

# Populációdinamika

Kovács Kristóf Péter

May 29, 2017



Kép: Fox and Hare,  
hannahaha (deviantart), 2011.

## A program

A program két állatcsoport, számának változásait mutatja be, egy ragadozókból álló, és annak zsákmányát. Nem differenciálegyenleteket használ, hanem minden állatot egyenként szimulál.

## Állatok szimulálása

Egy állat véletlenszerűen talál táplálékot, melyet egy faktor befolyásol (például a zsákmány száma). Ha nem talál semennyit, elpusztul. Ha egy bizonyos mennyiséghez hozzájut szaporodni fog.

Egy jóllakott hím és nőstény állat a szaporulatnak megfelelő utódot hoz létre. A nemeket az állatok véletlenszerűen kapják.

Az állatoknak van egy megadott élettartama is.

## Léptetés

Az állatokat két láncolt listában tároltam. Egy körben ezeken a listákon iterálnak végig ciklusok. A sorra kerülő állat élelmet keres, majd, ha eleget talált, egy szintén jóllakott párt.

A párkeresés a már korábban táplálkozott állat közül történik. A pázás egyel csökkenti a két példány éhségét, így nem kerülnek többször kiválasztásra.

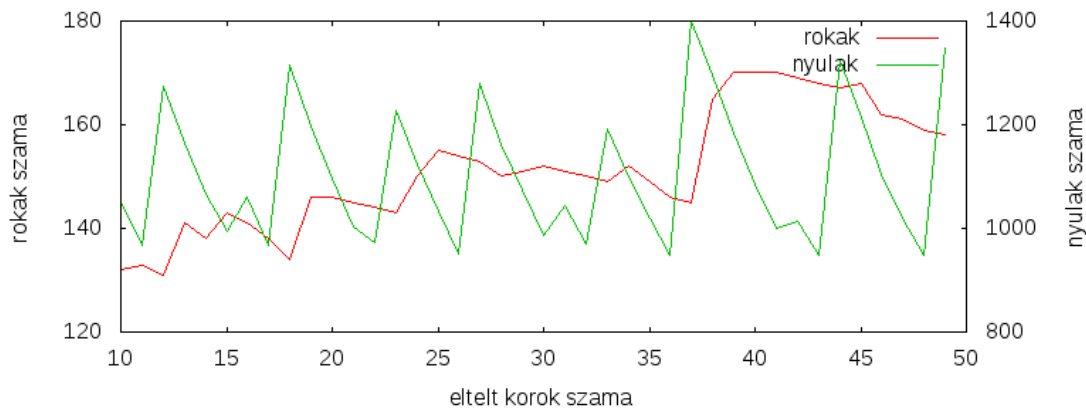
Egy kör lejátszása emiatt a módszer miatt magas példányszám mellett igen hosszúra nyúlhat.

## Eremények

A program nagyon érzékeny volt a kezdeti feltételekre. Sokszor előfordult, hogy pár kör alatt kihalt egyik vagy másik faj, vagy a számuk fokozatosan nőtt, amíg a program túlzottan le nem lassult.

## Kapott adatok

Az alábbi diagram egy fenntartható arányt mutat:



Kezdeti feltételek:

|             | róka | nyúl |
|-------------|------|------|
| példányszám | 10   | 100  |
| élettartam  | 7    | 3    |
| napi élelem | 2    | (1)  |

a rókák 11-szer annyi nyúl mellett vadásznak 100%-os sikerrel

Megfigyelhető, hogy a nyulak száma mindig 1000 köré esik vissza, ennek oka, hogy a rókák itt nem tudták szabályozni a nyulak számát, e miatt be kellett vezetni a nyulaknak is egy táplálékforrást, ami 1000 nyulat tud megfelelően eltartani. Az arány közel ugyanaz maradt, mint kezdetben, azaz 11-szeres.

Ezt azt jelzi, hogy mérvadóan csak a táplálékforrás szabályozta az populációt.

# A projekt

## Létrehozás

A projekt a **cmake** segítségével jön létre. A C++ és L<sup>A</sup>T<sub>E</sub>X fájlok fordítását, valamint az adatok Gnuplot-tal történő ábrázolását CMakeList.txt-k vezérlik.

A projekt legfelső szintjén csak az almappák vannak megadva. Az összetevők négy almappába vannak rendezve: kettő a programnak (futtatható és egyéb programrészek), egy a plottolásnak és egy az ezen dokumantum létrehozásához szükséges forrásoknak.

Három target szerepel a makefile-okban. Mindegyiknek feltétele az előtte levő megléte, tehát, ha üresen indítjuk el a dokumentumot, lefordulnak a forrásfájlok is, stb...

## szim\_futtatas

Gyakorlatilag a program lefordítása és futtatása. A program által kiadott adatsortól függ, amely annak a **custom\_command**-nak a kimenete, amely a programrészek lefordítását és összelinkelését vezérlő alprojektet indítja el.

## plottolas

A kiadott adatsort ábrázolja a Gnuplot programmal, az elkészült png fájl-tól függ. Az ezt létrehozó custom\_command a konzolon is beírható

```
> gnuplot script.p
```

parancsot hajtja végre. Szükséges hozzá még a Gnuplot package.

## dokumentum

A végső PDF létrehozásáért felel. Egy kiegészítő cmake fájlt használ fel, amit a mappája tartalmaz is. A latex fordító igen érzékeny a képek helyének hivatkozásaira, ezért a használt **UseL<sup>A</sup>T<sub>E</sub>X.cmake** minden képet lemásol és megfelelő útvonalat alkotó mappákba rakja őket. A még nem plottolt ábrákkal ezt nem tudná megtenni, ezért helyettük felülírható cserefájlokat generál.

## Megosztás

A programot és annak változásait a saját gépemen a **git** segítségével mentettem el a *git add* . és *git commit* parancsokkal a helyi repository-ba.

A kész mentéseket a *git push* paranccsal töltöttem fel a korábban elkészített online repository-ba, előtte beállítva annak url-jét a *git remote add* paranccsal.