

Poprvé s gnuplotem

gnuplot je volně dostupný software pro kreslení dat ve dvou a třech dimenzích. Zobrazuje data ze **souborů** i analyticky zadané **funkce**, pro svou mrštnost je užitečný jako **kalkulačka**. Ovládá se z příkazového řádku, nezbytné je tak osvojit si základní **příkazy** a jejich varianty (a jejich zkratky), což pak lze užít k vytváření **skriptů**. Ke stažení je pro Windows i Linux, ve Windows bylo až do verze 4.2.6 možné spustit soubor wgnuplot.exe samotný (bez dll knihoven), ve verzi 4.4 ale přibyla podstatná syntaxe.

Start a stop

Příkazem **gnuplot** se spustí **gnuplot** v příkazovém režimu, **wgnuplot** ve Windows spustí **příkazové okno**. Po vydání příkazu ke kreslení (**plot**, **splot**) se může otevřít **grafické okno**; nemusí, je-li aktivní jiný výstupní terminál (**png**, **postscript** aj.). Z grafického okna k příkazovému řádku vrací **mezerník**. Většinu příkazů i jejich klauzulí lze krátit na jednoznačné **zkratky**, níže podtrhané a v příkladech užívané. **Skripty** lze spouštět z příkazového řádku **gnuplotu** i operačního systému.

wgnuplot , gnuplot	spustitelné soubory pro Windows a Linux
gnuplot script1 ...	spuštění skriptů z příkazového řádku operačního systému (s použitím příkazu load)
load 'scriptname'	spuštění skriptu z příkazového řádku gnuplotu
call 'scriptname' args	spuštění skriptu z gnuplotu , včetně řetězových argumentů (\$0...\$9 ve skriptu)
pwd , cd 'dir'	pohyb v adresářích: print working directory , change directory , př. Windows cd 'D:/MyData'
save opts 'file'	uložení parametrů gnuplotu , proměnných a funkcí
quit , exit , CtrlD	ukončení
gnuplot.pdf	dokumentační soubor, ve Windows také wgnuplot.chm

Klíčové příkazy

Nápovědu zajišťuje interní systém (příkaz **help**), v adresáři **gnuplotu** je pdf soubor s tímž obsahem; pomocnou informaci přinášejí i další příkazy (**test**, **set terminal**, **show colornames** aj.). Typicky: člověk začne od boku příkazem (2D) **plot** nebo (3D) **splot**, pak iterativně upravuje parametry obrázku pomocí příkazů **set** a návratem k **plot** (nebo jednoduše **replot**), př. **plot sin(x); set xrange [0:pi*2]; set grid; replot**. Výchozí **plot** kreslí do grafického okna, příkazem **set terminal** se zjistí dostupné výstupní formáty (**png**, **gif**, **pdf**, **postscript** aj.) a tímž příkazem se přepínají, příkaz **plot** pak musí obklopovat dvojice příkazů pro otevření a zavření souboru: **set terminal png; set output 'file'; plot sin(x); set output**. Lze kreslit i v textovém režimu: **set terminal dumb; plot x**2**.

help , help cmd	nápověda k příkazu nebo k jeho variantě, př. h set term , též jen set term ap .
print	výpis číselných a znakových výrazů, př. pr 'result = ',1
plot , splot , replot	kreslení dat v 2D a v 3D, překreslení posledního kreslení
set , unset , show , reset	nastavení a rušení parametrů, výpis stavu, reset, př. h set; set title 'Nadpis'; sh title; reset
test	informace o vlastnostech výstupního formátu
set terminal type	nastavení výstupního formátu: wxt , pngcairo , gif , pdfcairo , postscript , dumb (textově) aj. př. set t pngc size 800,600 (Windows: zastaralý terminál png nahrazován pngcairo) set t gif animate 100;
set output file ; set o	přesměrování zápisu do souboru a zpět, př. set t pngc; set o 'z.png'; replot; set o
unset terminal	nastavení výchozího terminálu (Windows: wxt nebo win , Linux: wxt nebo x11)
# ; \	pro skripty: řádkový komentář, oddělovač příkazů na řádku, pokračování na dalším řádku

