C2110 UNIX and programming

9. lekce

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

National Centre for Biomolecular Research, Faculty of Science
Masaryk University, Kotlářská 2, CZ-61137 Brno

european social fund in the czech republic EUROPEAN UNION MINISTRY OF EDUCATION, YOUTH AND SPORTS OP Education for Competitivness

INVESTMENTS IN EDUCATION DEVELOPMENT

CZ.1.07/2.2.00/15.0233

Contents

- **Bash**
  - **Cycle using while**
- **Gnuplot**
  - **Language overview, command plot, terminals, command splot**

Bash

- **Cycle using while**

Cycle using while

Cycle (loop) is control structure, that repeatedly processes set of commands. Repeats are done until condition is fulfilled.

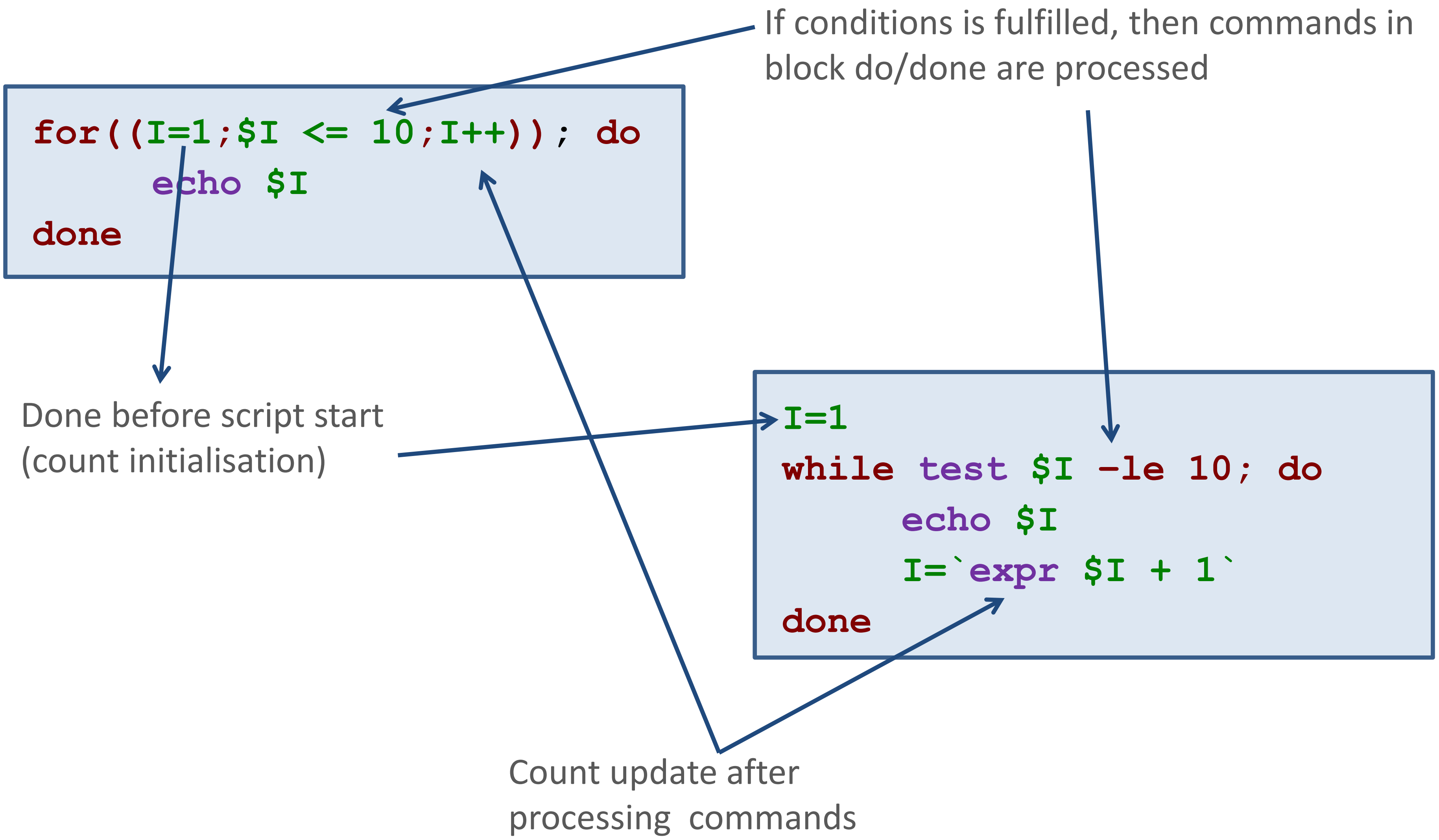
Cycle repeats **until** command1 return value is 0.

```
while command1
      do
            command2
            ...
      done
```

