

GnuPlot

Program do tworzenia wykresów

Zbigniew Koza

Uniwersytet Wrocławski
Instytut Fizyki Teoretycznej

Możliwości programu

- Rysowanie wykresów funkcji i zbiorów danych.
- Rysowanie obiektów dwu- i trójwymiarowych.
- Współrzędne kartezjańskie, biegunowe i parametryczne.
- Dopasowywanie danych do funkcji (fitowanie).
- Transformacje danych.
- Różne funkcje matematyczne i liczby zespolone.
- Generowanie wykresów w różnych formatach graficznych.
- Generowanie wykresów złożonych (wstawki, tekst, osie logarytmiczne/liniowe, etc.).
- Generowanie wykresów w trybie wsadowym.
- Obliczanie wartości wyrażeń arytmetycznych („kalkulator naukowy”).

Sterowanie programem

- Sterowanie programem odbywa się poprzez **konsole**.
- Wysłankę włożony w naukę „języka” gnuplota zwróci się z nawiązką!



```
GNUPLOT
Version 3.8j patchlevel 0
last modified Wed Nov 27 20:49:08 GMT 2002
System: CYGWIN_NT-5.1 1.5.12(0.116/4/2)

Copyright(C) 1986 - 1993, 1999 - 2002
Thomas Williams, Colin Kelley and many others

This is a pre-version of gnuplot 4.0. Please refer to the documentation
for command syntax changes. The old syntax will be accepted throughout
the 4.0 series, but all save files use the new syntax.

Type `help` to access the on-line reference manual
The gnuplot FAQ is available from
  http://www.gnuplot.info/faq/

Send comments and requests for help to <info-gnuplot-beta@dartmouth.edu>
Send bugs, suggestions and mods to <info-gnuplot-beta@dartmouth.edu>

Terminal type set to 'x11'
gnuplot> p x*x
gnuplot> █
```

Posługiwanie się konsolą gnuplota

- W nazwach poleceń odróżnia się litery małe od dużych.
- Polecenia z reguły posiadają opcjonalne parametry.
- Nazwy poleceń i parametrów można skracać, np.
`plot 'plik.txt' using 1:3 with dots`
można zastąpić poleceniem
`p 'plik.txt' u 1:3 w dots`
- Polecenia mogą zająć kilka wierszy; kontynuację polecenia oznaczmy znakiem `\` umieszczonym na końcu wiersza.
- Napisy (np. nazwy plików) umieszcza się między znakami apostrofu (`'`) lub cudzysłowu (`"`)
`p 'dane.dat'`.
- Historię poleceń przewijamy klawiszami `↑` i `↓`.

Źródła informacji

- W programie:
 - `> help plot`
 - `> ?plot`
- W lokalnej dokumentacji programu
 - Plik `gpcard.pdf` (do wydruku!)
 - Plik `gnuplot.*`
(html, dvi, ps, pdf, info, hlp, nroff,...).
 - 44 pliki demonstracyjne w katalogu demo.
- W Internecie:
 - www.gnuplot.info

Wyrażenia arytmetyczne

- Funkcje matematyczne: `abs`, `acos`, `acosh`, `arg`, `asin`, `asinh`, `atan`, `atan2`, `atanh`, `besj0`, `besj1`, `besy0`, `besy1`, `ceil`, `cos`, `cosh`, `erf`, `erfc`, `exp`, `floor`, `gamma`, `ibeta`, `inverf`, `igamma`, `imag`, `invnorm`, `int`, `lgamma`, `log`, `log10`, `norm`, `rand`, `real`, `sgn`, `sin`, `sinh`, `sqrt`, `tan`, `tanh`.
- Liczby zespolone: `{1,0}`.
- Operator potęgowania: `**`.
- Operator „silnia”: `!`.
- „Numer kolumny”: `$`, np: `p 'x.txt' u ($1):($3)`.
- Pozostałe operatory jak w języku C: `!`, `~`, `&`, `|`, `*`, `%` (reszta z dzielenia), `/`, `+`, `-`, `==`, `!=`, `<`, `<=`, `>`, `>=`, `&&` (i), `||` (lub).
- Operator `? :`, np. `plot x < 0 ? -x : x`.

