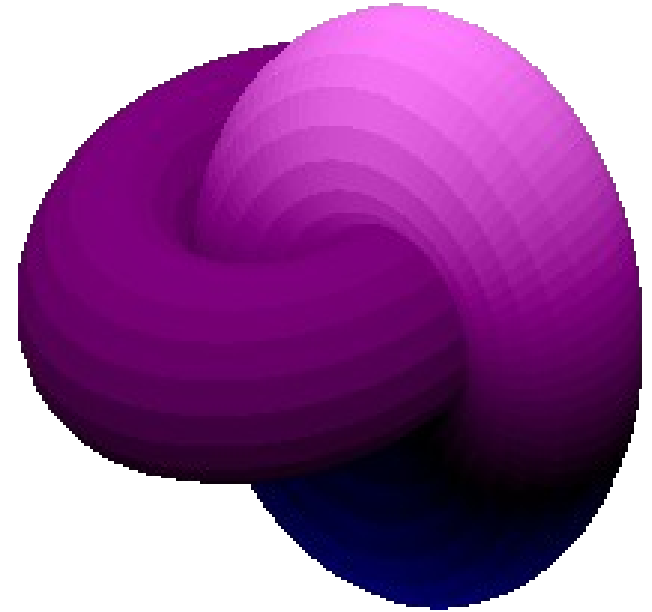
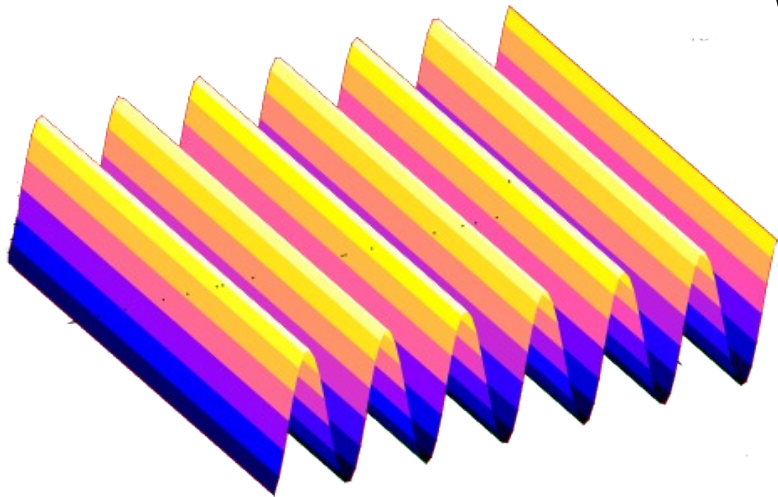


GNUPLOT

Wprowadzenie

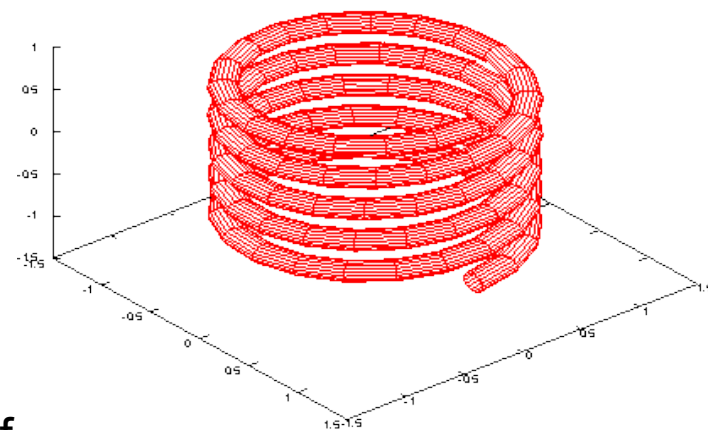


dr inż. Marzena Sala-Tefelska
marzena.sala-tefelska@pw.edu.pl
Wydział Fizyki,
Politechnika Warszawska
2023

Program **Gnuplot**

Służy do rysowania wykresów zdefiniowanych funkcji w **2D** lub **3D**, a także wczytanych danych liczbowych z eksperymentu, ponadto:

- aproksymuje dane poprzez zadaną przez nas funkcję,
- wyznacza współczynniki aproksymowanej funkcji,
- pozwala nanosić niepewności pomiarowe
- umożliwia odpowiednie formatowanie wykresów
- zapisuje wykresy do plików: .jpg, .bmp, .png, .eps, .pdf
- współpracuje z LaTeX'em



Praca może odbywać się w dwóch trybach interaktywnych:

- polecenia z **terminala**
- tryb wsadowy: **skrypt**

Ustawienie katalogu dla gnuplota

Komenda:

pwd (print working directory) – pozwala zobaczyć domyślny katalog gnuplota, w którym zapisywane będą wszystkie pliki wygenerowane za pomocą programu

Przykład:

```
gnuplot> pwd  
C:\Users\Admin\Documents
```

Jeżeli chcemy zmienić docelowy katalog należy zastosować komendę:
cd (change directory) czyli wpisujemy: cd 'ścieżka do katalogu'

Uwaga: ścieżka do katalogu musi być zapisana w cudzysłowach podwójnych "" lub pojedynczych "'

Przykład:

```
cd 'D:\dane_gnuplot'
```

Komendy Gnuplot: http://gnuplot.info/docs_5.5/Commands.html

Ustawienie pracy w terminalu, zapis wyniku

```
GNUPLOT
Version 4.6 patchlevel 6    last modified September
2014
Build System: Linux x86_64

Copyright (C) 1986-1993, 1998, 2004, 2007-2014
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h'
)

Terminal type set to 'x11'
gnuplot> █
```

