

## GnuPlot – Adatok ábrázolása mesteri fokon (2. rész)

Az első részben a GnuPlot alapjaival és alapparancsaival ismerkedhettünk meg. Az ottani információk alapján még nemigen tűnik ki a GnuPlot előnye a különféle táblázatkezelők (például az OpenOffice.org Calc része) diagram-készítőivel szemben.

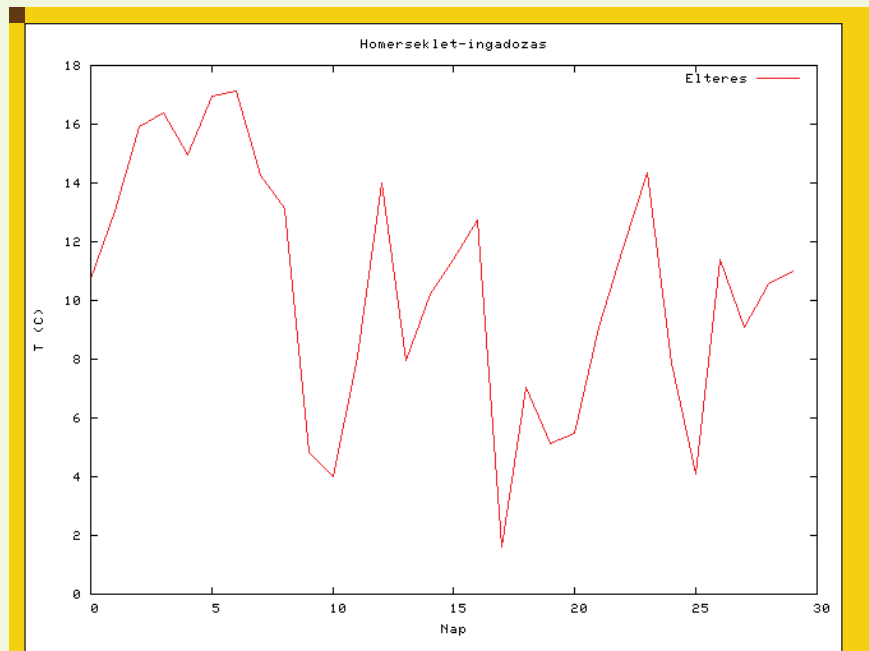
### Emlékeztető

Az első részben megismerkedtünk a *gnuplot*-tal, majd ennek öröme egy meteorológiai adatsort kezdtünk szemléletessé tenni. Az utolsó ábránk egy grafikon volt, amelynek vízszintes tengelyén 2005. áprilisának napjai voltak, míg a függőleges tengelyen hőmérsékleti adatok. Az ábrán három görbe szerepelt: a napi átlag-, minimum- és maximum-hőmérséklet. Itt már láthattuk a napi hőmérséklet-ingadozást, bár ez még nem volt az „igazi”, a lehető legpraktikusabb ábrázolási módja az ingadozásnak. Keresünk erre egy jobb megoldást!

### Műveletek az adatokkal

Először is gondoljuk meg, hogy mi jellemzi legjobban a napi hőingadozást! Rövidebb-hosszabb elmélkedés után rájövünk, hogy a napi legmagasabb és legalacsonyabb hőmérséklet közötti eltérés. Tehát akkor a maximum- és minimum-hőmérséklet különbségét kellene ábrázolni! Lehetséges-e az a *gnuplot*-ban, hogy két „oszlop” különbségét ábrázoljuk? Persze, és ráadásul nem is kell egy külön oszlopot létrehozni, amelyben a különbségek szerepelnek, hanem „dinamikusan” is lehet csinálni. Akkor indítsunk egy *gnuplot*-ot, majd az alábbi parancsokat adjuk meg neki:

