

ZSÁKMÁNSZERZÉSI MODELL

Sztlávi Péter

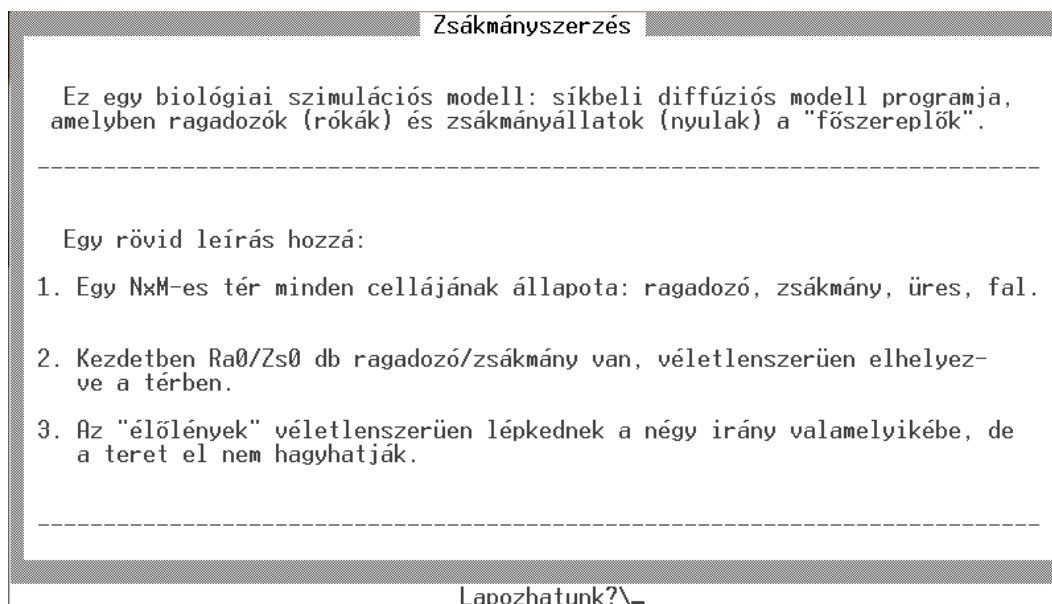
2000-2012

TARTALOM

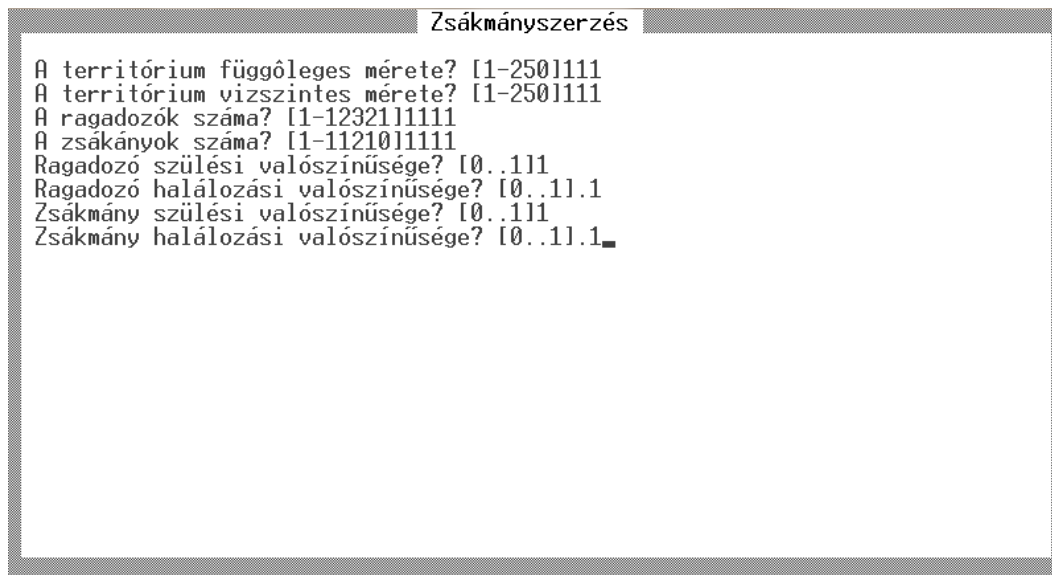
1. Elvárások	2
2. A problémák.....	3
2.1. Szimulációs keret.....	3
2.2. Szimulációs lépés – a lényeg.....	4
2.2.1. A szimulációhoz tartozó reprezentáció.....	4
2.2.2. A szimulációhoz tartozó implementáció	4
3. A keretprogram.....	6
4. Feladatok.....	6
4.1. Alapfeladat.....	6
4.2. Modellezési feladatok	6
4.2.1. „Intelligencia”	6
4.2.2. „Memória” – egyed-paraméterezés.....	6
4.2.3. „Csoportszellem”	7
4.3. Szorgalmi feladat.....	7
5. A teljes anyag.....	8
1. Függelék – Néhány futáseredmény.....	9
2. Függelék – További megfontolások	16

