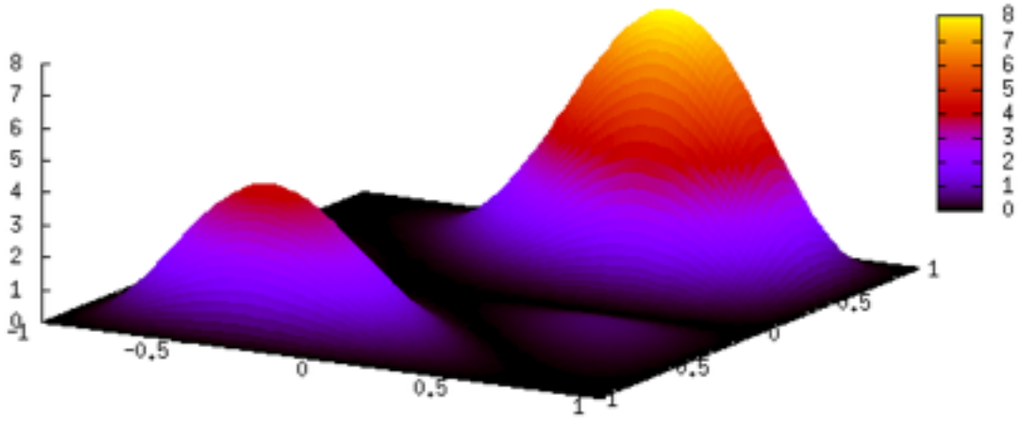
Colour plots

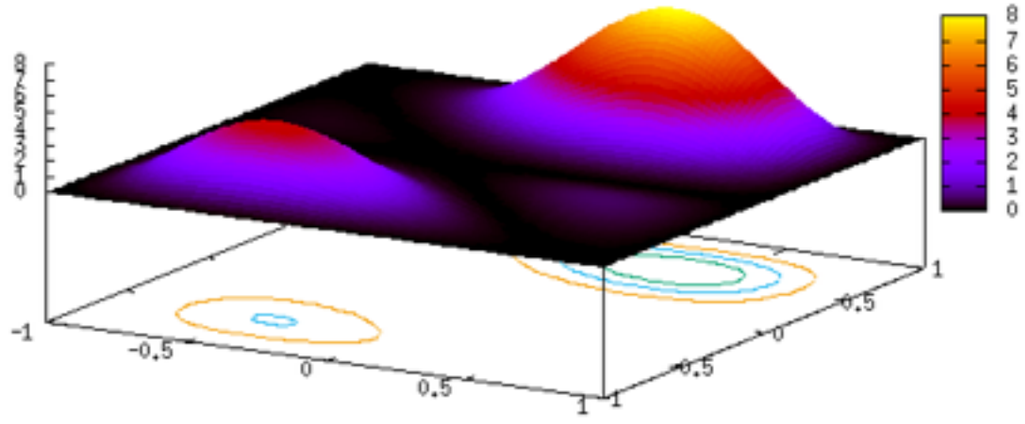
- *plot '2d.dat' w pm3d*
- *set pm3d map*
- *set/unset colorbox*
- *set palette …*
  see https://github.com/Gnuplotting/gnuplot-palettes
- hide parts of the plot that are obscured
  set hidden3d

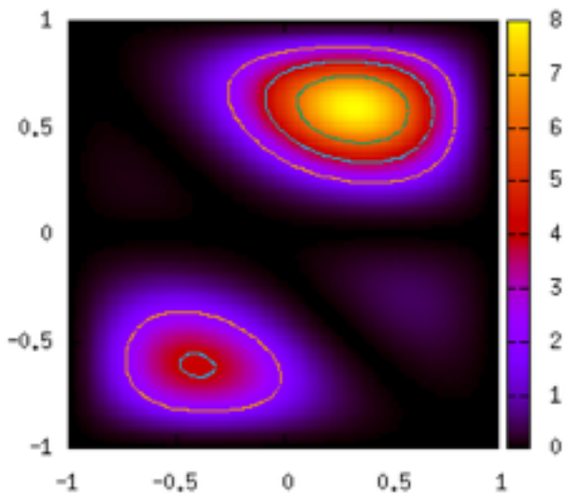
Contour plots

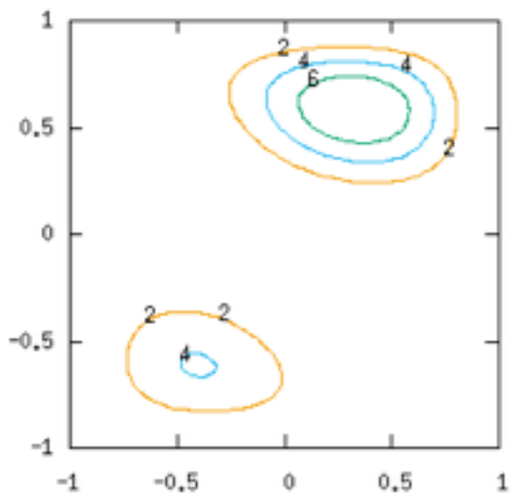
- turn on the contours
  set/unset contour (you may want to *unset surface* too)
- where are the contours drawn
  set contour base/surface/both
- at which z-values do we want the contour
  set cntrparam levels auto/discrete/incremental
- interpolations for the curves
  set cntrparam linear/cubicspline/…
- label the curves with their values
  set cntrlabel …
  splot 'data.dat' w lines, 'data.dat' w labels