Compact form:

```
while command1; do
      command2
      ...
done
```

Cycle using for vs. while cycle



If conditions is fulfilled, then commands in block do/done are processed

```
for((I=1;$I <= 10;I++)); do
     echo $I
done
```

Done before script start (count initialisation)

```
I=1
while test $I -le 10; do
     echo $I
     I=`expr $I + 1`
done
```

Count update after processing commands

Redirection and pipes

Reading file per lines:

```
cat file.txt | while read A; do
        command2
        ...
done
```

roura

```
while read A; do
        command2
        ...
done < file.txt
```

redirection

Redirection to file:

```
for((I=1;I <= 10;I++)); do
        echo $I
done > file.txt
```

Output of all commands is redirected to **file.txt**.

Home work I

Explain different behavior of folloving scripts file.txt obsahuje pět řádků.

```
#!/bin/bash
I=0
cat data.txt | while read A; do
     I=$(($I+1))
done
echo $I
```

Print number 0

```
#!/bin/bash
I=0
while read A; do
     I=$(($I+1))
done < data.txt
echo $I
```

Print number 5

Home work II

File rst.out (wolf.ncbr.muni.cz:/home/kulhanek/Data/rst.out) contains results of molecular dynamics simulation. Task is to extract dependence of temperature on simulation time from file.

Time

Temperature

```
..............
NSTEP =      500   TIME(PS) =       0.500  TEMP(K) =   288.02  PRESS =     0.0
Etot   =       942.6248  EKtot   =       151.0990  EPtot      =       791.5258
BOND   =        51.3204  ANGLE   =       292.3619  DIHED      =       176.5980
1-4 NB =        17.7099  1-4 EEL =       981.4071  VDWAALS    =       -68.3301
EELEC  =      -494.7423  EGB     =      -164.7991  RESTRAINT  =         0.1822
EAMBER (non-restraint)  =       791.3436
..............
```

WARNING: Script nust not contain commands grep, awk and their variants. Use command read and while.

