

GnuPlot – Adatok ábrázolása mesteri fokon (1. rész)

Tudományos munkánál, vagy akár egyéb adatok (például hálózati forgalom, processzor terheltség, stb.) feldolgozásánál sokszor többet mond egy jól elkészített grafikon, mint több száz (ezer, vagy még több) sornyi adat. Ezen grafikonok elkészítésében nyújt segítséget a gnuplot. És még sok másban is.

Bevezetés

A *gnuplot* parancssoros eszköz, interaktívan várja a beállításokat és utasításokat, de lehetőségünk van a parancsokat fájlba írni, amit a *gnuplot* (mint egy értelmező) végrehajt. A modernebb, kényelmesebb megoldások kedvelői rengeteg grafikus felületet (GUI) találhatnak az interneten (talán a legjobb a *Qgfe* → <http://www.xmlmath.net/qgfe/index.html>).

A program forrását a → <http://www.gnuplot.info> címről tölthetjük le, valamint a legtöbb disztribúció csomagként is tartalmazza.

Első lépések

A *gnuplot* parancs (opciók nélküli) kiadása után egy promptot (*gnuplot*>) kapunk. Kilépni az *exit* ill. a *quit* parancssal lehet. A program súgója nagyon jól meg van oldva, fastruktúra-szerűen elrendezve (topikok, altopikok, ...). A súgót a *help* parancs kiadásával érhetjük el. Az általános információk után az altopikok szerepelnek, amik közül választhatunk. Ezután további altopikok, stb. Egy altopikba közvetlenül is beléphetünk, a *help* parancsot paraméterezve. Például a

```
help commands plot
```

parancs a *commands* „főtema” *plot* altémáját „nyitja meg”.

1. táblázat *A GnuPlot leggyakrabban használt parancsai*

<i>size</i>	méret
<i>title</i>	ábra felirata
[<i>xyz</i>] <i>label</i>	tengelyek felirata
<i>datafile</i>	bemeneti adatfájl
[<i>xyz</i>] <i>range</i>	a tengelyeken ábrázolt tartomány
<i>style</i>	ábrázolás módja, például vonalak, pontok, stb.
<i>border</i>	keret

A *gnuplot*-tal az ábrákat többféleképpen kezelhetjük: egy ablakban nézegethetjük, vagy fájlba is menthetjük. Nagyon sok formátumot ismer: *png*, *LaTeX*, *fig*, *svg*, *pdf*, *postscript*, *gif*, hogy csak néhányat említsünk. A kimenetet a *set terminal* parancssal szabályozhatjuk. Például a

```
set terminal png
```

parancs kimenetként *png* formátumú képfájlt állít be. A fájl nevét a *set output "kep fajl neve"*-vel állíthatjuk be (a fájlnev idézőjelek között legyen!). Az ablakban történő megjelenítést a *terminal x11*-re állításával érhetjük el (egyébként ez az alapértelmezett).

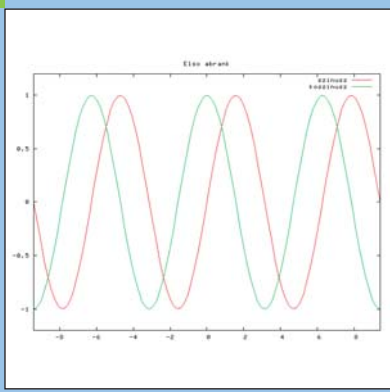
Ábrázoljunk!

Miután beállítottunk egy kimenetet, ideje, hogy megtöltsük tartalommal! Először tekintsük át az általánossá-

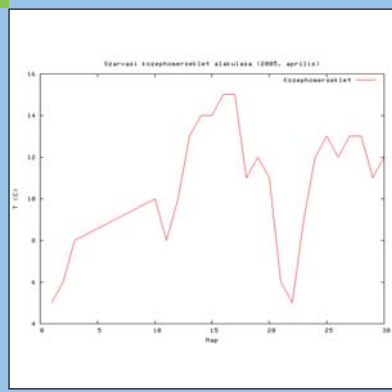
gokat! Egy ábrának a konkrét tartalomtól eltekintve sok jellemzője van, például felirat, tengelyek feliratai, méret (pixelben), stb. Ezeket szintén a *set* parancssal állíthatjuk (talán a *bash* alatti környezeti változóikhoz hasonlíthatóak). A beállított értékeket a *show* parancssal ellenőrizhetjük. Némi (ezirányú) angoltudással és a *help set* segítségével általában könnyedén megtalálhatjuk a nekünk szükséges opciót. Néhány gyakran használt változót az 1. táblázatban foglaltam össze.