Kalkulačka

Díky rychlosti startu a historii příkazů může **gnuplot** sloužit i jako pohotová kalkulačka: **pr 1+2*3, 2**3, 4!, tan(pi/4)**. Primární je ovšem schopnost vytěžit matematickou výbavu (operátory, standardní funkce) při kreslení analytických funkcí, **plot [-1:1] sqrt(1-x**2)**, a přepočtech dat ze souborů, **plot 'file' using 1:(\$2**2)** (kresleny kvadráty čísel v druhém sloupci). **Gnuplot** rozumí datovým typům (4bytový) **int**, (8bytový) **real** a **complex**, a **řetězcům**. Logické hodnoty splývají s numerickou nulou a nenulou. Řetězce se ve smysluplných případech převedou na číslo automaticky, podobně **integer** na řetězec. Lze definovat proměnné a jednopříkazové funkce (na velikosti písmen záleží).

print expression,...	vyhodnocení a výpis číselných a znakových výrazů, př. pr 'result = ',1
literály	int : 1, -1, real : 1., -2.2, 3e3, NaN, complex : {0,1}, quasi-logical: 0, !0, řetězce: 'a', "bcd"
konverzní funkce	int , real , imag : real(1),int(2.),imag({3,4}),real('5') ; automaticky: 1+'2', '1'. 2 (mezera za .)
var=expr , undef var , pi	přiřazení do proměnné, zrušení proměnné, proměnná pi , př. x=pi; pr x; undef x

<code>+ - * / % ** !</code>	aritmetické operátory: plus, minus, krát, dělení, zbytek po dělení, umocnění, faktoriál
<code>/</code>	celočíselné a reálné dělení, př. $1/2, 1./2$ pro 0 0.5
<code>.</code>	řetězení řetězců a integers, př. 'a'.bcd', 'file'. 01.'.dat' pro abcd file1.dat
<code>== != < <= ...; eq ne</code>	relační operátory, př. $10*.1==1$ pro 1, řetězcové relace: 'a' eq 'b' pro 0
<code>! && </code>	logické operátory: not, and, or, př. $!((1 \&\& 0) (0 \&\& 1))$ pro 1
<code>?:</code>	podmíněný operátor, př. $1 ? 'ano' : 'ne'$ pro ano
math functions	funkce z unixovské math knihovny pro int, real, complex argumenty: abs, acos, asin, atan, besj0, besj1, ceil, cos, cosh, exp, floor, gamma, log, log10, norm, rand, sgn, sin, sinh, sqrt, tan, tanh aj., viz help expressions functions
string functions	strlen, substr, word, words : <code>s='a b c'</code> ; <code>pr strlen(s),s[3:3],word(s,1),word(s,words(s))</code> 5bac sprintf('format',expr,...) : <code>sprintf('file%02d.dat',1)</code> pro file01.dat, „gnuplot“ varianta: gprintf
<code>func(var,...)=expr</code>	definované funkce, př. <code>f(x)=x**2</code> ; <code>g(x,y)=a*x*y</code> ; <code>min(a,b)=(a<b) ? a : b</code> ; volání: <code>a=2</code> ; <code>pr f(2),g(1,2),min(1.,min(2.,3.))</code> pro 4 4 1.0

Kreslení analytických funkcí v 2D

Těžko si představit pohotovější cestu ke grafu analyticky zadané **reálné funkce** reálné proměnné: (po svižném startu) zapsat znak **p** (příkaz **plot**), mezeru, předpis funkce a enter. V dalších iteracích se mohou upravit vlastnosti a popis os, vlastnosti čáry nebo použitých symbolů, legenda, nápisy atd. Jedním příkazem lze kreslit více funkcí, vlastnosti čar a symbolů automaticky rotují, jak – poradí příkaz **test**. (V gnuplotu 5 se změnilo pořadí barev na terminálově nezávislé, **set colorsequence default**, na dřívější barvy lze přepnout příkazem **set colorsequence classic**.) Funkce mohou být zadány parametricky, měřítko jednotlivých os může být logaritmické, lze upravit hustotu vzorkování. Podporu mají sekundární osy i datové a časové měřítko.