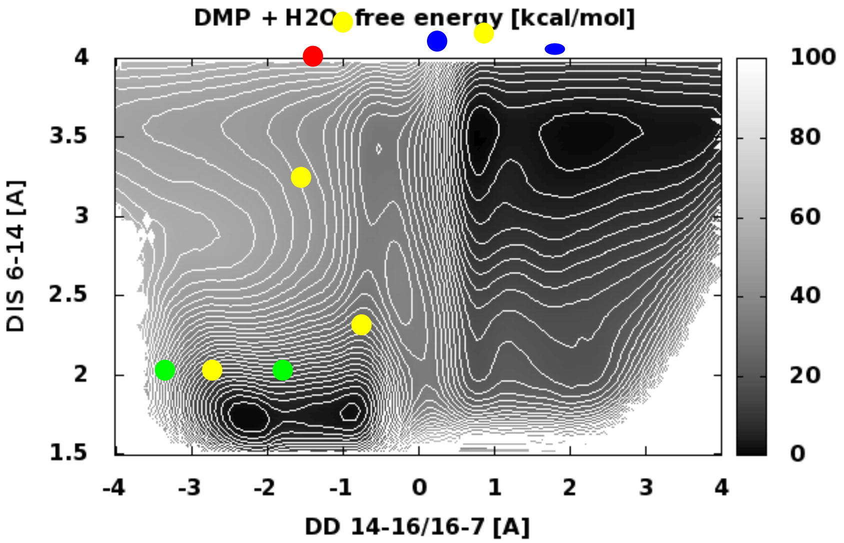
Gnuplot

- (Non)Interactive run
- Command plot
- Terminals
- Showcases

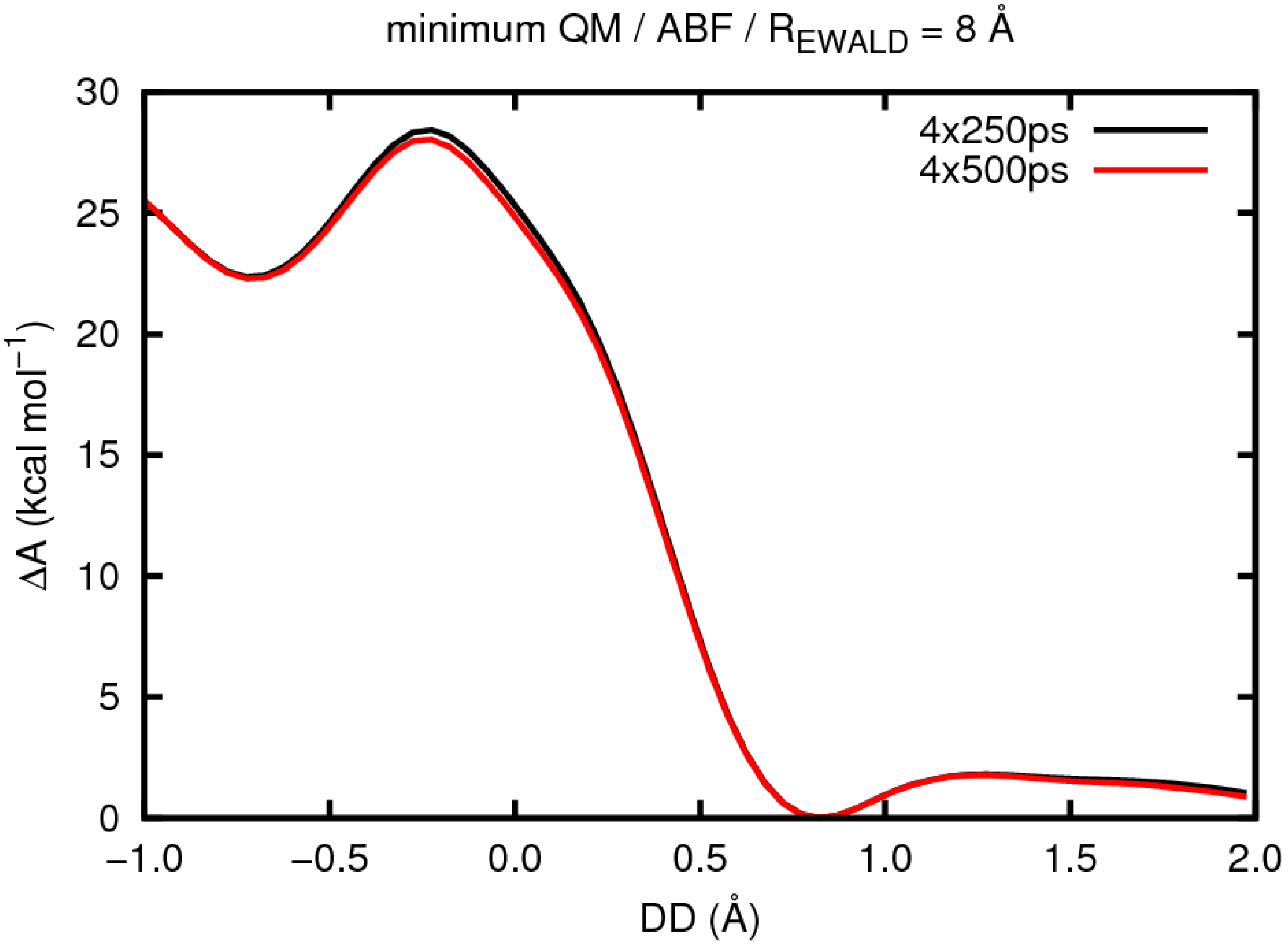
http://www.gnuplot.info/

(documentation, tutorials, source codes)