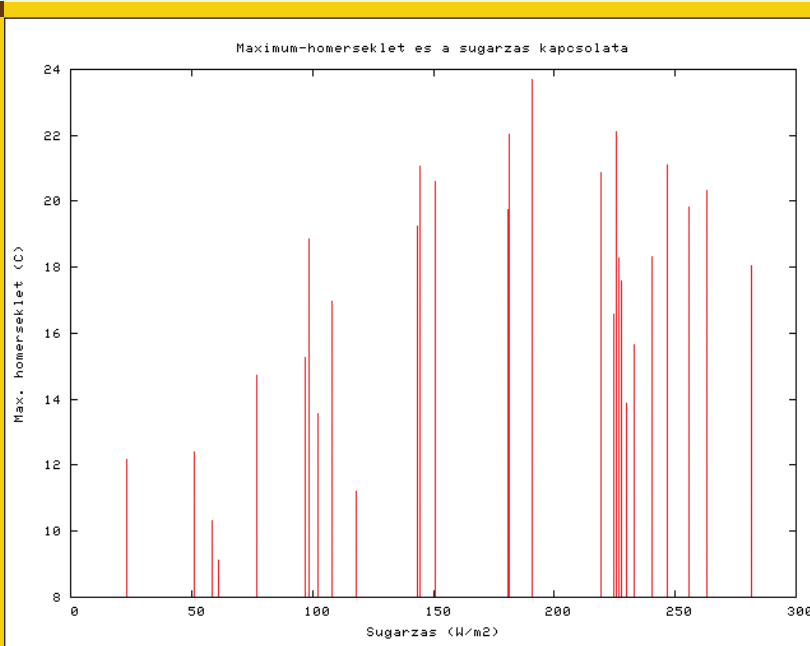
```
set title "Homerseklet-
ingadozas"
set xlabel "Nap"
set ylabel "T (C)"
plot "2005-apr.txt" using ($3)-
($4) with lines title "Elteres"
```



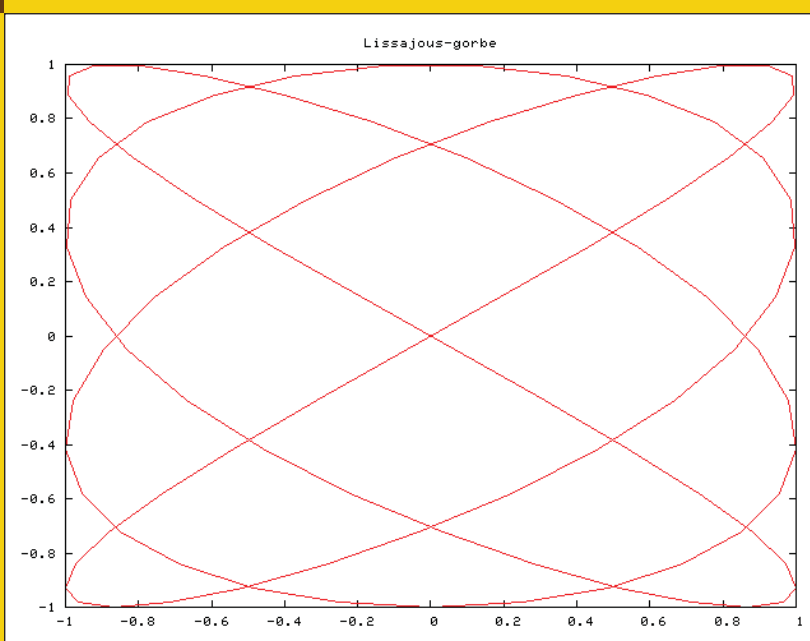
■ 1. ábra Hőmérséklet-ingadozás

A végeredmény az 1. ábrán látható. Az új dolog az eddigiekhez képest az utolsó sorban rejlik, mégpedig a (\$3)–(\$4)részben. Ez az, ami utasítja a programot arra, hogy a harmadik oszlop adataiból vonja ki az első oszlop adatait, és azt használja (*using*). Itt már tényleg nagyon jól látszódik az, amit már az előző részben még csak „szemre” állapítottunk meg: április 10. körül kicsi volt az ingadozás, míg a hónap elején viszonylag nagy. Hasonlóan használható a többi matematikai művelet, képlet, formula. Az alpműveletek jelei a „megszokottak”, a hatványozásé a kettő csillag (például a (\$3)\*\*2 a 3. oszlop

négyzetét jelenti), a trigonometrikus és egyéb operandusok szintén a szokásosak (*sin*, *cos*, *log*, ...), csak arra kell figyelni, hogy ezek egy zárójelen belül legyenek (pl. a *using ( sin(\$3) )* a harmadik oszlop szinuszt ábrázolja), különben a *gnuplot* nem fogja értelmezni. Olyan eset is előfordulhat, hogy nem az első oszlop függvényében akarom ábrázolni egy másik oszlop(ok) adatait, hanem pl. a második oszlop függvényében. Ez akkor fordulhat elő, ha a fájl készítője nem gondolt arra, hogy mi később a méréseit ábrázolni szeretnénk, vagy pedig egyszerűen egy más összefüggést is szeretnénk vizsgálni.



■ 2. ábra Maximum-hőmérséklet és a sugárzás kapcsolata



■ 3. ábra Lissajous-görbe

Ezen problémát megoldatjuk fájlmodosítással (például *awk*-val a második oszlopot első oszlopként kiírjuk), de minek ágyúval löni verébre, hiszen erre is van megoldás a *gnuplot* berkein belül.