ROUND 4: FIGHT!







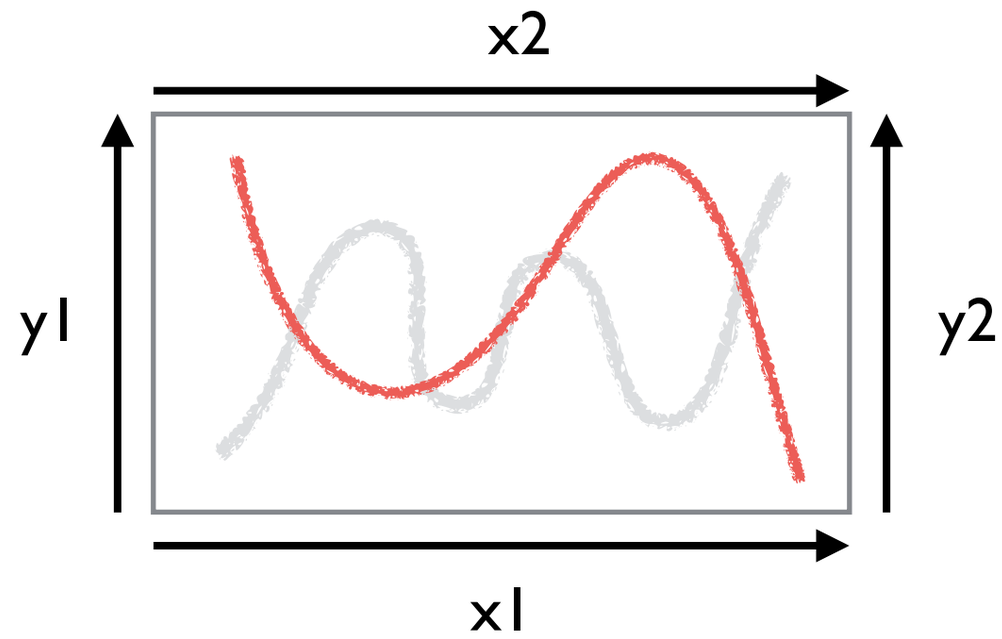


Bonus round!

```
set terminal epslatex   {default}
set terminal epslatex   {standalone | input}
                        {oldstyle | newstyle}
                        {level1 | leveldefault | level3}
                        {color | colour | monochrome}
                        {background <rgbcolor> | nobackground}
                        {dashlength | dl <DL>}
                        {linewidth | lw <LW>}
                        {rounded | butt}
                        {clip | noclip}
                        {palfuncparam <samples>{,<maxdeviation>}}
                        {size <XX>{unit},<YY>{unit}}
                        {header <header> | noheader}
                        {blacktext | colortext | colourtext}
                        {{font} "fontname{,fontsize}" {<fontsize>}}
                        {fontscale <scale>}
```

Secondary axes

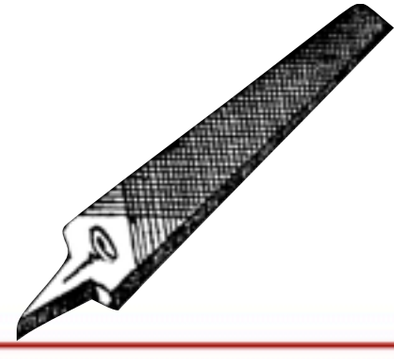
- gnuplot has two sets of axes



- *plot 'points.dat' axes x1y2*
- *set y2tics … / set y2range*
- *set ytics nomirror*

Special file names

- plot ‘file.dat’ u 1:2,“ u 1:3
  reuse the last filename
- plot ‘+’ …
  creates a column of xrange and samples
- plot ‘++’ …
  creates two columns of [xrange, samples] and [yrange,isosamples]
  fix the ranges first!
- plot ‘-’
  write below the data, end with e
  careful with replot/refresh/volatile



THE END

Other stuff?

SKILLPILLS

- Plots with labels
  Remember quotes for things with spaces
- Persist
- Binary data
- parametric
- set clip
- enhanced text / noenhanced
- plot … variables
  for line types, angles….