Do podglądu ustawień służą polecenia:

show terminal - pokazuje typ terminalu (typ wyjścia)

show output - pokazuje ustawienie wyjściowe pliku (plik wyjściowy)

set terminal - pokaże nam dostępne terminale

```
gnuplot> show terminal
      terminal type is x11
gnuplot> show output
      output is sent to STDOUT
```

```
gnuplot> set terminal
```

```
Available terminal types:
```

```
  caca  Colour ascii art using libcaca, the Colour AsCii Art library
  cairolatex  LaTeX picture environment using graphicx package and Cairo backend
  canvas  HTML Canvas object
  cgm  Computer Graphics Metafile
  context  ConTeXt with MetaFun (for PDF documents)
  corel  EPS format for CorelDRAW
  dumb  ascii art for anything that prints text
  dxf  dxf-file for AutoCad (default size 120x80)
  eepic  EEPIC -- extended LaTeX picture environment
  emf  Enhanced Metafile format
  emtex  LaTeX picture environment with emTeX specials
  epscairo  eps terminal based on cairo
  epslatex  LaTeX picture environment using graphicx package
  fig  FIG graphics language for XFIG graphics editor
  gif  GIF images using libgd and TrueType fonts
  hpgl  HP7475 and relatives [number of pens] [eject]
  jpeg  JPEG images using libgd and TrueType fonts
  latex  LaTeX picture environment
  lua  Lua generic terminal driver
  mf  Metafont plotting standard
  mp  MetaPost plotting standard
  pcl5  HP Designjet 750C, HP Laserjet III/IV, etc. (many options)
  pdfcairo  pdf terminal based on cairo
  png  PNG images using libgd and TrueType fonts
  pngcairo  png terminal based on cairo
  postscript  PostScript graphics, including EPSF embedded files (*.eps)
  pslatex  LaTeX picture environment with PostScript \specials
  pstex  plain TeX with PostScript \specials
  pstricks  LaTeX picture environment with PSTricks macros
  qms  QMS/QUIC Laser printer (also Talaris 1200 and others)
  qt  Qt terminal
  svg  W3C Scalable Vector Graphics
  texdraw  LaTeX texdraw environment
  tgif  TGIF X11 [mode] [x,y] [dashed] ["font" [fontsize]]
  tikz  TeX TikZ graphics macros via the lua script driver
  tkcanvas  Tk canvas widget
  tpic  TPIC -- LaTeX picture environment with tpic \specials
  unknown  Unknown terminal type - not a plotting device
  windows  Microsoft Windows
  wxt  wxWidgets cross-platform windowed terminal
```

WINDOWS

```
Available terminal types:
  cairolatex LaTeX picture environment using graphicx package and Cairo ba
ckend
  canvas HTML Canvas object
  cgm Computer Graphics Metafile
  context ConTeXt with MetaFun (for PDF documents)
  corel EPS format for CorelDRAW
  dumb ascii art for anything that prints text
  dxf dxf-file for AutoCad (default size 120x80)
  eepic EEPIC -- extended LaTeX picture environment
  emf Enhanced Metafile format
  emtex LaTeX picture environment with emTeX specials
  epscairo eps terminal based on cairo
  epslatex LaTeX picture environment using graphicx package
  fig FIG graphics language for XFIG graphics editor
  gif GIF images using libgd and TrueType fonts
  gpic GPIC -- Produce graphs in groff using the gpic preprocessor
  hp2623A HP2623A and maybe others
  hp2648 HP2648 and HP2647
  hpgl HP7475 and relatives [number of pens] [eject]
  imagen Imagen laser printer
  jpeg JPEG images using libgd and TrueType fonts
  latex LaTeX picture environment
```

Press return for more:

```
  lua Lua generic terminal driver
  mf Metafont plotting standard
  mif Frame maker MIF 3.00 format
  mp MetaPost plotting standard
  pcl5 HP Designjet 750C, HP Laserjet III/IV, etc. (many options)
  pdfcairo pdf terminal based on cairo
  png PNG images using libgd and TrueType fonts
  pngcairo png terminal based on cairo
  postscript PostScript graphics, including EPSF embedded files (*.eps)
  pslatex LaTeX picture environment with PostScript \specials
  pstex plain TeX with PostScript \specials
  pstricks LaTeX picture environment with PSTricks macros
  qms QMS/QUIC Laser printer (also Talaris 1200 and others)
  regis REGIS graphics language
  svg W3C Scalable Vector Graphics driver
  tek40xx Tektronix 4010 and others; most TEK emulators
  tek410x Tektronix 4106, 4107, 4109 and 420X terminals
  texdraw LaTeX texdraw environment
  tgif TGIF X11 [mode] [x,y] [dashed] ["font" [fontsize]]
  tikz TeX TikZ graphics macros via the lua script driver
  tkcanvas Tk/Tcl canvas widget [perltk] [interactive]
  tpic TPIC -- LaTeX picture environment with tpic \specials
```

Press return for more:

```
  unknown Unknown terminal type - not a plotting device
  vttek VT-like tek40xx terminal emulator
  wxt wxWidgets cross-platform windowed terminal
  x11 X11 Window System
  xlib X11 Window System (gnulib_x11 dump)
  xterm Xterm Tektronix 4014 Mode
```

LINUX

Ustawienie pracy w terminalu, zapis wyniku

Do zapisu ustawień służy polecenie:

set terminal *typ*

typ przyjmuje wartości z listy pokazanej na poprzednim slajdzie

set output - ustawia wyjście danych (tzn. czy ekran czy plik)

set output "nazwa_pliku"

set output - jeśli pominiemy nazwę, wyjście zostanie przekierowane na STDOUT (ekran)

Przykłady

```
set terminal png size 500,400 font 'Arial,12'  
set output 'plik1.png'
```

```
set terminal png size 350,262 enhanced font 'Verdana,10'  
set output 'plik1.png'
```

```
set terminal svg size 350,262 fname 'Verdana' fsize 10  
set output 'plik1.svg'
```

TimesNewRoman,14
Verdana,10



Przykłady

```
set terminal postscript eps enhanced color font 'Arial,10'  
set output 'plik1.eps'
```

Tworzenie wykresu do LaTeX'a:

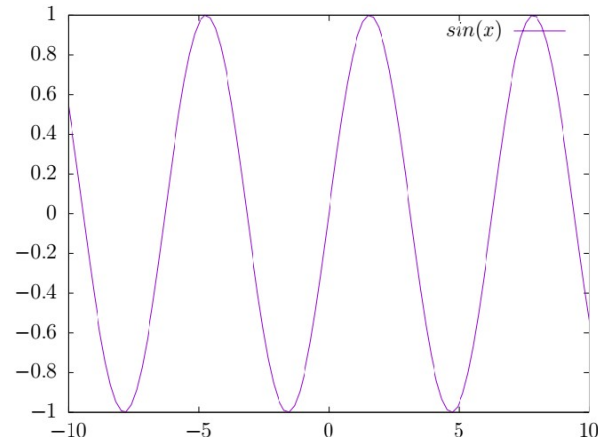
```
set terminal epslatex  
set output "wykres.tex"  
plot cos(2*x)  
unset output
```

Wygenerowane zostaną dwa pliki: wykres.tex oraz wykres.eps
Należy je skopiować do katalogu z plikami .tex.

