GNUplot gyorstalpaló

2002.04.03.

Ez a dokumentum javarészt Borsányi Szabolcs ELTE-s fizikus „GNUplot howto oldal” c. weboldalának (www.cs.elte.hu/~mazsx/gnuplot) felhasználásával készült.

Első lépések

A gnuplot egy ingyenes ábrarajzoló és illesztő program. Megtalálható minden Linux disztribúcióban, de létezik Windowsos változata is.

Indítsuk el a programot!

```
$ gnuplot
```

Ekkor bejelentkezik a program és egy bevezető szöveg után kiírja, hogy

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

Ez azt jelenti, hogy a kívánt grafikon egy új X-es ablakban fog megjelenni. A program ezek után vár egy parancsot. Pl.:

```
gnuplot> print "Helló világ!"
Helló világ!
```

és enter után végre is hajtja. A parancsok egyenrangú argumentumait vesszővel választjuk el. A karakterláncokat idézőjelbe kell tenni, különben azt hiszi, hogy valamilyen változóról van szó. Ha a gnuplot parancsának egyik argumentumában kiszámítható kifejezés van, azt ki is számítja.

A műveleti jelek nagyjából megfelelnek a C programozási nyelv műveleti jeleinek (**,+,–,*,/,& stb.). Ezen kívül ismer egy rakás matematikai függvényt (ezeket most nem sorolom fel, a program helpjéből előbányászhatóak), és képes kezelni komplex számokat is, amiket {<real>,<imag>} alakban kell megadni.

```
gnuplot> print {1,2}*{1,2}
{-3.0, 4.0}
gnuplot> print abs({3,4})
5.0
gnuplot> print arg({3,4})
0.927295218001612
```

Változók

Ha beírjuk:

```
gnuplot> a=1
gnuplot> print a
1
```

akkor értéket adunk az „a” változónak és kiolvassuk. Új függvényeket is definiálhatunk:

```
gnuplot> g(x)=sin(2*x)/pi
gnuplot> print g(3)
-0.0889407154296873
```

A „pi” egy előredefiniált változó. Értéke értelemszerűen π.

Segítségkérés, a `help (h)` parancs

```
gnuplot> help {parancsnév}
```

Itt és mindenütt a `gnuplot` helpjében a kapcsos zárójel opcionális részt jelöl, a szögletes zárójelben függőleges vonallal elválasztott részek pedig vagylagosak.

Rövidített parancsnevek

A `gnuplot` parancsokat nem kell végig begépelni, mert vannak rövidítéseik is; így kevésbé kopnak a billentyűk és az ujjaink. A `pint` parancs rövidítése például `pr`. A parancsok minden szavából elég csak annyit kiírni, amennyiből már egyértelmű, hogy mire gondoltunk. Tehát a

```
gnuplot> se fu s ye
gnuplot> set function style yerrorbars
```

sorok ekvivalensek.

Beállítások – `set, show`

Rengeteg állítható kapcsolóval rendelkezik a `gnuplot`. Ezeket a `set` parancs segítségével állíthatjuk be, és a `show` paranccsal olvashatjuk ki. Pl. a

```
gnuplot> set logscale
```

átállítja az alapértelmezett grafikonmegjelenítést logaritmikus skálájúra. A `show all` utasítással megkaphatjuk az összes beállítást.

Rajzolás

Két dimenziós grafikonokat a `plot` (röviden `p`) paranccsal készíthetünk. Alapértelmezésként a függvényeket a $-10..10$ intervallumon rajzolja ki. A független változó neve mindig `x`.