1. Elvárások

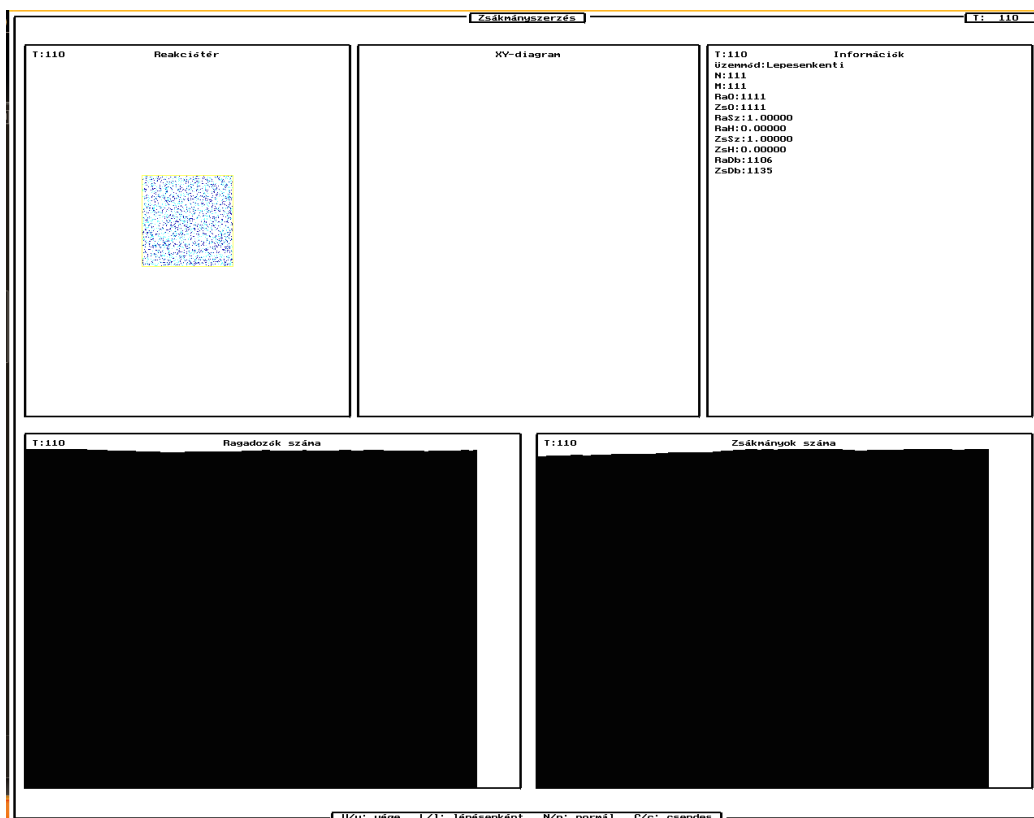
Az elvárásokat legjobban az alábbi, futás során keletkezett ábrásor fejezi ki.



1. ábra. Egy futási kép – Tájékoztató



2. ábra. Egy futási kép – Paraméterezés



3. ábra. Egy futási kép – Szimuláció közben (T=110)

Amint a futási képekből kitűnik: egy üres, XY-diagram nevű ablak is a felszínre került. Ez egy fejlesztési lehetőség, amely később szóba fog kerülni, mint megoldandó feladat.

Lássunk egy „in live” próbát: ZsakmanyMo.exe.

2. A problémák

Induljunk ki a Gáz-modellből a zsákmány-szerzési modell, és programja megalkotásához! Így számos reprezentációs és implementációs kérdést megoldottnak vehetünk. (L. [Gáz-modell!](#)) Algoritmikusan csak maga a szimuláció tisztázandó.

2.1. Szimulációs keret

A modellezés alapadatai:

- T – idő
- N, M – terület-méret
- $Ra0, Zs0$ – a kezdetben elhelyezendő kétféle állat száma
- $RaDb, ZsDb$ – a pillanatnyi darabszámok
- $RaSz, ZsSz$ – a szül(et)ési ráták, ennyi eséllyel szül pontosan egy utódot a kiválasztott egyed
- RaH, ZsH – a halálozási ráták, ennyi eséllyel múlik a kiválasztott egyed

A nagyvonalú algoritmus (alig változik a korábbihoz képest):

```

üzemMod:=Lépesenkénti
Tájékoztató
Inicializálás; T:=0
KezdőKépernyő
EredményMegjelenítés
Ciklus amíg nem VégeE
    T:=+1
    SzimulációsLépés
    EredményMegjelenítés
Ciklus vége
Zárás
ÖsszefoglalóMegjelenítés
    
```

A **Zárás** az Inicializálás párja. Feladata a futás „felhasználóbarát” lezárása. Ellenőrzi, hogy a felhasználó óhaja vagy valami automatizmus miatt ért-e véget a szimuláció. Ha az utóbbi miatt, akkor nyilván a felhasználónak fe kell hívni a figyelmét erre. Mivel ennél a modellenél az esetleges későbbi feldolgozás kedvéért naplózzuk a történéseket, ezért a zárás másik teendője a naplófájlba a végállapotot rögzíteni.

2.2. Szimulációs lépés – a lényeg

A véletlenszerűen kiválasztott egyeddel lezajlik egy teljes élet ciklus, amelybe beletartozik a kimúlás, vagy továbbélés; ez utóbbi esetében az esetleges szülés.

2.2.1. A szimulációhoz tartozó reprezentáció

Változó

```

[Ablakok:]
TérAblak: TAblak      [Territórium]
KarAblak: TKarAblak   [Karakteres információk]
Graf1,
Graf2   : THisztAblak [Hisztogramok]
XYDiag  : TXYAblak    [XY-diagram]
[Szimulációs adatok: paraméterek:]
T       : Egész      [Kellően nagy, mert hosszú futásra kell számítani]
N,M,
Ra0,Zs0,
RaDb,ZsDb: Egész
ReaTér  : TReaTér    [ideillőbb lenne persze a „Territórium” kifejezés]
RaSz,ZsSz,
RaH,ZsH : Valós
    
```

2.2.2. A szimulációhoz tartozó implementáció

Eljárás SzimulációsLépés:

Változó

```

i,j: Egész
[az aktív állat: (i,j)]
Ciklus
    i:=Random(N)+1; j:=Random(M)+1
amíg nem ReaTér(i,j) ∈ [Ragadozó,Zsákmány]
Ciklus vége
    
```

Zsákmányszerzési modell

Elágazás

ReaTér(i,j)=Ragadozó **esetén** RagadozóVáltozás(i,j)

ReaTér(i,j)=Zsákmány **esetén** ZsákmányVáltozás(i,j)

Elágazás vége

Mozgás(i,j)

Eljárás vége.