W dokumencie LaTeX'a w sekcji obrazka należy dopisać: `\input{wykres.tex}`

Przykład:

```
\begin{figure} [h]  
\begin{center}  
\input{wykres.tex}  
\end{center}  
\caption{Wykres sinusa}  
\end{figure}
```



Rysunek 1: Wykres sinusa

Przykłady

```
set terminal postscript eps enhanced color font 'Arial,10'  
set output 'plik1.eps'
```

set table "dane" – wynikiem jest tablica zawierająca współrzędne punktów wykresu, zapisana do katalogu z plikami z gnuplota

unset table - wyłącza zapis do tablicy

```
set terminal wxt lub x11 lub qt  
set output
```

- ustawia wyświetlanie wykresu w oddzielnym oknie
- ustawia wyjście jako STDOUT

Funkcja plot, splot - rysowanie funkcji

Polecenie:

plot - wykres 2D,

splot – wykres 3D,

Składnia: **plot** [function] [options]

plot {<ranges>} {<function> | {"<datafile>" {<datafile-modifiers>}} {axes <axes>}
{<title-spec>} {with <style>} {, {definitions,} <function> ...}

Funkcja może być podana jawnie poprzez swój przepis np.:

plot x

domyślny zakres dla wartości x to od -10 do +10

plot sin(x)

splot x*y

albo poprzez zadeklarowaną wcześniej funkcję: $f(x)=\sin(x)/x$

plot [-15:15] f(x)

(funkcja sinc - sinus cardinalis, pierwsza sferyczna funkcja Bessela)

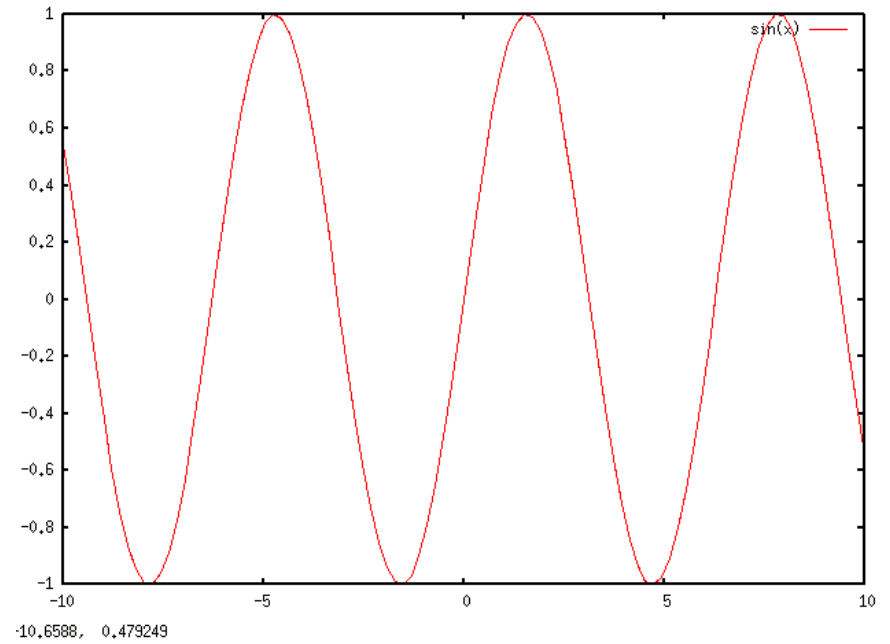
Jeśli posiadamy dodatkowo plik z danymi empirycznymi, wówczas:

plot 'plik_z_danymi.txt'

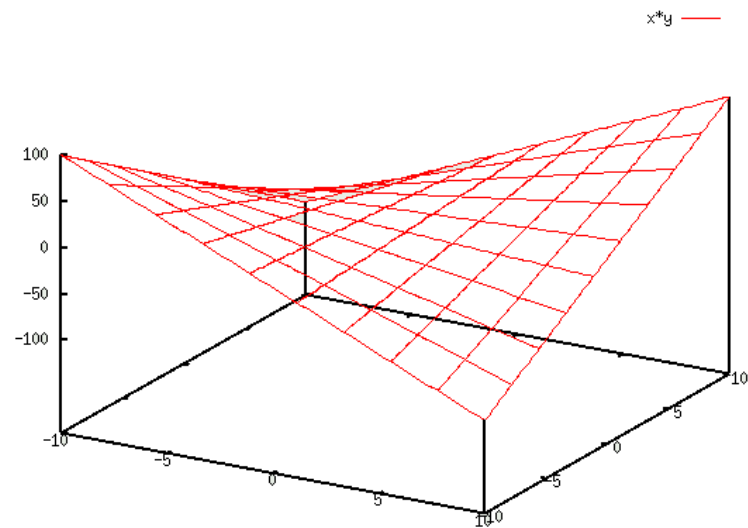
<functions> - kolejne funkcje do narysowania - oddzielane są przecinkami, nie można nadawać kolejnych wartości <range> (można to zrobić tylko raz, po komendzie plot)- pozostałe parametry można modyfikować i nadawać każdej z rysowanych funkcji

with <style>; - parametry określające styl rysowanych wykresów.

```
gnuplot> plot sin(x)
```



```
gnuplot> splot x*y
```



view: 60.0000, 30.0000 scale: 1.00000, 1.00000

Komenda: set table

Pozwala na dostęp do danych tworzących wykres w postaci tekstu.