Definiowanie funkcji i zmiennych

- `> a = 12 # parametr użytkownika`
- `> f(x) = sin(x) + cos(a*x)`
`> plot [0:2*pi] f(x)`
- `> dist(x,y) = sqrt(x**2 + y**2)`
- `> g(x) = \`
`(x != 0) ? sin(x)/x : 1`
- Spis funkcji użytkownika: **show functions**
- Spis zmiennych: **show variables**
- Predefiniowana zmienna: **pi**

Rysowanie funkcji – zakresy

Wykresy funkcji tworzy polecenie **plot**

- wykres $f(x) = x^2 - 1$ w domyślnym zakresie:
plot x*x - 1
- to samo, ale krócej: **p x*x - 1**
- to samo, ale dla $0 \leq x \leq 10$:
p [0:10] x*x - 1
- to samo dla $0 \leq x \leq 10, -100 \leq y \leq 100$:
p [0:10][-100:100] x*x - 1
- to samo dla $-100 \leq y \leq 100$:
p [:-100:100] x*x - 1
- to samo dla $x \leq 10, -100 \leq y \leq 100$:
p [:10][-100:100] x*x - 1

Złożone wykresy funkcji

- Wykres 2 funkcji:

```
> p [0:2*pi] x, sin(x)
```

- Funkcja ograniczona do $\sin x \geq 0$:

```
> p [0:2*pi] sin(x)/(sin(x) >= 0)
```

Wartością wyrażenia logicznego jest 1 (prawda) lub 0 (fałsz). Gnuplot pomija punkty o nieokreślonej wartości, np. 1/0.

- Ciekawa funkcja:

```
> p [0:2*pi] (sin(x) >= 0.5) ? \
sin(x) : 0.5
```

- Rysowanie funkcji użytkownika:

```
> f(x) = sin(x)
```

```
> p [0:2*pi] f(x), f(-x), f(f(x))
```

Częstość próbkowania

- Domyślnie funkcję rysuje się jako łamaną z **samples** wierzchołkami.
- Zmiana wartości tej zmiennej:
> **set samples 500.**
- Wartość domyślna:
> **show samples**
 sampling rate is 100, 100
- Zmienna ta używana jest także podczas interpolacji.

Wykresy biegunowe i parametryczne

- Współrzędne biegunowe:
 - > **set polar**
 - > **p [0:4*pi] t** (1 krzywa)
 - > **p [0:4*pi] t, 2*t, 3*t** (3 krzywe)
- Wykresy parametryczne:
 - > **set parametric**
 - > **p sin(t), sin(2*t)**
- Z powrotem do nieparametrycznych wykresów $y = f(x)$:
 - > **unset parametric**
 - > **unset polar**
- Domyślną zmienną zależną jest **x** dla wykresów kartezjańskich, $y = y(x)$, i kąt **t** dla wykresów biegunowych, $r = r(t)$.

Osie logarytmiczne

- Logarytmiczna oś 'x', podstawa 10:
> **set logscale x**
- Logarytmiczna oś 'y', podstawa 2:
> **set logscale y 2**
- Logarytmiczne osie 'x' i 'y':
> **set logscale xy**
- Powrót do współrzędnych kartezjańskich:
> **unset logscale xy**
- To samo przy pomocy konsoli:
 - klawisz 'l' jest przełącznikiem trybu logarytmicznego na osi 'y';
 - klawisz 'L' jest przełącznikiem trybu logarytmicznego na osi bliższej pozycji kursora myszy.