<code>plot func(x),...</code>	kreslení analytické funkce proměnné x : <code>p sin(x)</code> ; <code>p cos(x)</code> ; <code>p cos(x),sin(x)</code> pro 2 funkce
<code>set param; p fx(t),fy(t)</code>	funkce v parametrickém tvaru (symbol parametru: t): <code>set par</code> ; <code>p cos(t),sin(t)</code> ; <code>unset par</code>
<code>p [...:...] , set xrange ...</code>	rozsah ve směru jednotlivých os: <code>p [0:3][0:9] x**2</code> ; <code>set xrange restore</code>
<code>p with style</code>	styl čáry: points, lines, dots aj.: <code>p x w l</code>
<code>p linetype ...</code>	typ čáry nebo symbolu (viz příkaz test): <code>p x lt rgb 'yellow'</code>
<code>p title ...</code>	legenda k čáře: <code>p x t 'line 1'</code>
<code>set title ...</code>	nápis: <code>set title 'main title'</code> , dvouřádkový (\n) s uvozovkami: <code>set title "title\nsubtitle"</code>
<code>set xlabel, set xtics, set mxtics</code>	značení os: <code>set xla 'x'</code> ; <code>set xtic 0,5,10</code> format '%.2f'; <code>set mxt 4</code> ; <code>set xtic ('lo' 0,'hi' 10)</code>
<code>set key ...</code>	umístění popisu dat: <code>set key left top</code> ; <code>set key off</code> neboli <code>unset key</code>
<code>set logscale ...</code>	logaritmické osy: <code>set logscale xy</code> ; <code>unset log</code>
<code>set samples n</code>	počet bodů vzorkujících funkci, default: <code>set samples 100</code>
<code>set xzeroaxis, set grid</code>	vykreslení nulové souřadnicové čáry, vykreslení mřížky (též klávesa g v grafickém okně)
<code>h a g l u p n</code>	v grafickém okně: <code>help</code> , <code>autoscale</code> , <code>grid</code> , <code>logscale</code> , <code>unzoom</code> , <code>previous/next zoom</code> aj.
výřezy	výřezy pravým tlačítkem myši, klávesa u pro unzoom
mezerník	návrat z grafického do příkazového okna

Kreslení 2D dat ze souboru

Při kreslení ze souborů gnuplot použije data v prvním sloupci pro **x** souřadnice, data v druhém sloupci pro **y** souřadnice, automaticky zvolí vhodný rozsah v obou osách a vynesou default symbol pro každý řádek. Pro jednosloupcové soubory vynesou závislost dat na číslu řádku (počínaje 0). Lze změnit sloupec s daty (číslo řádku je virtuálním nultým sloupcem), lze přeskočit hlavičku souboru, lze přeskakovat řádky, lze se omezit na bloky řádků (navzájem oddělené 2 prázdnými řádky). Data ve sloupcích lze transformovat libovolným výrazem. Čára prokládaná daty je po částech lomená, může být i zhlazená. 2D data mohou být doplněna dalšími sloupci o chybový interval nebo o index barvy.

<code>plot file,...</code>	kreslení dat ze souboru, př. <code>plot 's1.dat','s2.dat'; f='s3.dat'; p f</code>
<code>index s1, i s1:s2</code>	výběr datové sekce (indexování od 0, sekce oddělená 2 prázdnými řádky), př. <code>p f i 0</code>
<code>every ip:ib:sp:sb:ep:eb</code>	krokování v bodech a sekcích, increment/start/end, př. <code>p f ev 2::0:::</code>
<code>using x:y, 0, \$i,</code> <code>" (dva apostrofy)</code>	výběr sloupců, číslo řádku, sloupcový operátor, př. <code>p f u 0:1, p f u 1:(\$1**2)</code> naposledy použitý soubor, př. <code>p 's1.dat' u 1:2," u 1:3</code>
<code>column(i), \$i</code>	i-tý sloupec v souboru
<code>with points lines dots</code>	výběr propojení bodů, př. <code>p f u 1:2 w l</code>
<code>u x:y:dx:dy w vectors</code>	propojení bodů (x,y) a (x+dx,y+dy) vektory, př. <code>p 1:2:3:4 w vect head filled</code>
<code>u x:y:z palette z</code>	obarvení podle třetího sloupce, <code>set palette</code> a <code>set cbrange</code> pro nastavení palety a color-boxu př. <code>set pale defi (0 'black',1 'yellow')</code> ; <code>set cbra [min:max]</code> ; <code>p f u 1:2:3 pale z w d</code>
<code>show colornames</code>	seznam jmen barev