Aby wygenerować dane w postaci tekstu należy:

```
set table 'nazwa_pliku'
```

```
plot [funkcja]
```

np.

```
f(x)=sin(x)/x
```

```
set table 'sinc.txt'
```

```
plot f(x)
```

Aby przywrócić generowanie wykresów należy użyć komendy:

```
unset table
```

Podstawowe operatory i funkcje w Gnuplot

+, -, *, /, ** (potęgowanie)

x^{**2}

$-x^{**2}$

exp(x)

log10(x) – logarytm przy podstawie 10

log(x) - logarytm naturalny

sqrt(x)

abs(-x**2)

abs(-3)

rand(0) - generator liczb pseudolosowych z zakresu [0:1], np. plot 3*rand(0)

sin(x)

cos(x)

tan(x)

asin(x)

acos(x)

atan(x)

sinh(x)

cosh(x)

tanh(x)

asinh(x)

acosh(x)

atanh(x)

} Funkcje trygonometryczne

} Odwrotne funkcje trygonometryczne

} Funkcje hiperboliczne

} Odwrotne funkcje hiperboliczne

plot [0:pi] tan(x)

plot [-pi:pi] sin(x)

Podstawowe operatory i funkcje w Gnuplot

Uwaga! operacje na zmiennych zdefiniowanych jako całkowite będą dawać inne rezultaty, niż dla zmiennych rzeczywistych np.

```
a = 1  
b = 2  
y = a/b = 0
```

```
a= 1.0  
b=2.0  
y = a/b = 0.5
```

W gnuplocie aby wyświetlić wynik należy użyć komendy print:

```
print y
```

Przydatne opcje/parametry do funkcji plot

title "tytuł" - nadaje rysowanej krzywej nazwę "tytuł" w legendzie wykresu.

notitle - nie nadaje krzywej nazwy (nie występuje w legendzie)

```
plot sin(x) title "Funkcja sinus"
```

przykłady:

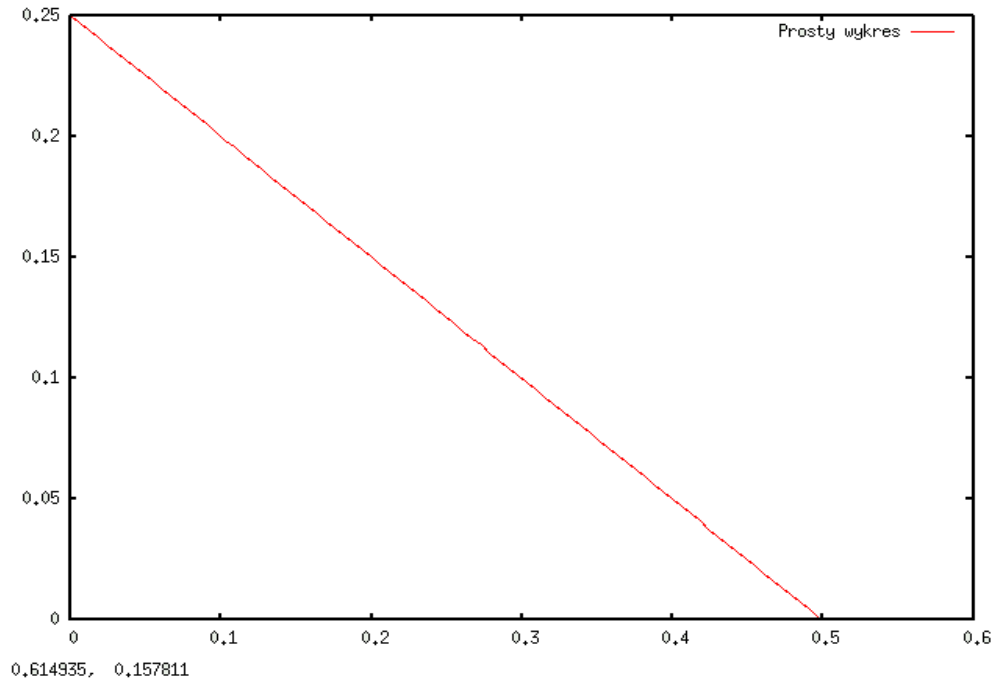
```
plot 2*x+3 title "Prosta o rownaniu y=2x+3"
```

```
plot 0.5*x+5 notitle
```

```
plot [0 : 0.6] [0: ] -0.5*x+0.25 title "Prosty wykres"
```

Zakres X

Zakres Y



Przydatne opcje/parametry do funkcji plot

Parametry stylu podajemy po słowie **with** (w skrócie „w”)

linespoints (w skrócie lp) - łączy punkty pomiarowe prostą

pointtype (w skrócie pt) – typ punktu

points (w skrócie p) – same punkty, często stosujemy z parametrem pt

plot sin(x) title "Funkcja sinus" with linespoints pointtype 3

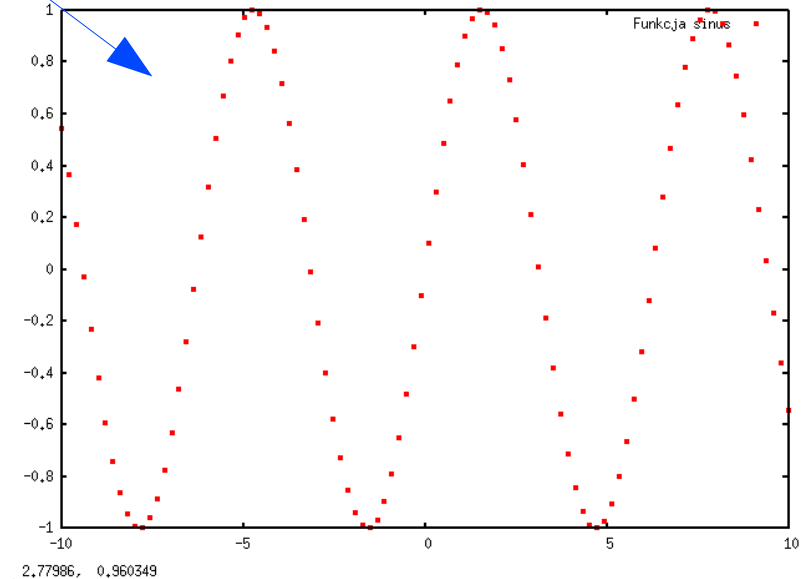
plot sin(x) title "Funkcja sinus" w lp pt 3

plot sin(x) title "Funkcja sinus" with points pointtype 5

plot sin(x) title "Funkcja sinus" w p pt 5

pointtype:

1 - +	11 - ▼
2 - ×	12 - ◇
3 - *	13 - ◆
4 - □	
5 - ■	
6 - ○	
7 - ●	
8 - △	
9 - ▲	
10 - ▽	



Przydatne opcje/parametry do funkcji plot

pointsize - wybór rozmiaru punktów (w skrócie jako 'ps').
Po tej opcji podajemy liczbowy rozmiar punktów

pointinterval - wybór odstępu między punktami. W skrócie można zapisywać jako pi.
Po tej opcji podajemy odstęp między sąsiednimi punktami

przykłady:

```
plot sin(x) w p pt 2 ps 5
```

```
plot sin(x) w p pt 2 ps 2
```

```
plot sin(x) w lp pt 4 ps 2 pi 2
```

```
plot sin(x) w lp pt 4 ps 2 pi 4
```

linespoints - rysowanie za pomocą linii z punktami,
(skrót lp)

dots - rysowanie za pomocą kropek (skrótowy zapis - d)

przykłady:

```
plot sin(x) with dots
```

```
plot sin(x) w d
```

lines - rysowanie za pomocą linii (l), opcje:

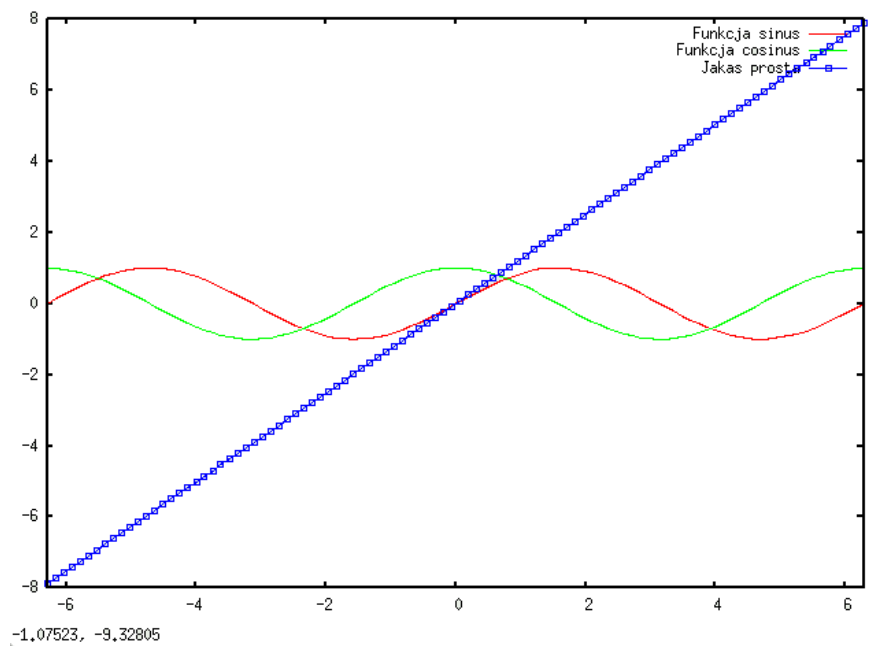
- **linetype** - kolor linii (lt), lub typ linii (lt) – np. gdy terminal nie obsługuje kolorów. Następnie należy podać liczbę całkowitą przypisaną w programie do danego koloru lub typu. Możemy też podać numer koloru w RGB.

plot sin(x) w | lt rgb "#9966CC"

- **linecolor** - wybór koloru linii (lc), a następnie podajemy numer koloru w RGB,
plot sin(x) w | lc rgb "#E61C66"

- **linewidth** - ustawienie szerokości linii (lw). Po tej opcji należy podać liczbę stanowiącą szerokość linii

plot sin(x) w | lc rgb "#33CC66" lw 5



Trzy funkcje na jednym wykresie:

```
plot [-2*pi: 2*pi] [] sin(x) title "Funkcja sinus" w l lt 1, cos(x) title "Funkcja cosinus" w l  
lt 2, 1.254*x title "Jakas prosta" w lp pt 4
```

Wczytywanie pliku z danymi, reprezentacja danych

Plik Halleffect i galton jest na stronie www.if.pw.edu.pl/~martef/Halleffect.txt
www.if.pw.edu.pl/~martef/galton.txt
www.if.pw.edu.pl/~martef/galton2.txt

```
plot 'Halleffect.txt'
```

```
plot 'galton.txt'
```

Zmiana kolumn tzn chcemy wyświetlić kolumnę 3 jako argumenty, zaś kolumnę drugą jako wartości - posługujemy się poleceniem:

```
plot 'galton.txt' using 3:2
```

lub

```
plot 'galton.txt' u 3:2
```

zmiana skali na osi x oraz linie z punktami

```
plot [5:15] 'galton.txt' w lp ps 2
```

impulses - rysowanie pionowych linii łączących punkt na osi X (argument) z odpowiadającą jemu wartością
przykłady:

```
plot sin(x) with impulses
```

```
plot tan(x) w i
```

```
plot 'galton.txt' w i
```

UWAGA: Plik z danymi musi być odpowiednio przygotowany!

Plik tekstowy ASCII

Dane w kolumnach oddzielone białymi znakami

Liczby rzeczywiste zapisane z kropką!