Határok megadása

Kis ujjgyakorlat:

```
gnuplot> hullam(x)=sin(2*x)/3
gnuplot> p [0:3/pi] hullam(x/2)*3
```

A `pi` ugyebár előredefiniált változó, és amint láthatjuk, a határmegadásban lehet műveleteket is végezni. Az ordináta és abszcissza tengely egyidejű korlátozása:

```
gnuplot> plot [-1:1] [-20:20] 1/x
```

Csak az abszcissza:

```
gnuplot> plot [] [0:1] 1/sqrt(x*x)
```

Ha csak az egyik oldalról akarunk határt szabni:

```
gnuplot> plot [0.1:]  log(x)
```

A `set [x|y]range` paranccsal is be lehet állítani a határokat a `plot` kiadása előtt. A hatása olyan, mintha a `plot`-nál adtuk volna meg a határt, de az összes későbbi grafikonra az így beállított intervallum lesz érvényes. A default értékeket a `set autoscale` utasítással állíthatjuk vissza.

Csillagok fénygörbéinek kirajzolásakor a függőleges tengelyt a magnitúdóskála sajátossága miatt illik megfordítani. Ezt a legelegánsabban a `set yrange [] reverse` utasítással tehetjük meg.

Logaritmikus skála – `set logscale`

`set logscale (set log)`: log-log skálára való áttérés

`set nologscale (set nolog)`: a lineáris skála visszállítása

`set logscale x (set log x)`: csak az x tengely (létezik y-os analogonja)

`set nologscale x (set nolog x)`: a tengely újra lineáris

`set logscale y 2`: az y tengelyt kettes logaritmussal skálázza (a 10-es alapú logaritmus a default)

Polárkoordináták

Polárkoordinátákra a `set polar` utasítással térhetünk át (visszaállás a Descartes-koordinátákra: `set nopolar`). Ilyenkor a `plot` egy sugár(szög) alakú függvényt vár. A szögváltozó alapértelmezésben a „t”. Pl. egy függvény a kedvesünknek:

```
gnuplot> set polar
gnuplot> plot [-pi/3:3*pi/2] abs(t-pi/2)*(pi+abs(t-pi/2))**(-0.1)
```

A szögeket alapértelmezésként radiánban méri a `gnuplot`. Ettől eltérni a `set angles degrees` paranccsal lehet (visszaállítás: `set angles radians`).

Több görbe egy ábrán

A legegyszerűbb módszer, ha a `plot`-nak vesszőkkel elválasztva több görbét adunk meg:

```
gnuplot> plot [0:1] x,x**2,x**3
```

A megadott intervallum a görbék értelmezési tartománya. Másik lehetőség a `replot` parancs. Ez megtartja az előző görbét, és rárajzolja az argumentumában megadott függvényt. A fenti ábrát így is kirajzolhattuk volna:

```
gnuplot> plot [0:1] x
gnuplot> replot x**2,x**3
```

A `replot` parancson belül a határok állítására nincs mód. Harmadik lehetőség a `multiplot` környezet (`set multiplot, set nomultiplot`), amin belül a kirajzolt függvények egymásra kerülnek.

Rajzolás fájlból

Tegyük fel, hogy egy „meres.dat” nevű fájlban három számoszlop van. A

```
gnuplot> plot "meres.dat" using 1:2
```

utasítással kirajzolhatjuk a második oszlop értékeit az első függvényében. Láthatjuk, hogy idézőjelbe kell tenni a fájl nevét, mint minden string változót.

Előfordulhat, hogy a második oszlop gyökének harmadik oszlopszorosát akarjuk látni az első oszlop szinuszának függvényében. Ennek sincs akadálya:

```
gnuplot> plot "meres.dat" u (sin($1)):(sqrt($2)*$3)
```

Hibasávok (error bars) megadása

Mért, szimulált, vagy illesztett adatok előszeretettel rendelkeznek hibával. Tegyük fel, hogy az előző példában szereplő adtafájl harmadik oszlopa az y tengely mentén megjelenő hibákat adja meg. A hibasávok kirajzolása az alábbi módon történik:

```
gnuplot> plot "meres.dat" using 1:2:3 with yerrorbars
```

vagy rövidebben

```
gnuplot> p "meres.dat" u 1:2:3 w ye
```

Ha a negyedik oszlopban megvan az x értékek hibája, akkor ebben az irányban is ki lehet rajzolni a hibasávokat:

```
gnuplot> p "meres.dat" u 1:2:4:3 w xye
```

ahol `xye` az `xyerrorbars` rövidítése

Formázás

Ebben a fejezetben megtudhatjuk, hogyan lehet ellátni az ábrát feliratokkal, hogyan lehet segédvonalakat behúzni, miként tudjuk megadni a grafikon típusát stb.