Ha továbbra is a jól bevált adathalmazunkat használjuk, akkor arra is kíváncsiak lehetünk, hogy a napsugárzás és a maximum-hőmérséklet között van-e valami összefüggés. Nyilván azt sejtjük, hogy minél nagyobb az átlá-

gos sugárzás, annál nagyobb a maximum-hőmérséklet. Ahhoz, hogy ezt lássuk is, ábrázoljuk! Először gondoljuk meg azt, hogy ha az első oszlop lenne a napsugárzás és a második a maximum-hőmérséklet, akkor mit is tennénk? Nyilván ugyanazt, mint eddig is. Ha ezt megpróbáljuk (érdemes megnézni!), akkor nem egy szép görbét kapunk, hanem egy „összevisszaságot”, amelynek az az oka, hogy a program az első adatpárnak

megfeleltetett pontot köti össze (with lines) a második adatpár pontjával, a másodikat a harmadikkal, stb. Ha az *x*-tengely adatai nincsenek a fájlban nagyság szerint rendezve, akkor kapunk ilyen „hibát”. Persze ez sokszögek rajzolására nagyon jó. Visszatérve a problémánkra, a megoldást a következő *gnuplot*-parancssorozat adja:

```
set title "Maximum-homerseklet
↳ es a sugarzas kapcsolata"
set xlabel "Sugarzas (W/m2)"
set ylabel "Max. homerseklet
↳ (C)"
plot "2005-apr.txt" using
↳ ($6):($3) with impulses
↳ title ""
```

A 2. ábrán nagyjából igazolva is látjuk „elméletünket”, miszerint több napsugárzás nagyobb maximum-hőmérsékletet eredményez. Nyilván mivel az időjárás nagyon sok mindentől függ, így csak tendenciát állapíthatunk meg. Már látjuk is, hogy hogyan is kell megadni, hogy mi is a megoldás: a *using* után szereplő *(\$6):(\$3)* kifejezés. Tehát egy kettőspont és egy előtte álló *(\$6)* mondja meg, hogy mi a hatodik oszlop függvényében szeretnénk ábrázolni. Azt is észrevehetjük, hogy a *with lines* helyett *with impulses* szerepel, a már fentebb említett ok miatt, valamint a *title* után egy „üres sztring” következik, ami azt eredményezi, hogy nincs jelmagyarázat.

Az *x*-tengelyt megadó oszlopon is végezhetünk műveleteket, így készíthetünk pl. logaritmikusan ábrázolást is. Megjegyzem, erre van egy standard mód is, a *set logscale* parancs.

A következőekben elsősorban a matematikával és fizikával foglalkozók találhatnak hasznos információkat.

### Paraméteres görbék és felületek

Először is a paraméteres görbék ábrázolása: azon görbék, amelyek *x* és *y* koordinátája is egy-egy egyváltozós függvény (ilyen például a középszerűen egyenes paraméteres egyenlete), amelynek változóját általában *t*-vel jelöljük. A *gnuplot* is ezt használja (amiről tájékoztat is minket), viszont át is írhatjuk a *set dummy parameter*-neve paranccsal.

Paraméteres ábrázolásba a set parametric begépelése után léphetünk, kilépni pedig a set noparametric-kel lehet. Egy Lissajous-görbe (egymásra merőleges rezgések együttes „pályája”) előállítására:

```
set parametric
set title "Lissajous-gorbe"
plot [-pi:pi] sin(4*t),cos(3*t)
title ""
```

Egy másik érdekes terület a nem-Descartes-koordináták használata. Két dimenzióban (a síkon) a polárkoordinátarendszert szokták még használni (itt is kettő paraméter jellemzi a pont helyzetét: a pont és az origó távolsága valamint az őket összekötő szakasz és a vízszintes által bezárt szög).

A 4. ábrán látható virágot a következőképp lehet „legyártani”:

```
set polar
set title "Virag"
plot [-pi:pi] 1+cos(5*t)
title ""
```

Felületeket az splot paranccsal tudunk létrehozni, a változókat x-szel és y-nal jelöljük. A paraméterezése hasonló a plot parancséhoz, egy-két újdonsággal bővítve (lásd a help splot súgót). Az x11 terminálban még arra is van lehetőségünk, hogy az ábrát az egérrel forgassuk.