Właściwości linii

- Styl linii ustala się w parametrze **with**:

```
> p [0:2] x*x with points
```

```
> p [0:2] x*x w p
```

- Dostępne rodzaje prezentacji danych:

lines	points	linespoints
impulses	dots	steps
fsteps	histeps	errorbars
xerrorbars	yerrorbars	xyerrorbars
boxes	boxerrorbars	boxxyerrorbars
financebars	candlesticks	vector

Szerokość, kolor i rodzaj linii

- Po parametrze **with** można podać dodatkowe opcje:
 - **linetype** (**lt**): rodzaj linii (kolor; linia ciągła, przerywana, punktowana,...).
 - **linewidth** (**lw**): szerokość linii
 - **pointtype** (**pt**): kształt symbolu
 - **pointsize** (**ps**): wielkość symbolu
- Po każdej z tych dodatkowych opcji należy podać liczbę całkowitą z przedziału $-1, \dots, 75$.
- Przykład:

```
> p x*x w linesp lt 2 lw 2 pt 4 ps 5
```

Komenda **test**

- Znaczenie numerycznych argumentów opcji **linetype**, **linewidth**, **pointtype** i **pointsize** najłatwiej określić na podstawie wydruku pliku [ps_symbols.ps](#) dołączanego do programu gnuplot.
- Inna metoda to uruchomienie komendy **test** i zrobienie stosownego wydruku.
- Każdy „terminal” inaczej interpretuje te liczby!

Wykresy zbiorów danych

Większość parametrów używanych przy rysowaniu funkcji ma też zastosowanie do wykresów zbiorów danych.

- Wykres danych w pliku dane.dat:
> **p 'dane.dat'**.
- Bardziej złożony przykład:
> **p [1:] 'dane.dat' w points ps 2 pt 4**
- Kilka wykresów na raz:
> **p [1:10] 'dane.dat' w p pt 4,\
sin(x) w l lt 4,\
'drugi.dat' w p pt 5**

Struktura plików z danymi

- Pliki z danymi mogą mieć dowolną liczbę kolumn:
`> p 'dane.dat' u 7:2`
- Pliki z danymi mogą zawierać dowolną liczbę „bloków” danych;
- Bloki oddziela się od siebie pustymi wierszami.
- Do bloków odwołujemy się opcją **index**:
`> p 'dane.dat' index 2:4`
- Numeracja bloków zaczyna się od 0.
- Wykres bloków 0,3,6,9,12:
`> p 'dane.dat' index 0:12:3`
- Jeszcze większe możliwości ma opcja **every**...

Dane ze znaną niepewnością

- Dane z estymatorem błędu pomiaru (3. kolumna):
> `p 'dane.dat' with errorbars`
- Inne tryby wyświetlania błędów:
 - `xerrorbars`
 - `yerrorbars`
 - `xyerrorbars`

Opcja `using`

Opcja `using` służy do manipulowania danymi.

- Wykres kolumny 3 względem 1:

```
p [:] 'dane.dat' u 1:3
```

- Wykres kolumny 2 względem $x = 0, 1, \dots$:

```
p 'dane.dat' u 0:1
```

- Wykres sumy kolumn 2 i 3 względem kol. 1:

```
p 'dane.dat' u 1:($2 + $3)
```

`$n` oznacza liczbę odczytaną z n -tej kolumny

- Transformacja danych:

```
p 'dane.dat' u 1:(log(abs($2)))
```

- Filtrowanie danych:

```
p 'dane.dat' u 1:($2 < 10 ? $2 :  
1/0)
```

Wygładzanie danych

- Aproksymacja danych krzywą ciągłą:
> plot 'dane.dat' u 1:2:(1.0) \
smooth acsplines
- Trzeci parametr opcji **using** określa stopień wygładzania danych; im jest mniejszy, tym wykres będzie słabiej „przyciągany” do punktów wykresu.
- Interpolacja danych krzywą ciągłą:
> plot 'dane.dat' smooth csplines

Opis osi

- Ustalenie tytułu osi:
`set xlabel "czas t"`
`set ylabel "energia"`
- Główne znaczniki osi:
`set xtics 5` (... , -5, 0, 5, ...)
`set ytics 0 2 10` (0, 2, 4, 6, 8, 10)
`set log x; set xtics 1,100,1e6`
(1, 10^2 , 10^4 , 10^6)
- Znaczniki pomocnicze (10 przedziałów, 9 znaczników):
`set mxtics 10`
- Chowanie prawego i górnego brzegu rysunku:
`set border 3`

Dodatkowe napisy

Do umieszczania dodatkowych napisów służy polecenie **set label**.