A későbbiekben többféle helykeresésre is szükség lesz. Ehhez be kell vezetnünk a helykeresés eredményét tároló adat típusát (l. keresés tétel):

Típus

TVanHol=**Rekord**(vanE:Logikai, si,sj:Egész [szomszéd i-, j-koordinátája])

Eljárás RagadozóVáltozás(**Konstans** i,j:Egész):

Változó

ker:TVanHol [a keresés tétel kimenete]

ker:=SzomszédKereses(Zsákmány,i,j)

Elágazás

nem ker.vanE **vagy** Random<RaH **esetén** [meghal]

ReaTér(i,j):=Üres

RaDb:-1

[ker.vanE **és** Random<RaSz **esetén** [zsákmányt eszik + szül]

ReaTér(ker.si,ker.sj):=Ragadozó

RaDb:+1

ZsDb:-1

egyéb esetben [zsákmányt eszik]

ReaTér(ker.si,ker.sj):=Üres

ZsDb:-1

Elágazás vége

Eljárás vége.

Eljárás ZsákmányVáltozás(**Konstans** i,j:Egész):

Változó

ker:TVanHol [a keresés tétel kimenete]

Ha Random<ZsH **akkor** [meghal]

ReaTér(i,j):=Üres

ZsDb:-1

különben [túlél]

ker:=SzomszédKeresés(Üres,i,j)

Ha ker.vanE **és** Random<ZsSz **akkor** [szül]

ReaTér(ker.si,ker.sj):=Zsákmány

ZsDb:+1

Elágazás vége

Elágazás vége

Eljárás vége.

Függvény SzomszédKeresés(**Konstans** mit:TCella; i,j:Egész):TVanHol

Konstans

MaxIsm:Egész(18) [a véletlen keresés maximális ismétlés-száma]

Változó

k,l,ism: Egész

vanE: Logikai

```

vanE:=Hamis; ism:=1
Ciklus amíg ism≤MaxIsm és nem vanE
    k:=i+Random(3)-1; l:=j+Random(3)-1
    vanE:=(i≠k vagy j≠l) és ReaTér(k,l)=mit
    ism:=+1
Ciklus vége
SzomszédKeresés:=TVanHol(vanE,k,l)
Függvény vége.

Eljárás Mozgás(Konstans i,j:Egész):
    Változó
        k,l: Egész

    Ciklus
        k:=i+Random(3)-1; l:=j+Random(3)-1
    amíg ReaTér(k,l)=Fal
    Ciklus vége
    Csere(ReaTér(i,j),ReaTér(k,l))
Eljárás vége.

```

3. A keretprogram

Az ablak-kezelés rutinjai ugyan nem változnak meg, de a szimulációs tér megjelenítése miatt van változás. Ezért új az Ablakok unit. L. [ZsakmanyKe.pas](#). [AblakokBio.pas](#).

4. Feladatok

4.1. Alapfeladat

A keretprogram hiányzó részeit pótolni, azaz '(*ide kell a kód*)' megjegyzéssel megjelölt részek kitöltése:

- SzimulaciosLepes-ben
- OsszefoglaloMegjelenites bővítése a KonkluzioFajl-lal összefüggésben

4.2. Modellezési feladatok

4.2.1. „Intelligencia”

Hogyan vihető némi „intelligencia” a modellbe, pontosabban az élőlények mozgási stratégiájába? Ötlet: a ragadozó már könyörtelenül elpusztít egy környezetében leledző zsákmányállatot, de a zsákmány oktanul akár a közelben lévő ragadozó karmába szalad, illetve nem törekszik a távolság növelésére. Ezt –alighanem– a SzomszédKeresés függvény bonyolításával lehetne megoldani.

4.2.2. „Memória” – egyed-paraméterezés

Az életszerűség növelése céljából értelmezzük át a egyes eseményeket. Például a ragadozó ne feltétlenül pusztuljon bele abba, ha nem talál maga körül élelmet. Egy paraméter írja le azt, hogy hány sikertelen koplalás után legyen vége. Ehhez azonban tudnunk kell a „próbálkozások” számát.

Az ötlet: a) egy a territóriummal (ReaTér-rel) „párhuzamos” sík létrehozása (paramétertér: ParTér(i,j) a ReaTér(i,j)-beli egyed paraméterei); b) a territórium minden egyede egy

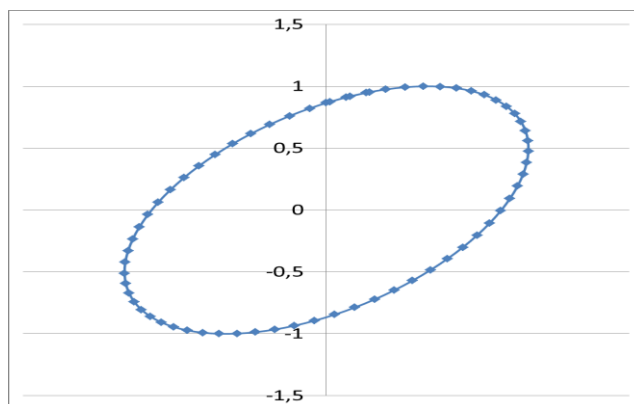
összetett típussal legyen leírható, amelyben helyet kapnak az egyed fajtája mellett a paraméterei: `TCella=Rekord(fajta:TFajta, éhség:Egész, ...)`. A szimuláció során, ha a ragadozó sikerrel járt a zsákmányolásban, akkor az éhség-paramétere (`ParTér(i,j)` vagy `ReaTér(i,j).éhség`) a beolvasott kiinduló értéket veszi föl. Amint kudarcos volt a zsákmányolás, e paramétere eggyel csökken. Ha a paramétere 0-ra csökken, akkor éhen hal. Mindkét esetben az Ablakok unithoz is nyúlni kell.

4.2.3. „Csoportszellem”

Hogyan vehető bele a modellbe az a fajta „társas viselkedés”, hogy ha sok azonos zsákmány egyed egymáshoz közel tartózkodik, akkor a ragadozókkal szemben némileg védve vannak?