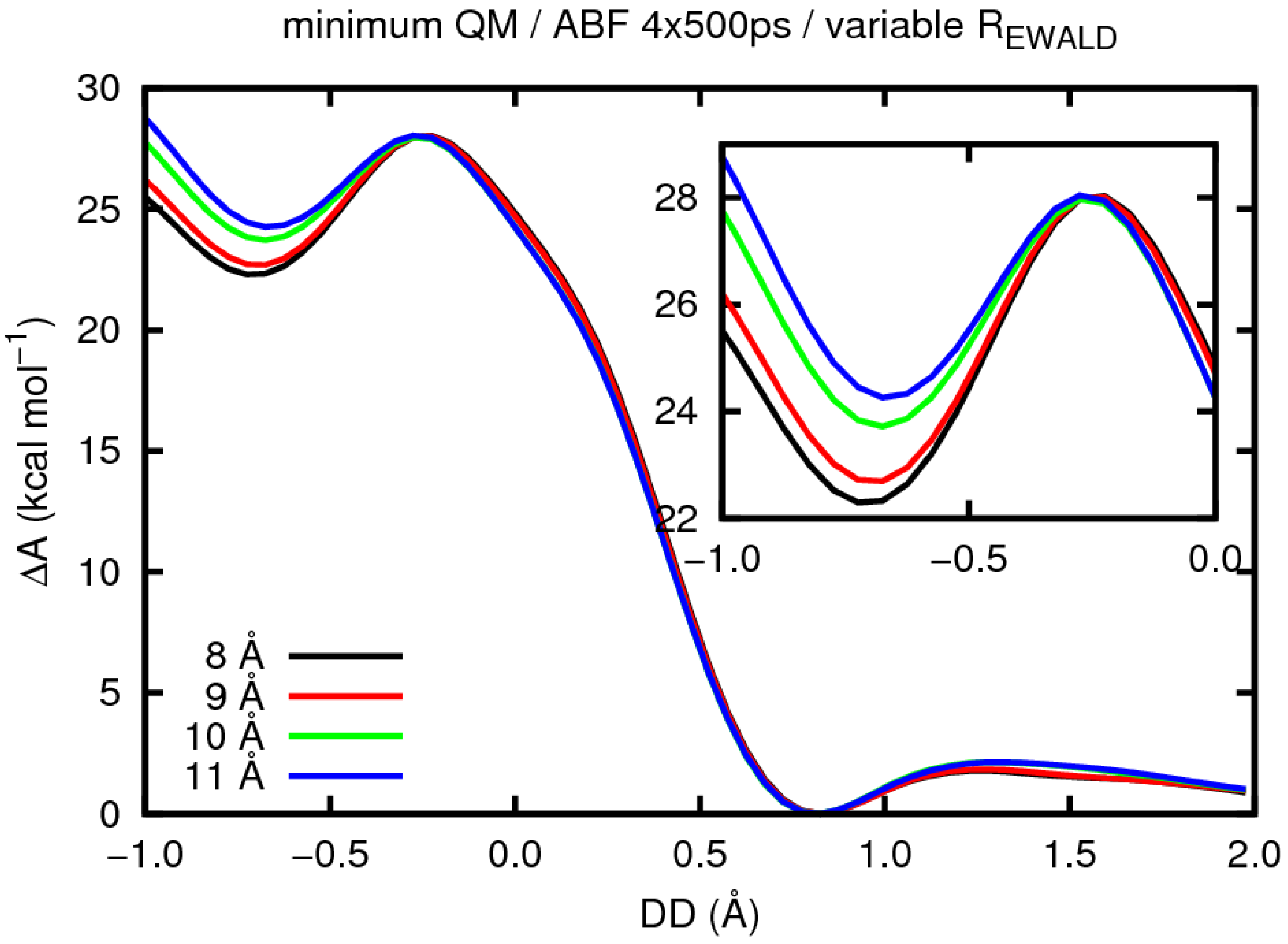
PM3



Showcase


minimum QM / ABF / R_{EWALD} = 8 Å
ΔA (kcal mol^{-1})
0
5
10
15
20
25
30
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
DD (Å)
4x250ps
4x500ps

Showcase



minimum QM / ABF 4x500ps / variable R_{EWALD}
ΔA (kcal mol^{-1})
30
25
20
15
10
5
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
DD (Å)
28
26
24
22
−1.0
−0.5
0.0
8 Å
9 Å
10 Å
11 Å

Interactive run

Gnuplot is dedicated to draw 2D and 3D graphs, through interactive and scripting mode as well.