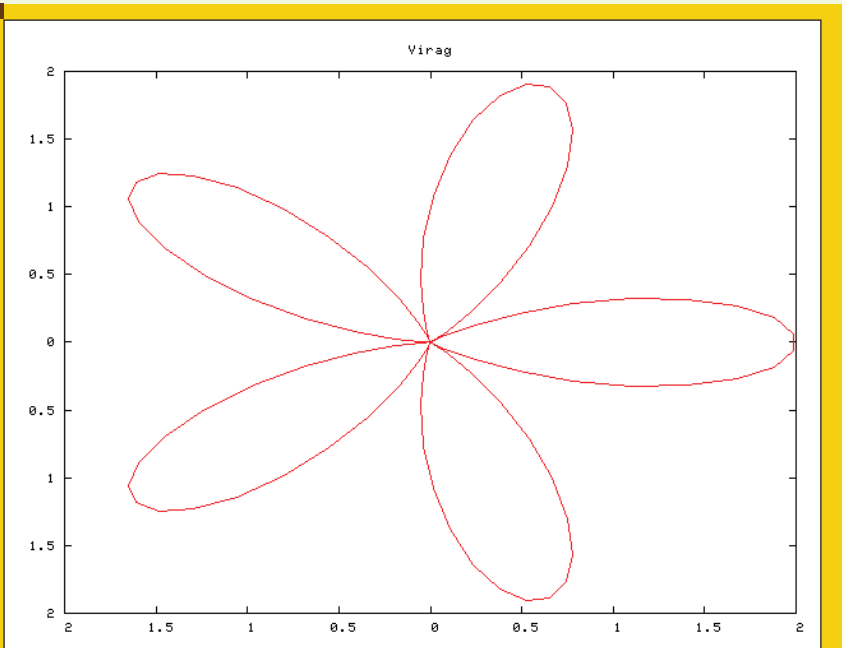
Szebb (szemléletesebb) képet ad, ha a pm3d környezetet állítjuk be:

```
set pm3d
set title "Paraboloid"
splot x**2+y**2 title ""
```

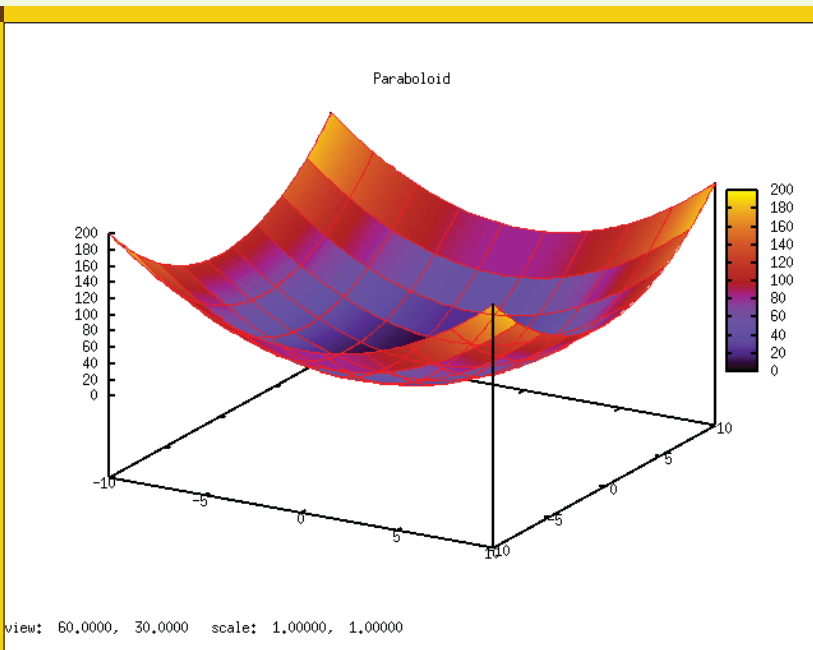
Térbeli polár- (spherical) és hengerkoordinátarendszer (cylindrical) a set mapping paranccsal állítható be. Ezek kezelése is a fentiekhez hasonló.

**Zárszó**

A gnuplot képességei a fentiekkel még egyáltalán nem merülnek ki, nagyon sok lehetősége van még benne. Például a vonalak színeit, típusát, az adattartományt, címkék, jelmagyarázatok helyzetét, ... állíthatjuk, sőt, animációk készítésére is alkalmas. Érdeemes a honlapján levő és a forráskóddal



4. ábra Virág polárkoordinátákkal



5. ábra Paraboloid

(vagy a csomagban) szállított demókat és példákat tanulmányozni. Összességében tehát egy nagyon jól használható, nagyon jól átgondolt eszköz kerül a kezünkbe a gnuplot személyében. A kiforrottságát és széleskörű használhatóságát mi sem jellemzi jobban, mint az, hogy rengeteg egyéb (grafikus és nem grafikus, új és kevésbé új) program használja, valamint a legújabb, 4.0-ás verziója 2004. április 16-án jelent meg.



**Udvari Zsolt**

(udvzsolt@gmail.com)

25 éves vagyok, egy gimnáziumban tanítok matematikát és fizikát.

A Linuxszal először 2004

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