4.3. Szorgalmi feladat

Az **XY-diagram** jól ismert a táblázatkezelők révén. Ebben két azonos hosszúságú számsorozatot egyetlen ábrában „újszerűen” vonnak össze. A két sorozat azonos indexű elemei egy síkbeli pont koordinátáiként kerül ki a képre. Nagy előnye ennek a diagramtípusnak, hogy ha a két sorozat véges értékhalmoz rendelkezik (ez a mi modellünkben igaz, hiszen a tér befogadóképessége határt szab az egyes fajtájú egyedeknek), akkor előre meghatározott síktartományban ábrázolható egy tetszőleges hosszúságú sorozat-pár.



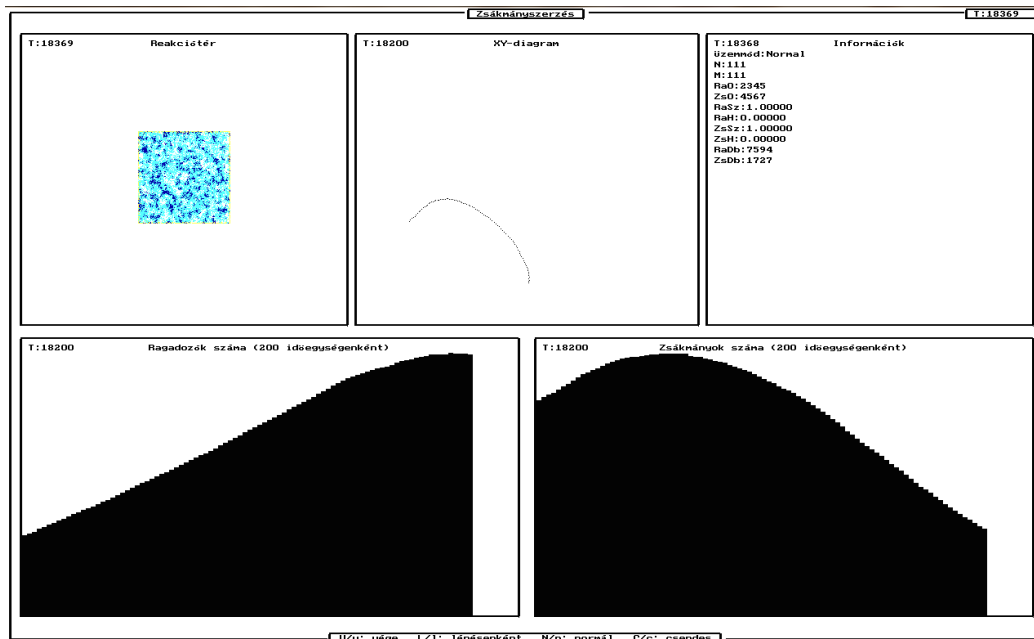
4. ábra. A $\sin(x)$ és a $\sin(x+\pi/3)$ ($x=0..6,5$; $0,1$ lépésközzel) két függvény által leírt egyetlen síkgörbe az XY-térben.

Mint látható nagyon látványosan és helytakarékosan mutat be egy ciklikus sorozatpárt.

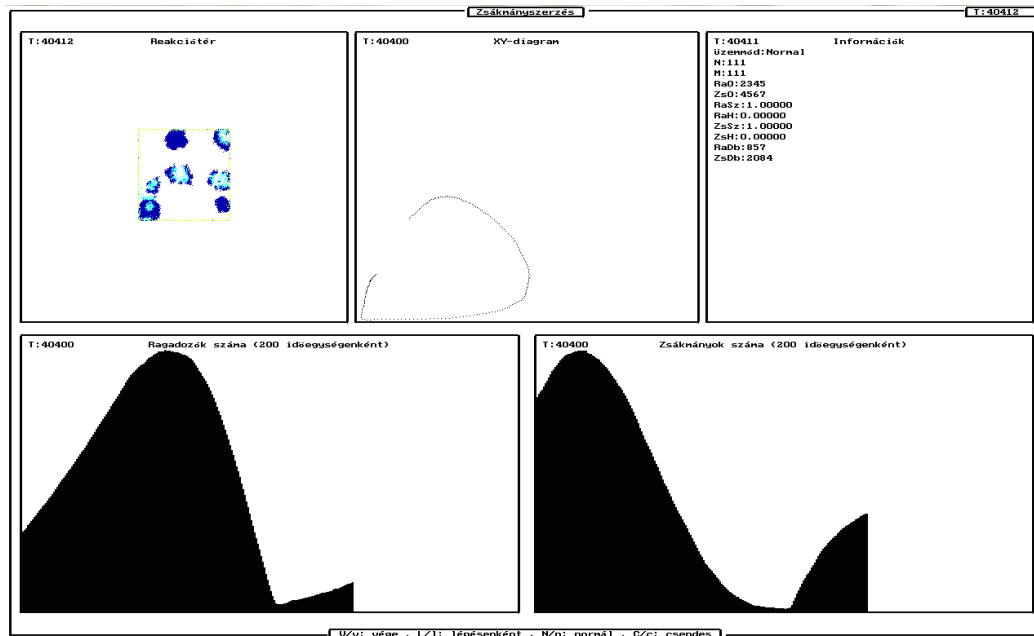
A vizsgálatok szerint a ragadozó-zsákmány modellek rendelkeznek (kvázi) ciklikussággal, ezért lehet hasznos egy ilyen ablakfajta.

Valósítsa meg az XY-diagram fajtájú ablakot, azaz bővítse a unitot ilyen típussal, és hangolja ehhez a fő programot is!