A *gnuplot* nem csak adatfájlokat képes feldolgozni, hanem konkrét függvényeket is ábrázolhatunk. Nézzünk egy konkrét példát:

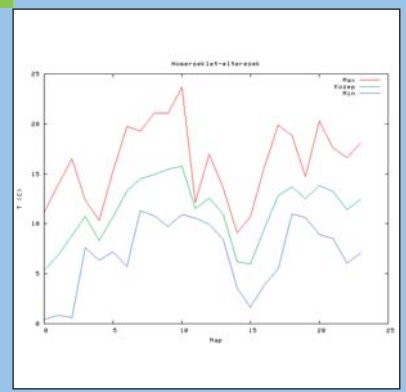
```
set title "Első abrakn"
set xrange [-3*pi:3*pi]
set yrange [-1.2:1.2]
plot sin(x) title
"szinusz", cos(x) title
→ "koszinusz"
```



1. ábra Első ábránk



2. ábra Áprilisi középhőmérséklet



3. ábra Áprilisi hőmérséklet-eltérések

A parancs-sorozat végeredménye az 1. ábrán látható (ne felejtjük el a kimenetet még a plot kiadása előtt átírányítani, ha az `x11` nem megfelelő!). A 2. és a 3. sorban látható a `range`-ek szintaxisa, míg a negyedik sorban a plot ábrázolja a két szögfüggvényt. Fontos észrevenni, hogy több grafikont egy ábrára úgy rakhatunk, hogy az ábrázolandókat vesszővel választjuk el, valamint egy másik fontos dolog, hogy az ábrázolandók megjelenési és egyéb tulajdonságait külön-külön is szabályozhatjuk, ezt a két `title` opció láthatjuk. Persze ezeket még csicsázhathatjuk, és bonyolultabb függvényeket is ábrázolhatunk.

Ez mind nagyon szép és nagyon jó, de ezt bármilyen függvényábrázoló programmal megtehettük volna. Mégis miben nyújt többet a `gnuplot`?

Adathalmazok ábrázolása

Nagyon sok és érdekes adat található különféle meteorológiai adatbázisokban, amelyek kiáltanak azért, hogy ábrázoljuk őket. Egy ilyen adatsort a <http://www.mvk.tsf.hu/Meteorologia/met.htm> címen is találhatunk, havi csoportosításban, azon belül pedig napi lebontásban (Prjevara Zoltán és Tanszékének munkája). A lementett `HTML`-fájlokból az adatokat a következő paranccsal kinyerhetjük, amelyet már feldolgozhatunk a `gnuplot`-tal:

```
w3m -F 2005-apr.htm | grep
  -E "^[1-3]" | sed "s@\\ \\ *@\\
  @g" | sed "s@, @\\. @g" >
  2005-apr.txt
```

A `w3m` egyszerű szöveges kimenetet állít elő, a `grep` csak azokat a sorokat

listázza, amelyek 1-gyel, kettővel vagy hárommal kezdődnek, az első sed az egymás utáni szóközöket egy szóközzel helyettesíti, a második sed pedig a (tizedes)vesszőket cseréli (tizedes)pontokra. Ezután a következő parancssorozattal a 2. ábrán látható grafikon állítható elő:

```
set title "Szarvasi
kozephomerseklet alakulasa
(2005. aprilis)"
set xlabel "Nap"
set ylabel "T (C)"
plot "2005-apr.txt" with lines
  title "kozephomerseklet"
```

Az első három sornak az eddigiek alapján érthetőnek kell lennie. A negyedik sor se szorul sok magyarázatra: ezúttal nem egy függvényből vesszük az adatokat, hanem egy fájlból, ez a "2005-apr.txt" rész. A `with` kulcsszó állítja be a jellemzőket: a `lines` résszel azt kérjük, hogy a diszkrét pontokat kösse össze vonallal, a `title` pedig a jelmagyarázat szövegét állítja be.

Jogosan merül fel az a kérdés, hogy a napi maximum- és minimum-hőmérsékletek mennyire térnek el a napi középhőmérsékletektől. Nyilván végig lehetne böngészni és kiszámolgatni az eltéréseket, de sokkal szemléletesebb, ha szintén egy grafikonon ábrázoljuk a három mennyiséget.

Azt tudjuk már, hogy hogyan lehet egyszerre több grafikont ábrázolni egyszerre (vesszővel elválasztjuk), de hogyan közöljük azt, hogy az adatfájlból melyik oszloppal kívánunk dolgozni?

```
set title "Homerseklet-
  elteresek"
set xlabel "Nap"
set ylabel "T (C)"
plot "2005-apr.txt" using
  ($3) with lines title "Max" ,
  "2005-apr.txt" using ($2)
  with lines title "Kozep" ,
  "2005-apr.txt" using ($4)
  with lines title "Min"
```

A fentebbi kódsorban egyedüli ismeretlen kifejezés a `plot` sorában szereplő `using`. Ez utasítja a `gnuplot`-ot, hogy pl. a harmadik oszlopot „használja” (`using ($3)`). Nem ezen cikk feladata az adatok kiértékelése, de azért egy-két dolgot érdemes észrevenni: április 11-én nem volt nagy hőmérséklet-változás, viszont 10-én nagy volt az ingadozás.

Összefoglalás

Az eddigiek a `gnuplot` lehetőségeinek csak kis részét mutatták be, de már ebből is látható, mi mindenre használható. A következő részben a hőmérséklet-ingadozást még szemléletesebbé tesszük, valamint egyéb ábrázolási módok is előkerülnek.



Udvari Zsolt
(udvzsolt@gmail.com)

25 éves vagyok, egy gimnáziumban tanítok matematikát és fizikát.

A Linuxsal először 2004 elején találkoztam, az UHU Kamionja volt, ami elgázolt. Azóta 4-5 disztribúciót hosszabban is használtam, jelenleg egy saját építésű LFS-t nyúzok.