Görbék és ponthalmazok stílusbeállításai

Ha kirajzolunk egy görbét a `plot` paranccsal, akkor az alapértelmezésnek megfelelően az első görbét piros színnel rajzolja ki a program (a következőt zölddel, majd kékkel stb.). A tengelyeket nem látja el felirattal, és megjelenik a jobb felső sarokban egy jelmagyarázat, melyen a görbe, az ábrázolt függvény képlete, vagy a használt fájlnév és using opció olvasható le. A képleteket alapértelmezésben vonallal, az adatfájlokat pöttyel rajzolja.

Vonal- és pöttytípus

A `plot` parancsba beszúrt `with` opcióval állíthatjuk be a megjelenítési stílust:

```
gnuplot> p 'meres.dat' u 1:2 with lines
```

Hatására a második oszlopot az első függvényében ábrázolja, és a pontokat egyenessel köti össze. Összekötögetés helyett *spline*-okat is illeszthetünk a pontokra a `smooth {unique | csplines | acsplines | bezier | sbezier}` kapcsolóval. Pl. ha egyszerű köbös spline-okat akarunk illeszteni:

```
gnuplot> p 'meres.dat' u 1:2 smooth csplines
```

A fenti parancsokat természetesen lehet rövidíteni is. A `lines` paraméter helyébe mást is írhatunk:

stílus	rövidítés	hatás
`lines`	`l`	szakaszokkal köti össze a pontokat
`points`	`p`	pöttyökkel rajzol
`linespoints`	`lp`	pöttyöz, és még össze is köti
`dots`	`d`	leheletfinom pontok
`impulses`	`i`	függőleges szakasszal köti össze az (x,min)(x,max) pontokat
`xerrorbars`	`xe`	az x irányú hibákat jelöli a `using` 3. paraméterének megfelelően
`yerrorbars`	`ye`	az y irányú hibákat jelöli a `using` 3. paraméterének megfelelően
`steps`	`st`	lépcsőt rajzol felfelé: (x[i],y[i])→(x[i+1],y[i]), majd: (x[i+1],y[i])→(x[i+1],y[i+1])
`fsteps`	`fs`	lépcsőt rajzol lefelé: (x[i],y[i])→(x[i],y[i+1]), majd: (x[i],y[i+1])→(x[i+1],y[i+1])
`histeps`	`his`	lépcsőt, de a vízszintes vonal közepén van a mérési pont (hisztogramokhoz)
`boxes`	`boxes`	oszlopdiagram
`boxerrorbars`	`boxer`	hiba az oszlopdiagramhoz
`vector`	`vec`	nyilat húz a `using`ban megadott első és második két oszlop pontpárjai közé

Színek és stílusok

A `with` opcióval még sokmindent beállíthatunk:

```
guplot> p sin(x) w l lt 2
```

Ezzel a kettes vonaltípust választottuk, ami a képernyőn zöld színű folytonos görbe, kinyomtatva szaggatott. A program nyolc vonaltípust ismer (–1,0,1,2,3,4,5,6); több görbe esetén ezeket ismétli. Az `lt` opciót el is hagyhatjuk:

```
gnuplot> p sin(x) w l 2
```

A vonalvastagság beállítására az `lw` kapcsoló szolgál:

```
gnuplot> p cos(x) w l lt 4 lw 5
```

A sorrend fontos! Az `lt 0` fekete, a görbére merőleges vonalkákkal rajzolja ki a grafikont. Ilyenkor az `lw` ezen vonalkák hosszát állítja be. Az `lt -1` fekete, folytonos vonalat húz az `lw`-vel megadott vonalvastagsággal.