Zsákmányszerzési modell



**5. ábra. Egy futási kép – XY-diagrammal:
amikor a ragadozók fogyatkozni kezdenek.**



**6. ábra. Egy futási kép – XY-diagrammal:
a „ciklus” majdnem bezárul.**

5. A teljes anyag

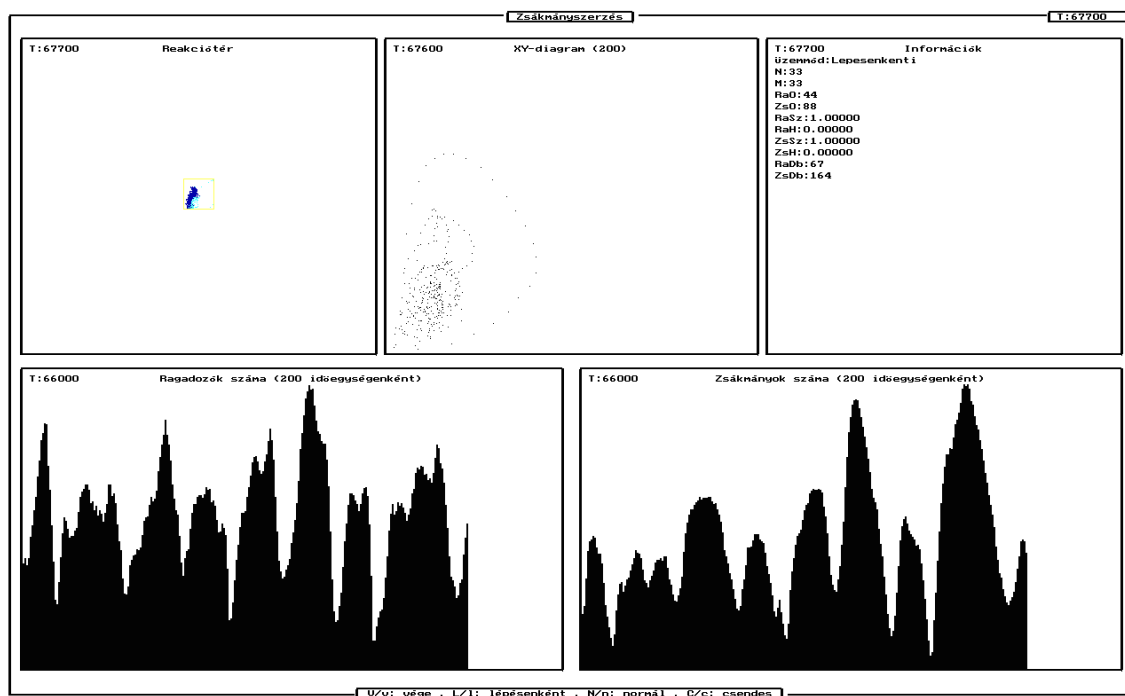
Ez a kézirat, források, lefordított kódok, súgófájl...: [ZsakmanyKe.zip](#).

1. Függelék – Néhány futáseredmény

Az alábbi ábrák alapján milyen következtetésekre lehet jutni? Illetve milyen paraméterezés mellett futtatná újra, hogy további következtetéseket vonhasson le?

A reprodukálhatóság kedvéért megjegyzem: Randomize nélküli futások eredményeit láthatjuk alább.

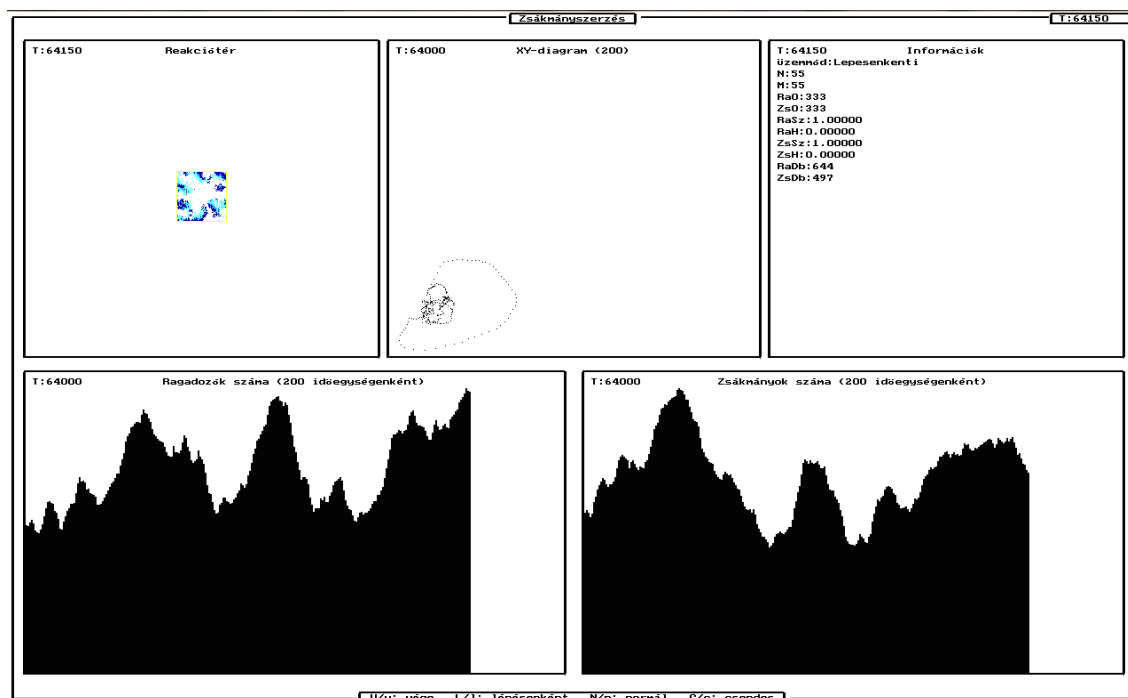
Tudjuk, hogy a kezdő állapotot egy véletlenszerűen generált állapot jelent. (Ez a következtetések egyike-másika szempontjából fontos.¹)