Niepewności pomiarowe:

yerrorbars - rysowanie słupków niepewności dla osi Y

przykłady:

`plot 'galton.txt' w yerrorbars`

xerrorbars - rysowanie słupków niepewności dla osi X

przykłady:

`plot 'galton.txt' w xerrorbars`

xyerrorbars - rysowanie słupków niepewności dla osi X oraz Y

przykłady:

www.if.pw.edu.pl/~martef/galton2.txt

`plot 'galton2.txt' w xyerrorbars`

xerrorbars (yerrorbars,xyerrorabrs) - pionowe słupki błędów (poziome, "krzyżowe")

boxerrorbars - prostokąty błędów

boxes - tworzy prostokąty od osi odciętych do punktu krzywej - przydatne przy histogramach

Niepewności pomiarowe:

`show bars`

pokazuje wielkość krzyżyka prezentującego błąd na wykresie

`set bars 2`

ustawia wielkość krzyżyka

Podpisy osi oraz tytuł wykresu

Podpis dolnej osi X ustalamy w następujący sposób:

```
set xlabel "Podpis dolnej osi X1"
```

Dla górnej osi X składnia wygląda tak:

```
set x2label "Podpis górnej osi X2"
```

Analogicznie dla obu osi Y:

```
set ylabel "Podpis lewej osi Y"  
set y2label "Podpis prawej osi Y"
```

Aby ustalić tytuł wykresu należy wydać polecenie w postaci:

```
set title "Tytuł"
```

Uwaga!:

reset – resetuje wszystkie dotychczasowe ustawienia

replot - przerysowanie wykresu z nowymi ustawieniami

```
set title "Hall effect" font 'Arial,14'  
set xlabel "Is [mA]" font 'Arial,14'  
set ylabel "Uh [mV]" font 'Arial,14'
```

```
plot 'Halleffect.txt' notitle w p pt 3 ps 3
```

Dopasowanie krzywej - fit

Dopasowanie zadanej przez nas krzywej do posiadanych punktów empirycznych.

Gnuplot wykorzystuje do tego celu metodę regresji liniowej (**najmniejszych kwadratów**) podając wartości żądanych parametrów krzywej razem z niepewnościami.

Składnia:

fit [function] 'Halleffect.txt' via var1,var2...varn

Gdzie:

[function] - funkcja którą chcemy dopasować

'Halleffect.txt' - plik z danymi

var1,var2...,varn - zmienne dopasowywanej funkcji

Przed wykonaniem fitowania musimy oczywiście zadeklarować funkcję, a w niektórych przypadkach (funkcje `exp()` i `log()`) nie obędzie się bez podania początkowych wartości zmiennych.

Dopasowanie krzywej - fit

fit [0:10] f(x) 'dane.txt' u 1:2:3 via a,b,c

↑
zakres

↑
funkcja

↑
dane

↗
format danych
(zalecane)

↑
obowiązkowa lista
dopasowywanych
parametrów

Przykład:

```
plot 'Halleffect.txt' using 1:2:3:4 with xyerrorbars
```

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

```
fit f(x) "Halleffect.txt" u 1:2 via a,b
```

```
plot 'Halleffect.txt' using 1:2:3:4 with xyerrorbars, f(x)
```


Dopasowanie krzywej – fit

```
a          = 21.4602
b          = 0.893446
```

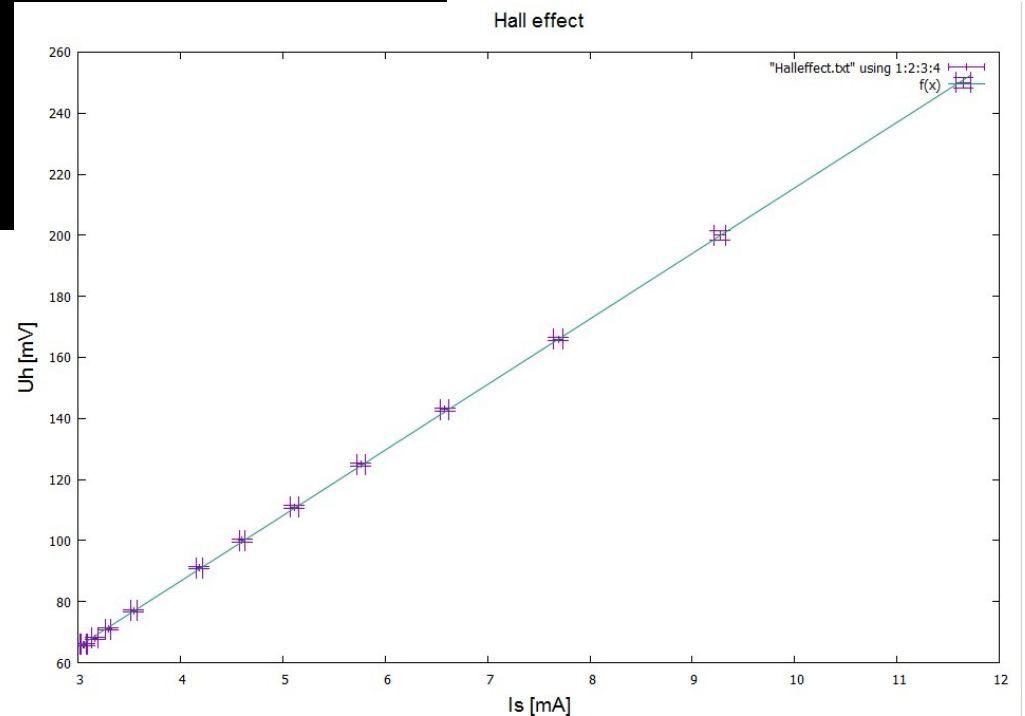
```
After 4 iterations the fit converged.
final sum of squares of residuals : 3.62196
rel. change during last iteration : -1.79981e-08
```

```
degrees of freedom (FIT_NDF)          : 11
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.57382
variance of residuals (reduced chisquare) = WSSR/ndf : 0.329269
```

```
Final set of parameters          Asymptotic Standard Error
=====                          =====
a          = 21.4602          +/- 0.06155      (0.2868%)
b          = 0.893446        +/- 0.3717       (41.6%)
```

```
correlation matrix of the fit parameters:
```

```
          a          b
a          1.000
b          -0.904   1.000
```