A pöttyök stílusa is átírható:

```
gnuplot> p sin(x) w p pt 3 ps 4
```

A `pt` a típust, a `ps` a méretet állítja. Egyszerre is működik a vonal- és pontbeállítás, ha a `linespoints` stílust használjuk.

Az ábra feliratozása

Az ábra címe

A `set title` paranccsal az egész ábra fölé írhatunk szöveget (címet):

```
gnuplot> set title "A kedvenc szinuszgörbém"
```

A cím csak egy `replot`, vagy `plot` parancs kiadása után tűnik elő, és minden további ploton ott lesz. Levenni a

```
gnuplot> set title
```

paranccsal lehet. A cím az ábra felett, középen jelenik meg. A helyét a következő módon lehet változtatni:

```
gnuplot> set title "A kedvenc ..." 1,3
```

Itt `1,3` az alapértelmezett helytől mért, karakterméret egységben megadott relatív koordinátákat jelenti. A parancs végén idézőjelben megadhatjuk a betűtípust és betűméretet, de ez csak nyomtatásban látszik majd:

```
gnuplot> set title "A kedvenc ..." 1,3 "Helvetica, 13"
```

Jelmagyarázat és feliratok

Hogy ne az jelenjen meg a jelmagyarázatban, ami a `plot` parancssorában, hanem amit mi szeretnénk:

```
gnuplot> p sin(2*x) title "görbe" w p pt 2 ps 5
```

Ha `using`ot is használunk, azt a fájl/görbenév és a `title` közé kell írni! Ha ki akarjuk kapcsolni a jelmagyarázatot:

```
gnuplot> p sin(2*x) notitle w p pt 2 ps 5
```

A `set label "szöveg"` paranccsal az ábra tetszőleges pontjára írhatunk feliratokat.

```
gnuplot> set label "valami"
```

Ha nem adunk meg neki koordinátákat, a `plot` az ábra origójába helyezi a feliratot. A pozícionálás az `at` kapcsolóval történik. A pozíciót a grafikon koordináta-rendszerében kell megadni. Ezenkívül beállíthatjuk még a betűtípust, betűméretet, 90 fokos forgatást és az igazítást.

```
gnuplot> set label "felirat" at 0.5,0.5 center rotate font "Helvetica, 16"
```

De a forgatás és a betűtípus csak nyomtatásban jön elő. Az igazítás opció `left, right,` vagy `center` lehet. Most nézzük meg a `show label` paranccsal a feliratokat:

```
gnuplot> show label
label 1 "valami" at (0, 0, 0) left not rotated
label 2 "felirat" at (0.5, 0.5, 0) centre rotated (if possible) font "Helvetica, 16"
```

A feliratok automatikusan sorszámozódnak. Ez nagyon fontos lehet, ha törölni akarjuk valamelyik feliratot:

```
gnuplot> set nolabel 1
```

A tengelyek feliratozására szolgál az `xlabel` és `ylabel` parancs. Használata a `label`-éhez hasonló, de iagzítani nem lehet (mindig `center` marad).

```
gnuplot> set xlabel "Ez a vízszintes tengely" font "Times, 14"
```

Az `ylabel` felirata a képernyőn hülye helyre kerül (az y tengely tetejére, vízszintesen), de nyomtatásban már jó lesz (párhuzamosan a tengellyel, középre rendezve).

Scriptek

A `gnuplot` nem csak interaktív módban képes fogadni a parancsokat, hanem scriptek formájában, fájlból is be tudja olvasni. A script egy olyan fájl, melyben `gnuplot` parancsok találhatók, amiket sorban végrehajt a program. Konvenció, hogy `.plt` kiterjesztést adunk a `gnuplot` scripteknek.