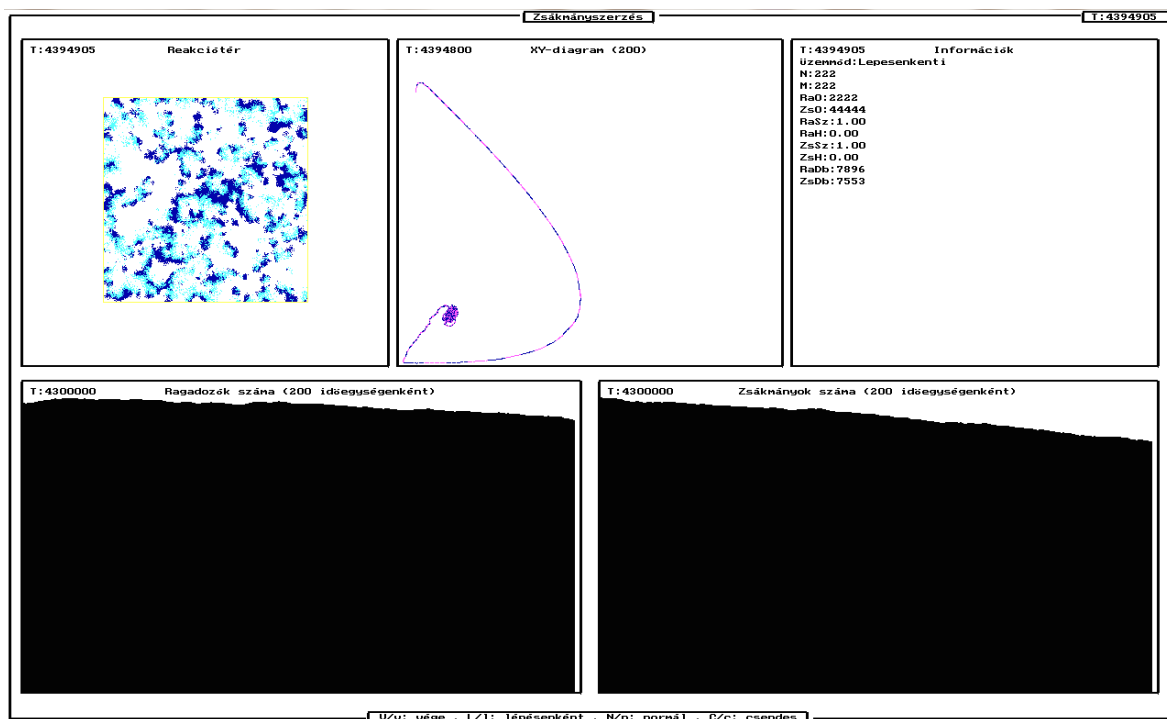
F1.1. ábra. Optimális „életesélyű” ragadozók és zsákmányok, extra kicsi méretű térben, T=67700 (4' futás!).

¹ A később az XY-diagramon megjelenő „szabályosnak” minősített görbe azt az állapotváltozást írja le, amely során a véletlenszerű feltöltöttségből egy „csapatokba rendezett” állapotba vezet. Ekkor a darabszámok nagy változásokat mutatnak!

Zsákmányszerzési modell

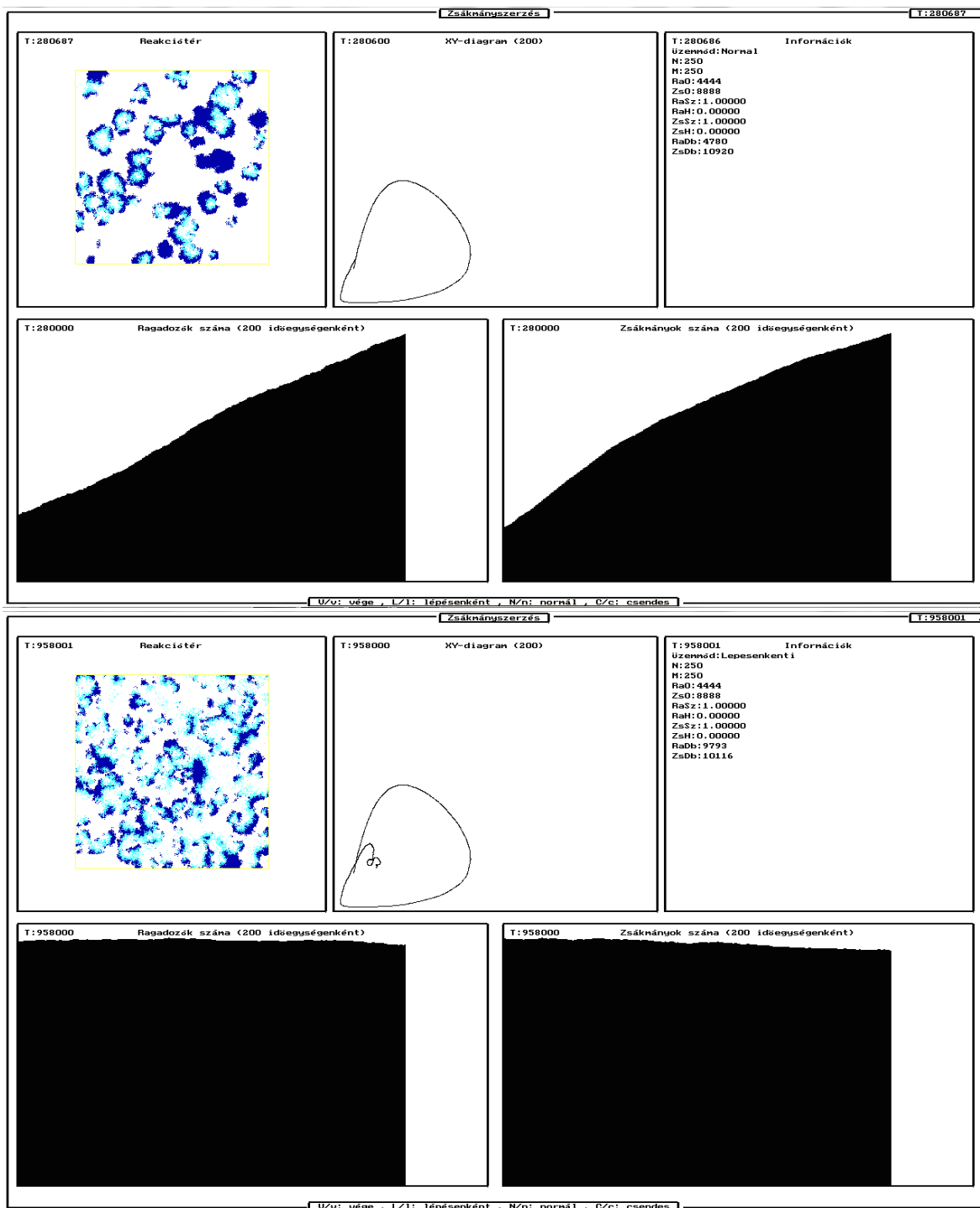


F1.2. ábra. Optimális „életesélyű” ragadozók és zsákmányok, kisméretű térben, T=64150 (4' futás!).