Dopasowanie Gaussa

```
stat 'wzrost_studentow.txt'
```

```
show variable
```

```
print STATS_mean, STATS_stddev
```

```
u=STATS_mean
```

```
sigma=STATS_stddev
```

```
f(x)=1/(sqrt(2*pi)*sigma)*exp(-(x-u)**2/(2*sigma**2))
```

```
plot 'wzrost_studentow_bins.txt' using 1:2:3 with yerrorbars
```

```
fit f(x) 'wzrost_studentow_bins.txt' u 1:2 via u, sigma
```

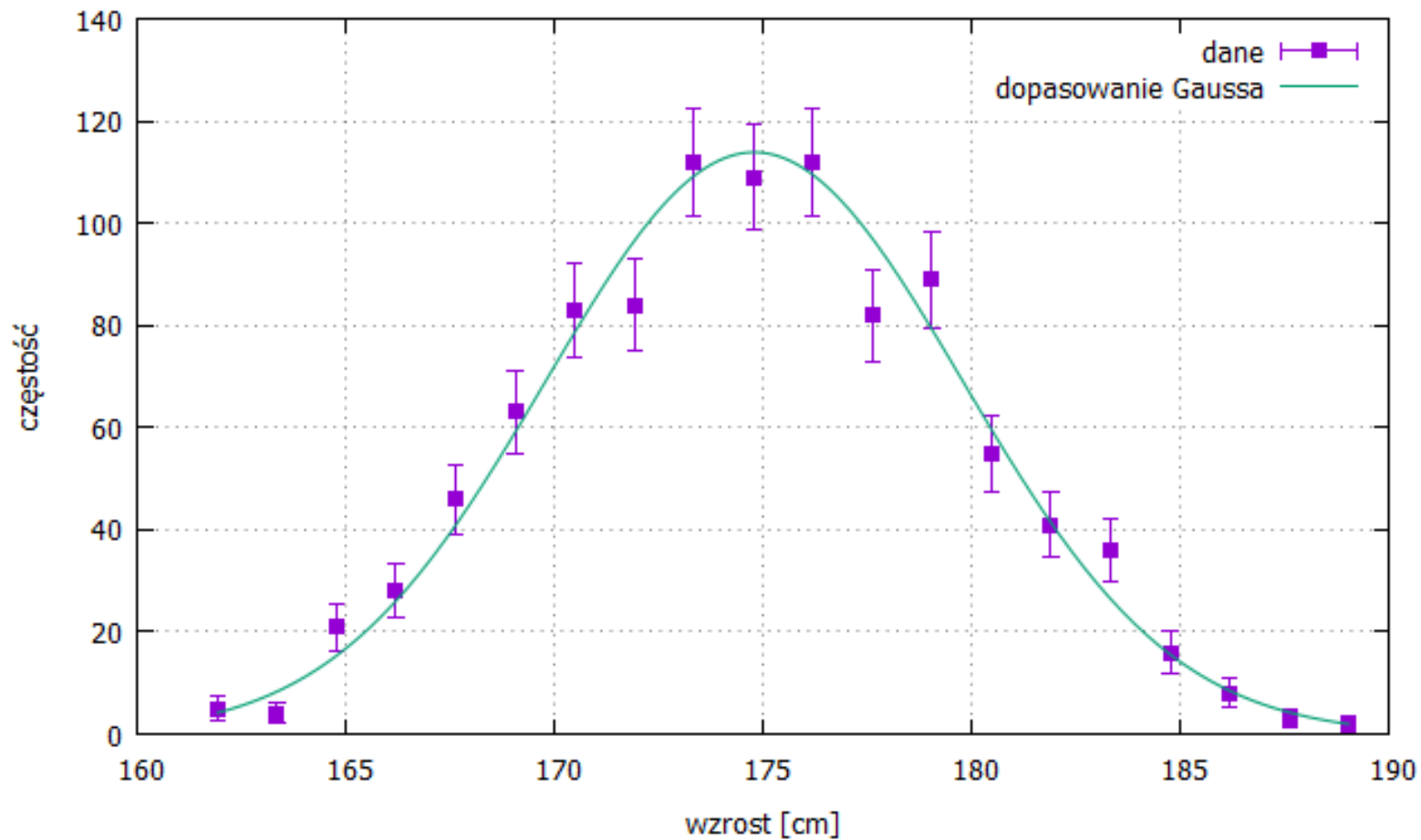
```
plot 'wzrost_studentow_bins.txt' using 1:2:3 with yerrorbars, f(x)
```

```
set xlabel 'wzrost [cm]'
```

```
set ylabel 'częstość'
```

```
plot 'wzrost_studentow_bins.txt' using 1:2:3 with yerrorbars pt 5 title 'dane',  
f(x)*999*1.43 w l lt 2 title 'dopasowanie Gaussa'
```

Dopasowanie Gaussa



www.if.pw.edu.pl/~martef/wzrost_studentow.txt

www.if.pw.edu.pl/~martef/wzrost_studentow_bins.txt

www.if.pw.edu.pl/~martef/wzrost_studentow_dane.txt

Tworzenie wykresów 3D

reset – resetuje wszystkie ustawienia

splot wartość - rysuje wykres podanego wyrażenia (funkcje dwuwymiarowe)

```
splot sin(x)*cos(y)
```

set pm3d - nakłada teksturę na siatkę

```
replot
```

set isosamples wartość - reguluje jakość wykresu

im większa wartość, tym lepsza jakość (przy większej wartości znacznie wzrasta czas tworzenia wykresu)

```
set isosamples 10
```

```
set isosamples 100
```

```
set hidden3d - ukrywa linie siatki
```

```
set pm3d map - tworzy mapę (rzut pionowy) wykresu
```