Tetszőleges ascii szövegszerkesztővel (pico, joe, vi, nedit, emacs stb.) írhatunk scriptet, álljon itt egy egyszerű példa (rajz.plt):

```
p [0:2*pi] sin(x) w l lt -1 lw 2
pause -1 "Nyomd meg az entert!"
```

A `pause -1` utasítás kiír egy üzenetet,majd `enter`re vár.

A scriptet ezután legegyszerűbben a

```
[csakb@localhost abra]$ gnuplot rajz.plt
```

paranccsal lehet shellből futtatni. Ekkor megjelenik a fekete szinuszgörbe, vár egy entert, majd megszakad a program futása. De interaktív módban is lehet scriptet futtatni a `load` parancs meghívásával:

```
gnuplot> load "rajz.plt"
```

Szintén lefut a script, de a végén nem áll le teljesen a program, hanem visszakapjuk a `gnuplot` promptot.

Parancsok mentése

Előfordulhat, hogy kínkeservvel létrehozunk egy grafikont, amit a későbbiekben is szeretnénk reprodukálni. Ennek érdekében el lehet menteni a begépelt parancsokat:

```
gnuplot> save "rajzom.plt"
```

A mentett parancsok `load`dal való betöltése bármikor visszavarázsolja a grafikont. A felhasználói függvények, változók, beállítások és az utolsó plot parancs kerülnek elmentésre.

Nyomtatás

Mint azt láttuk, a `gnuplot` ábrát lehet a parancsok mentésével, scriptbe írásával reprodukálni, de lehetőségünk van hordozható formátumúvá alakítani, azaz elmenteni gif, postscript, latex, xfig vagy egyéb formátumba.

Ha interaktív módban vagyunk, és elkészült a nyomtatható ábránk:

```
gnuplot> set terminal postscript enhanced
gnuplot> set out "rajz.eps"
gnuplot> plot ...
gnuplot> set out
gnuplot> set terminal x11
```

A harmadik sor helyett `replot`ot is írhatunk, ha előzőleg a kinyomtatandó ábra volt megjelenítve. Az első sor átállítja a kimenet típusát postscriptté, majd a kimenő fájl nevét adjuk meg. Utána visszaállítjuk, hogy a további rajzok ne menjenek ugyanabba a fájlba. A képernyőre rajzolás neve az `x11`.

Érdemes a következő scriptet megírni (`ment.plt`):

```
set terminal postscript enhanced
set out "$0"
replot
set out
set terminal x11
```

A `"$0"` a második sorban a script argumentuma, amit a meghívása során kell megadnunk. Jelen esetben a fájl nevét, amibe el szeretnénk menteni a grafikonunkat. Ezután, ha a sok-sok bíbelődés után sikerült olyan ábrát készíteni, amit el szeretnénk menteni postscript formátumba, csak üssük be:

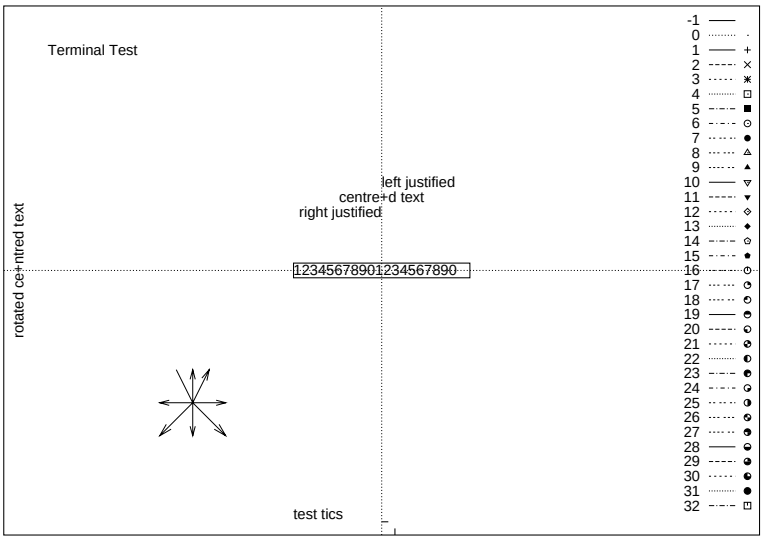
```
gnuplot> call "ment.plt" "joabra.eps"
```