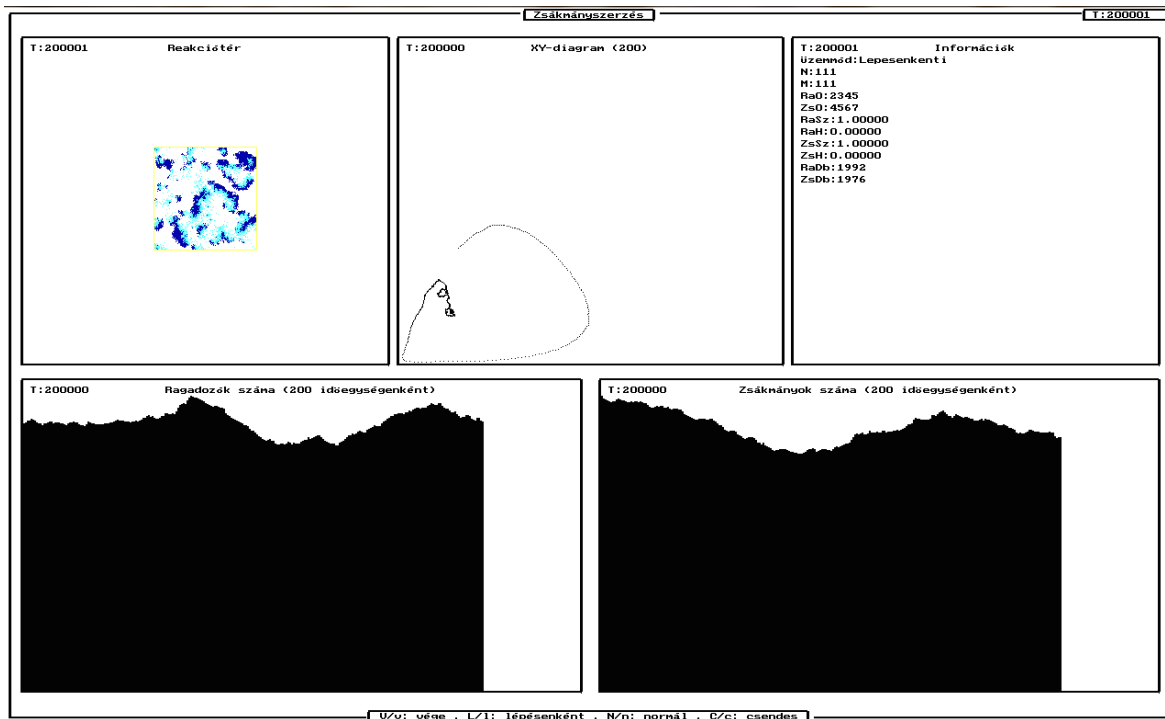
F1.3. ábra. Optimális „életesélyű” ragadozók és zsákmányok, közel maximális méretű térben, T=4 394 905 (110' futás).

Zsákmányszerzési modell



F1.4. ábra. Optimális „életesélyű” ragadozók és zsákmányok, nagyméretű térben, T=280 687 (felül); és a folytatása; T=958 001 (alul, 76’ futás).