Shell BASH command line

Interactive mode

```
[kulhanek@wolf ~]$ gnuplot

        G N U P L O T
        Version 4.4 patchlevel 3
        last modified March 2011
        System: Linux 3.2.0-31-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> 
```

gnuplot command line

Noninteractive run

1) Un-direct running

We run interpreter and as its argument we put script name.

```
$ gnuplot my_gnuplot_script
```

Scripts **does not need** permission x (executable).

2) Direct running

We run directly script (shell automatically start interpreter).

```
$ chmod u+x my_gnuplot_script
$./my_gnuplot_skript
```

Scripts **needs** permission **x** (executable) and **interpreter** (part of script).

```
#!/usr/bin/gnuplot
plot sin(x)
pause -1
```

Command - plot

```
> plot function/file [plot_setup] [, fce/file ...]
```

Shows XY graph of function or data from file.

Examples:

lines, points, linespoints, dots

Line color

```
> plot sin(x)
> plot cos(5.7*x+3.4) with points linecolor rgb "red" \
                       linewidth 2 title "cos"
```

Line thickness

legend

Data file name

```
> plot "input.txt" using 1:2 with lines
```

Second column are y-axis values.

First column are x-axis values.

```
> plot sin(x), cos(x)
```

Plots functions sin and cos to one graph.

Exercise

1. Plot function graph of $y=x^2$
2. Plot function from task 1 again, but with blue line.
3. Plot graph of temperature dependency on time from data file **/home/kulhanek/Data/temp.txt** Time is in first column, temperature is in second column.
4. Plot graph of function sin(x) using red line and function cos(x) using orange line with points.

Do all tasks in interactive mode.

Other commands

```
> set title "title"                            # Graph header

> set xrange[min_value:max_value]              # sets range of x axis

➢ set xlabel "title"                           # sets title of x axis

> set yrange[min_value:max_value]              # sets range of y axis

> set ylabel "title"                           # sets title of y axis

> set nokey                                    # disables key for data in plot

> pause -1                                     # wait for key press
```

Exercise

1. Write script, that plots function y=x^2 in range 0-10 for x axis. Run script un-directly using gnuplot interpreter.

2. Write script, that plots dependence of temperature on time from data in file **/home/kulhanek/Data/temp.txt** . Add axis labels including units. Time is in picoseconds, temperature in kelvin.

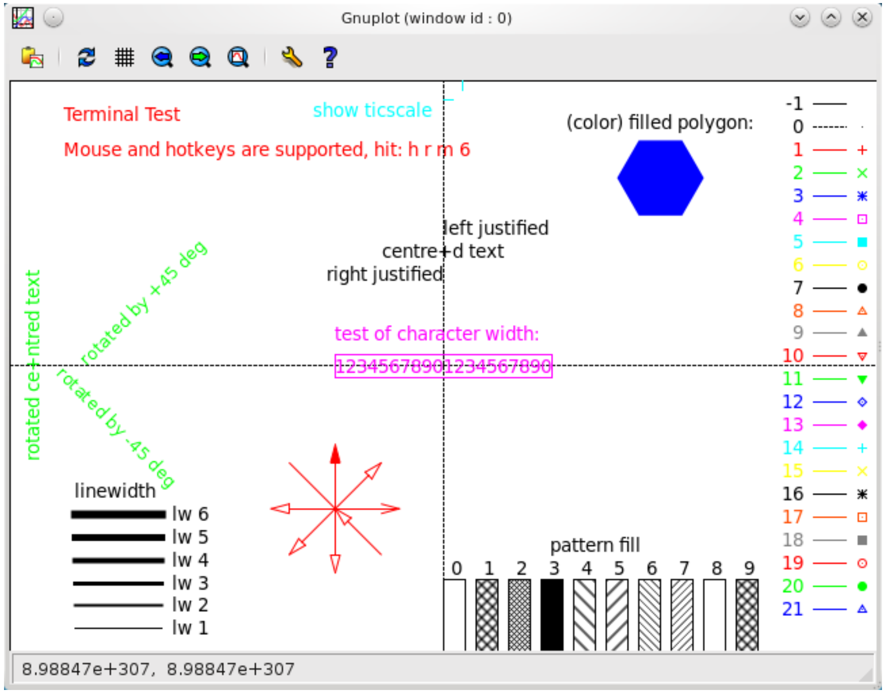
Terminals

Terminal determines output form.