- Napis nr 3, „ $f(x)$ ”, obok punktu (1, 2):
> **set label 3 "f(x)" at 1,2**
- Domyślnie używa się układu współrzędnych zdefiniowanych przez osie wykresu.
- Napis x^2 wycentrowany w środku wykresu:
> **set label "x^2" **
at graph 0.5,0.5 center
- Napis „ π ” o wysokości 24pt:
> **set label "p" font "Symbol,24"**
- Informacja o wszystkich napisach
> **show label**

Dodatkowe strzałki i linie

Do umieszczania dodatkowych strzałek i linii służy polecenie **set arrow**.

- Strzałka nr 1 od punktu (1, 2) do (1, 3):
> **set arrow 1 from 1,2 to 1,3**
- Linia nr 3, gruba na 2pt, od (1, 2.5) do (1, 3):
> **set arrow 3 from 1,2.5 to 1,3
nohead lw 2**
- Usunięcie wszystkich strzałek i linii:
> **unset arrow**
- Usunięcie strzałki nr 3:
> **unset arrow 3**
- Wyświetlenie wszystkich strzałek i linii:
> **show arrow**

Legenda i tytuł rysunku

- Opis linii w legendzie:
`> p [0:2] x*x title "x^2"`
- Pominięcie opisu linii w legendzie: `> p [0:2] x*x notitle`
- Zdefiniowanie tytułu całego rysunku:
`> set title "Piękny rysunek"`

Położenie i rozmiar rysunku

- Standardowo położenie i rozmiar rysunku ustala się w „jednostkach ekranu”, w których terminal (kartka) ma wymiar 1×1 .
- Położenie lewego dolnego wierzchołka ustala się poleceniem `set origin`
`set origin 0.1, 0.1`
- Rozmiar rysunku ustala polecenie `set size`
`set size 0.5, 0.5`
- Równa długość osi:
`set size square`
- Odwzorowanie „geograficzne”:
`set size ratio -1`

Rysunek w rysunku

położenie całości:

```
> set origin 0,0
```

```
> set size 1,1
```

```
> set multiplot
```

parametry rysunku nr 1:

```
> set origin 0,0
```

```
> set size 1,0.45
```

```
> ...
```

```
> plot [-1:1] erfc(x)
```