A postscript terminál

A postscript terminál opciói:

```
gnuplot> set terminal postscript  {<mode>} {enhanced | noenhanced}
                                  {color | monochrome} {solid | dashed}
                                  {<duplexing>}
                                  {"<fontname>"} {<fontsize>}
```

Az első kapcsoló megadja, hogy a papíron hogyan helyezkedjen el az ábra. Nagy, egyoldalas ábrához használjuk a `portrait`-et, publikációhoz `eps`-t. A második kapcsoló `enhanced postscript` módba kapcsolja a terminál driverét, aktiválva ezzel egy csomó speckó lehetőséget. A harmadik kapcsolóval megadhatjuk, hogy színes ábrát szeretnénk, vagy fekete-fehéret. A negyedik opció azt állítja be, hogy fekete-fehér ábrán a vonalak csak folytonosak, vagy különféle szaggatott stílusúak legyenek. Az ötödik kapcsolóval kétoldalas ábrát készíthetünk; a hatodik opció pedig a betűk típusát és méretét szabályozza. Postscript módban hatnál több vonaltípus és 26 ponttípus létezik. Ezeket az alábbi ábra foglalja össze.



Az `enhanced postscript` mód

Itt rövid összefoglalása következik a David Denholm és Matt Heffron által fejlesztett postscript mód parancsainak, amivel görög és más speciális karaktereket lehet beépíteni az ábrákba.

A parancsokat a `gnuplot`ban előforduló szövegekbe (címek, címkék stb.) kell beszúrni. Az irodalom végén egy összefoglaló táblázat található a kódokról. Felhívnám a figyelmet arra, hogy ezek a parancsok csak enhanced postscript módban működnek, képernyőre rajzoláskor feldolgozatlanul jelennek meg. Az itt közölt anyag Dick Crawford nyomán készült, az eredeti háromoldalas PDF dokumentum alapján.

Példa indexekre, görög és más furcsa betűkre:

```
plot "valami.dat" u 1:2 title "A_Z" w l
plot "" u 1:2 title "l^{Symbol n}" w lp
set title "I {ZapfDingbats \250} \351"
```

Ha ékezetes betűket is szeretnénk látni az ábránkon, akkor mindenképp írjuk be a következő parancsot:

```
gnuplot> set encoding iso_8859_1
```

különben csak mindenféle kriksz-kraksz lesz helyettük a képen.

Illesztések

A `gnuplot` alkalmas tetszőleges függvényalak illesztésére adatfájlok pontjaira.

Legyen egy `meres.dat` fájlunk valamilyen mért adatpárokkal. Illesszünk rá egyenest:

```
gnuplot> fit a+b*x "meres.dat" u 1:2 via a,b
```

Ekkor elkezd iterálni, és előbb-utóbb konvergál az a és b értékekhez, majd kiírja az illesztett paraméterek végleges értékeit, az illesztés hibáját, az eltérésnégyzetet stb. Megadhatjuk az illesztési tartományt és a pontok hibáit is (3. oszlop):

```
gnuplot> fit [0:4] a+b*x "meres.dat" u 1:2:3 via a,b
```

Nemlineáris görbéket is illeszthetünk, akár több paraméterrel is:

```
gnuplot> fit [0:4] a*exp(b*x)+c "meres.dat" u 1:2:3 via a,b,c
```

Illesztés után az illesztett görbét természetesen rá lehet plottolni az adatsorra:

```
gnuplot> p "meres.dat" u 1:2:3 w ye, a*exp(b*x)+c
```

Az illesztés iterációs lépéseit és a végeredményeket a program egy `fit.log` nevű fájlba menti. Ha több illesztést is végzünk, akkor az újabb eredmények a fájl végéhez hozzáíródnak (append).