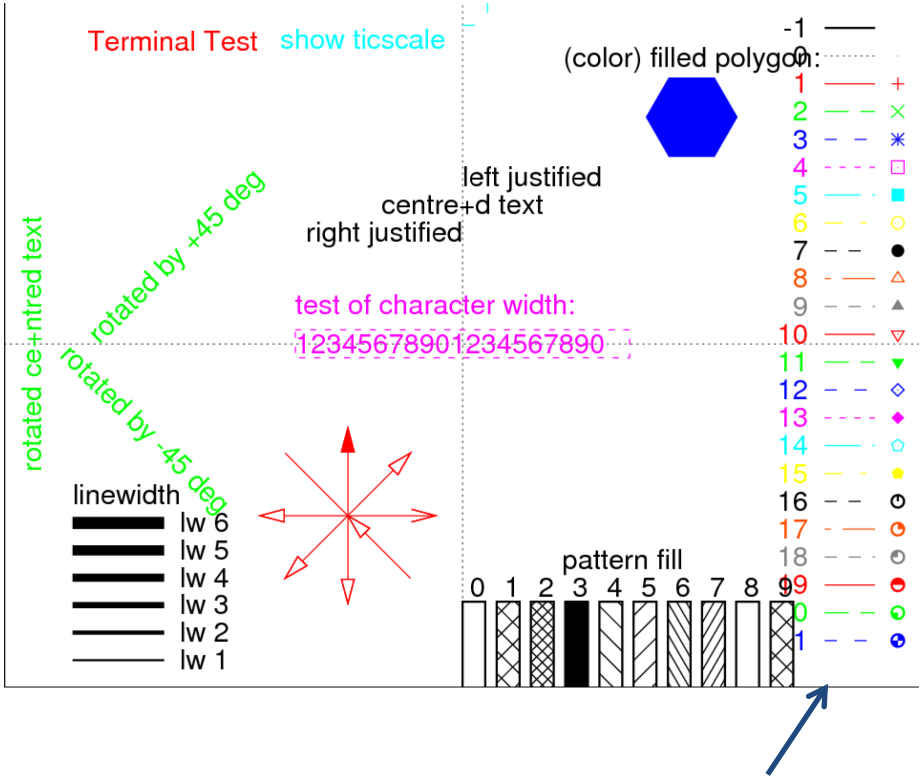
```
> set term x11                        # output is shown in window

> set term wxt                        # output is shown in advanced window

> set term png size 800,600
                                      # output is plot as a picture in png format

> set output "output.png"             # output will be saved to file output.png

> test                                # prints page with terminal features, terminals does
                                      have different abilities
```

Sample terminal outputs

wxt



postcript/eps



Supports dotted lines

Exercise

1. What possibilities offer terminals x11 and wxt. Work in interactive mode and use command **test**.
2. Write script, that plots graph of function $y=5.x^3 + 6.\ x^2 - 7$ in range -10 to 5 on x axis. Run script directly with interpreter in script header.
3. Change previous script to plot graph to format png. Picture size will be 640x480. Show picture using command **display**.
4. View result of command **test** for terminal png and postscript.
5. What terminals gnuplot supports (set terminal with no argument)?

Command - splot

To plot function of two variables, command splot may be used.

```
> splot functionkce/file [plot_setup] [, fce/file ...]
```

Shows **XYZ** graph of function or data from file.

View point is set by command **set view a,b**, where **a** and **b** are view angles. View from top is set by **set view map**

Sampling of density for x and y axis is set by command
set isosamples a,b, where **a** and **b** gives number of samples in given direction.

To highlight surface by its function value **pm3d** command may be used, for example:

```
> splot x*x+y*y with pm3d
```

Exercise

1. Plot function x^2+y^2
2. Set top view (**set view**)
3. Unset current view (**unset view**)
4. Raise density of points for function plot (**set isosamples**). Use values 10,20; 20,10 and 20,20
5. Use view **pm3d**
6. Set top view (**set view**)

Do all tasks in interactive mode.