parametry rysunku nr 2:

```
> set origin 0,0.5
```

```
> set size 1,0.45
```

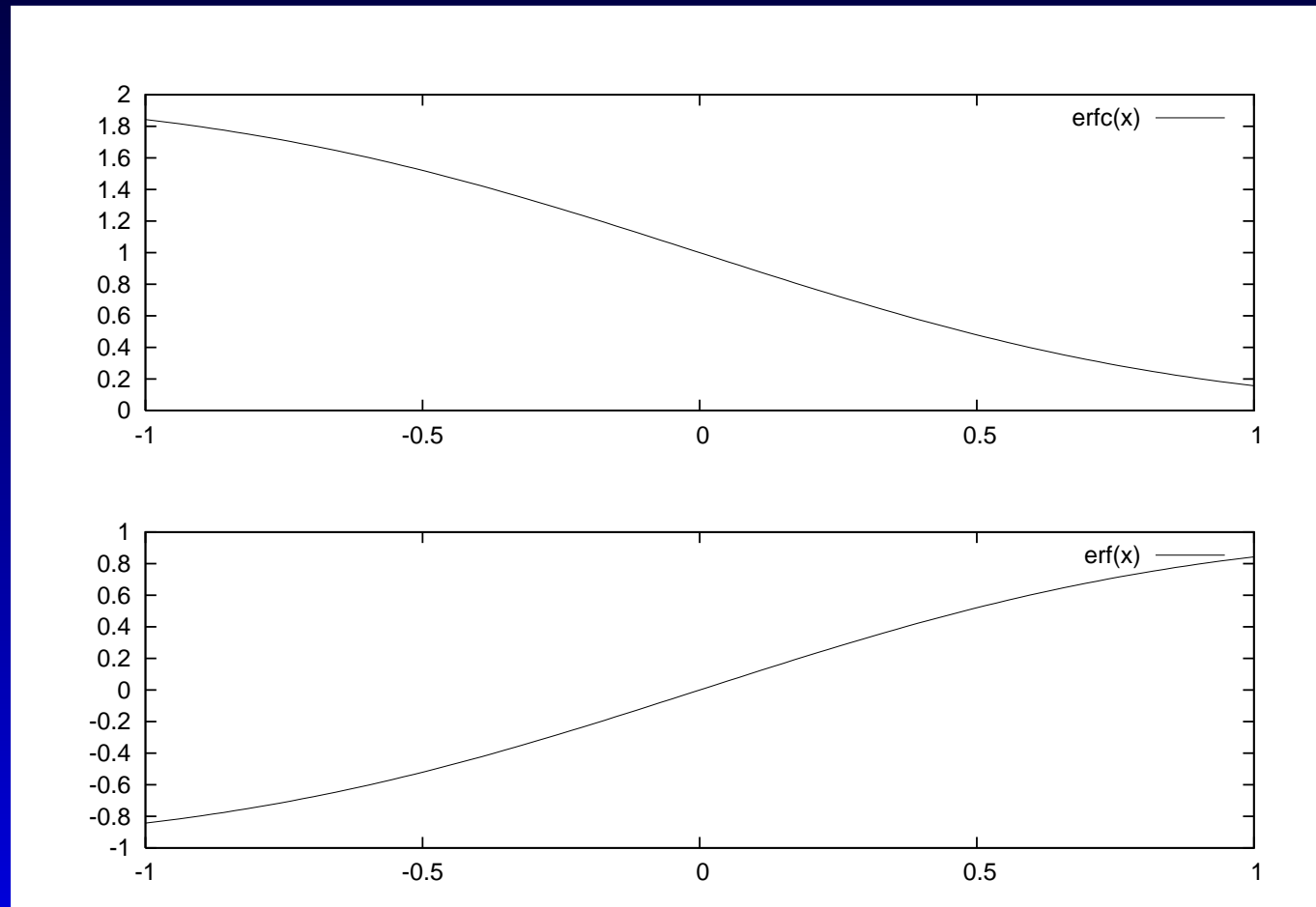
```
> ...
```

```
> plot [-1:1] erf(x)      rysunek #2
```

```
> unset multiplot
```

Rysunek w rysunku

Efekt:



Drukowanie do pliku

1. Zmiana terminala:

```
> set terminal postscript eps  
enhanced
```

2. Zmiana urządzenia wyjściowego:

```
> set output "rysunek.eps"
```

3. Zapis rysunku w pliku:

```
> p [1:3] f(x), 'w.dat'
```

4. Zamknięcie pliku:

```
> set output
```

5. Powrót do terminala „ekranowego”:

```
> set term X (X-windows)
```

```
> set term windows (MS Windows)
```

- Wykaz (≥ 35) dostępnych terminali:

```
> set term
```

Zapisywanie sesji

- Zapisanie zmiennych, funkcji, bieżących parametrów rysunku i ostatniego polecenia plot:
`save 'nazwa_pliku'`.
- Zapis zmiennych / funkcji / parametrów:
`save functions 'nazwa_pliku'`
`save var 'nazwa_pliku'`
`save set 'nazwa_pliku'`
- Odczytanie tego pliku:
`load 'nazwa_pliku'`
- Zapisanie historii sesji:
`history 'nazwa_pliku'`

Polecenia pomocnicze

- Wyświetlenie nazwy bieżącego katalogu:
`pwd`
- Zmiana bieżącego katalogu:
`cd 'podkatalog/inny_katalog'`