Zobrazení 3D dat

splot $f(x,y)$, file u 1:2:3 kreslení 3D analytických funkcí a dat ze souborů
set pm3d pro barevné plochy a průřezy, **set palette** a **set cbrange** jako výše
set view map pro 2D mapu 3D plochy
set contour base|surface|both pro izočáry, **set cntrparams** pro jejich nastavení, př. **set co bo**; **set cn levels 10**

Další možnosti zobrazení

set multiplot více panelů (plots) na jedné obrazovce (screen)
set size změna velikosti panelu
set origin posun počátku
unset multiplot návrat k 1 plot/screen

Numerické metody

Gnuplot zpřístupňuje uživateli **metodu nejmenších čtverců**: po definování funkce s vnějšími parametry se volá **příkaz fit** pro vyčíslení parametrů tak, aby funkce ve smyslu metody nejmenších čtverců co nejlépe aproximovala data ze souboru. Jiné numerické metody jsou skryty pod kapotou, jako např. interpolace pomocí **kubických splinů**.
fit expr file **via** params fitování (nelineární) metodou nejmenších čtverců
př. $f(x)=a*x+b$; **fit f(x) file using 1:2 via a,b**; **plot file,f(x)**
plot file smooth csplines zhlazení dat kubickými spliny

Příkazy

Nedávným přírůstkem ve (skriptovacích) schopnostech gnuplotu je **příkaz cyklu do for** v indexované a výčtové variantě; **klauzuli for** lze použít i samostatně v příkazech plot, splot, set a unset. Přibyl také **podmíněný příkaz if**, blokový i jednořádkový.

do for [...] {cmds} indexovaný cyklus: **do for [i=1:10] {print i}; s=0; do for [i=1:10:2] {s=s+i}; print s**
výčtový cyklus: **do for [i in '1 2.3 a bc'] {print i}**
plot for [...] clauses př. **plot for [c=2:3] 'file' using 1:(column(c)) w l**; **plot for [file in 'a.dat b.dat'] file title file**
set for [...] parameter př. **set for [i=1:4] style line i pointsize i pointtype 7; set key left; set samples 10;**
p [0:1] for [i=1:4] xi ls i w lp t 'x^'.i # linestyle i with linespoints title 'x^'.i**
if (cond) {cmds} else {cmds} **blokový if**: **x=1; if (x>0) {print 'pos'} else {if (x<0) {print 'neg'}} else {print 'zer'}}**
if (cond) cmds ; else cmds **jednořádkový if**: **x=1; if (x>0) print 'pos'; else if (x<0) print 'neg'; else print 'zer'**

Data z binárních souborů

Programy v C, Fortranu aj. mohou pro úsporu místa, uchování přesnosti i vyšší rychlost vytváření zapisovat data do binárních souborů. Gnuplot binární data čte (klauzule **binary**, **format** ad. příkazů plot a splot) rychleji než z textových souborů.

p file binary filetype=auto w rgbalpha zobrazení png/jpg souborů; další formáty: **show datafile binary**
p file bin format='%i%f%lf' u x:y kreslení 2D dat; formátování sloupců s int, float/real(4), long float/real(8)
skip=s record=r form='%*pi%qf' skok přes s bytů, čtení r řádků se skoky přes 4p bytů na q real sloupců
př. **p f bin form='%i%2f' rec=50 u 1:2," bin form='%i%2f' skip=50*12 u 1:3," bin form='%i%*f%f' u 1:2 w l**
p file bin array=mxn form='%f' w image vizualizace matice o velikosti mxn (fortranský sloupcový index je na ose x)
sp file bin rec=mxn form='%4f' w pm3d vizualizace plochy na síti mxn: souřadnice x, y, z a 1 datový sloupec
Pravidelnou síť lze tedy generovat automaticky pomocí **array** (s užitím **origin**, **dx**, **dy** aj.), jinak se síť načítá spolu s daty pomocí **record**. Problém s řazením bytů může vyřešit **endian=swap**. Pro png/jpg, **format** s multiplikátory a syntaxi **array**, resp. **record=(m,n)** je třeba verze gnuplotu vyšší než 4.2.6.
Fortran a binární zápis: **open(id,file='f',form='unformatted',access='stream',status='replace'); write(id) a; close(id)**

Odkazy

www.gnuplot.info, download www.gnuplot.info/download.html
dokumentace v souborech gnuplot.pdf, gpcard.pdf, wgnuplot.chm (Windows)