set palette - definiuje teksturę nakładaną na wykres

Przykłady

set palette positive – tekstura normalna

set palette negative – tekstura odwrócona względem positive

set palette gray – tekstura w odcieniach szarości

set palette rgbformulae 21, 22, 23 – przykładowa tekstura (0-36),

show palette rgbformulae – pokazuje zakres formuł mapowania kolorów rgb

Definiowanie własnej funkcji

Rozkład Gauss'a

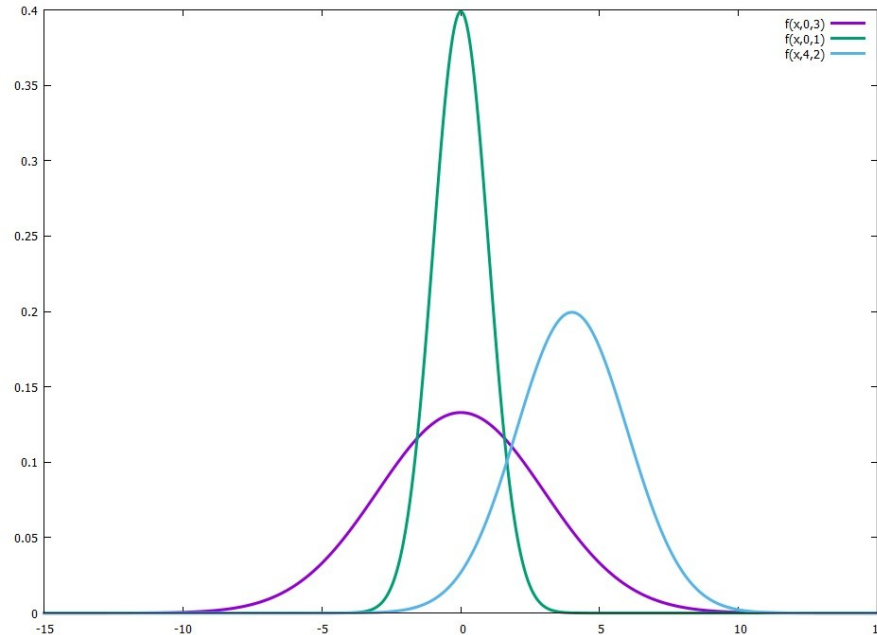
$$f_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Gdzie μ to wartość średnia, a σ to odchylenie standardowe rozkładu

```
set samples 10000
```

```
f(x,u,sigma)=1/(sqrt(2*pi)*sigma)*exp(-(x-u)**2/(2*sigma**2))
```

```
plot [-15:15] f(x,0,3) w lw 3,f(x,0,1) w lw 3, f(x,4,2) w lw 3
```



set samples 10000 - zwiększamy ilość punktów, gładki wykres

Wygląd wykresu

set key left, right, top, bottom, center, inside, outside, lmargin, rmargin, tmargin, bmargin (, above, over, below and under) - umieszcza w konkretnym miejscu legendę na wykresie

unset key – wyłącza legendę

show key – pokazuje ustawienia legendy

set label 1 'text' at x,y center - umieszcza napis w punkcie x,y, centruje tekst, 1 to oznaczenie napisu

set label 1 'hello' at 2,0 center

plot sin(x)

set arrow 1 from x1,y1 to x2,y2 - tworzy strzałkę biegnącą od punktu x1,y1 do x2,y2

set arrow 1 from 0,0.2 to 5,0.8

replot

set arrow from 2 x1,y1 to x2,y2 nohead - tworzy odcinek

set arrow from 2,0.2 to 8,0.7 nohead

replot

set grid - włącza siatkę

unset grid - wyłącza siatkę

set tics in - podziałka wewnątrz wykresu

set tics out - podziałka na zewnątrz wykresu

Tworzenie skryptów

Iteracja poleceń. Konstrukcja blokowa do

Wyświetlenie liczb od 1 do 10

```
do for [i=1:10] {print i;}
```

Jeśli potrzebuję wyświetlić, na jednym wspólnym wykresie, wiele podobnych danych lub funkcji to można to zrobić w sposób wygodny, przez iterację wspólnego polecenia plot.

Składnia:

```
plot for [<variable> = <start> : <end> {:<increment>}]
```

```
plot for [<variable> in "string of words"]
```

```
plot for [j=1:3] sin(j*x)
```

```
plot for [dataset in "galton Halleffect"] dataset. ".txt" title dataset
```

W powyższym przykładzie, iteracja jest używana zarówno do generowania nazwy pliku, jak i odpowiedniego tytułu.

Tworzenie skryptów

Sumowanie:

sum [<var> = <start> : <end>] <expression>

```
print sum [i=1:10] i
```

<var> jest traktowana jako zmienna całkowita, która przyjmuje kolejne wartości całkowite od <start> do <end>. Dla każdego z nich bieżąca wartość <expression> jest dodawana do sumy bieżącej, której wartość końcowa staje się wartością wyrażenia sumującego.

Tworzenie skryptów

```
a = 1  
load "skrypt1.txt"
```

```
a=a+1  
plot sin(x+a)  
pause 0.2  
if (a<30) reread
```

pause -1 – oczekiwanie na naciśnięcie klawisza ENTER
pause 1 – oczekiwanie przez 1 sekundę
pause 3 "Czekaj" – wyświetlanie napisu Czekaj przez 3 sekundy

```
u = 0  
sigma = 2  
load "skrypt2.txt"
```

```
set samples 10000  
f(x,u,sigma)=1/(sqrt(2*pi)*sigma)*exp(-(x-u)**2/(2*sigma**2))  
set xlabel "x"  
set ylabel "gęstość prawdopodobieństwa"  
set grid  
plot [-15:15] f(x,u,sigma) w l lw 3
```

Tworzenie skryptów

Cztery wykresy z różnymi wartościami

```
set multiplot layout 2,2
f(k,x)=cos(k*x)+2
do for [power=0:3]{
z=10**power
set title sprintf("%g cos",z)
plot sum [k=1:z] f(k,x) notitle
}
unset multiplot
```

Pojedyncze wykresy:

```
plot sum [k=1:10] cos(k*x)+2
```

```
plot sum [k=1:100] cos(k*x)+2
```

```
plot sum [k=1:1000] cos(k*x)+2
```