Zsákmányszerzési modell

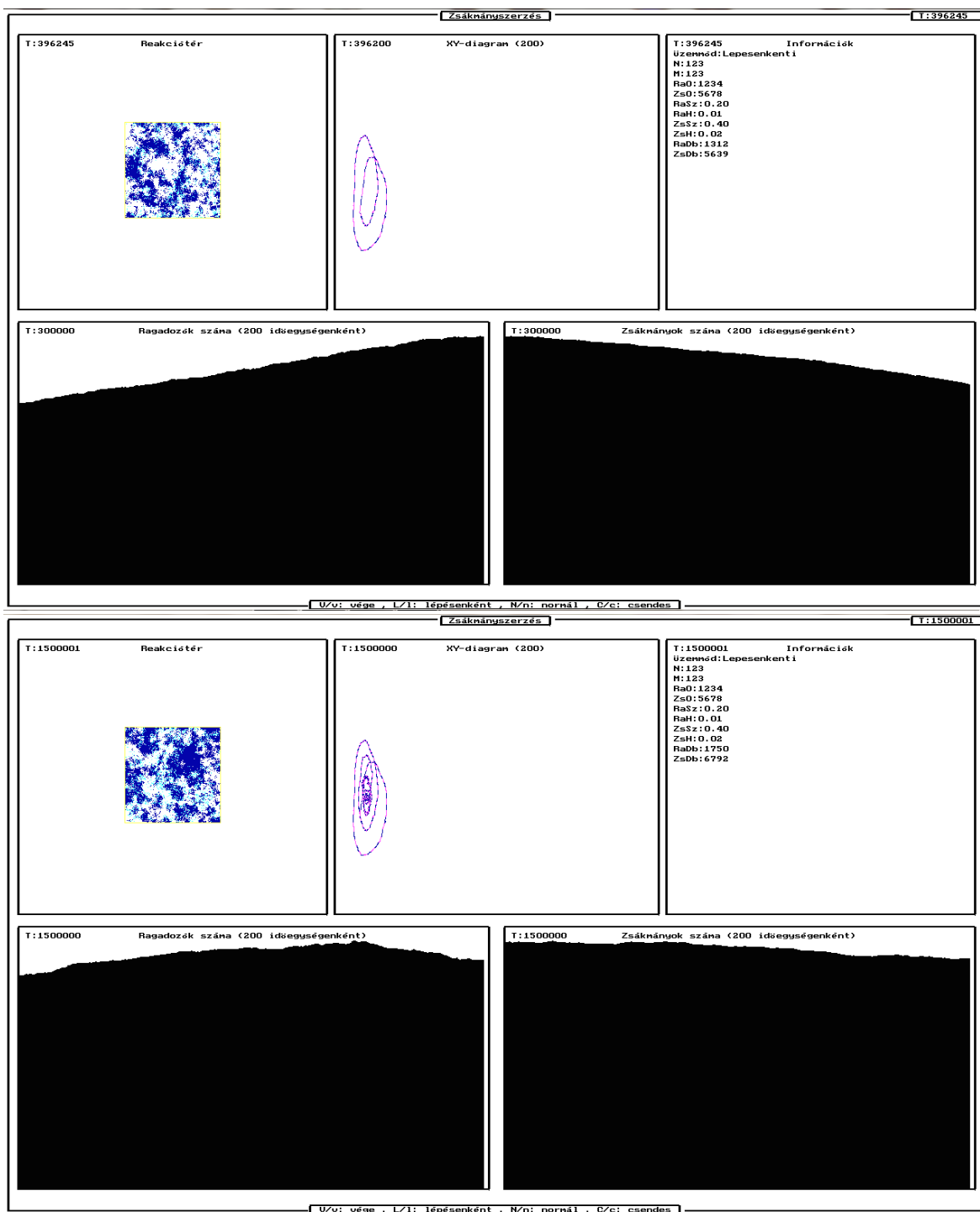


F1.5. ábra. Optimális „életesélyű” ragadozók és zsákmányok, közepes méretű térben, T=200 001 (12' futás).

Mire lehet következtetni: hogyan függ a szimuláció a tér méretparamétereitől?

Miért bomlik az XY-diagram jól érzékelhetően egy *szabályosan* változó és egy *szabálytalanul* kunkorodó szakaszra? Annyit tudni kell és lehet, hogy a szabályossal kezdődik.

Zsákmányszerzési modell

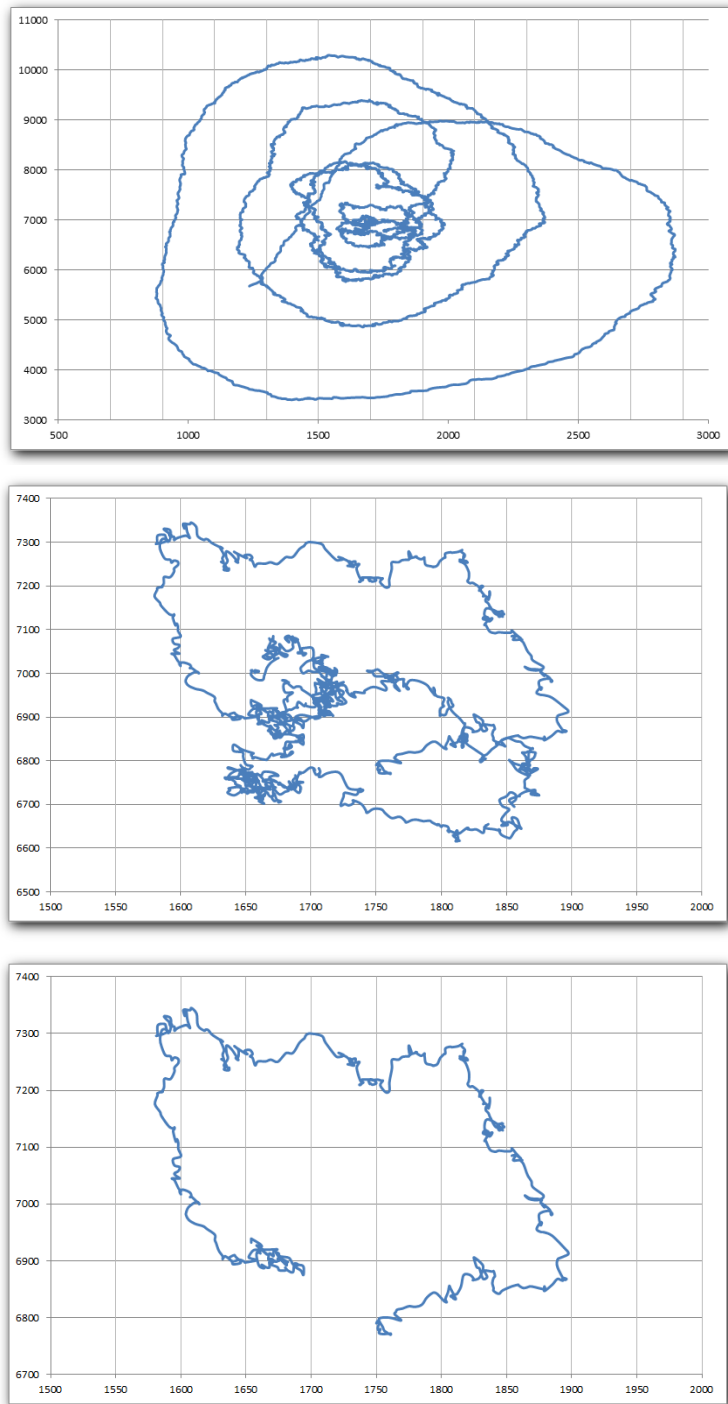


F1.6. ábra. A valóságoshoz jobban hasonlító paraméterekkel, közepes méretű térben, T=396 245 (felső kép); egy T=1 500 001-kori állapot (alsó kép; 37').

Már a korábbiaknál is megfogalmazódhatott az a vágy, hogy jó lenne az XY-diagram egyes részleteit finomabban megvizsgálni. Ne feledjük, hogy 1) a naplófájl relatíve kis időegységenként (most éppen 200) minden fontos állapot-paramétert megőrzött; 2) mégpedig a táblázatkezelők körében általánosan ismert fájl-formátumban (csv), ami bármikor újra felhasználható! Ha tehát egy táblázatkezelőbe beolvassuk, akkor abban további vizsgálatokat végezhetünk. Például az

Zsákmányszerzési modell

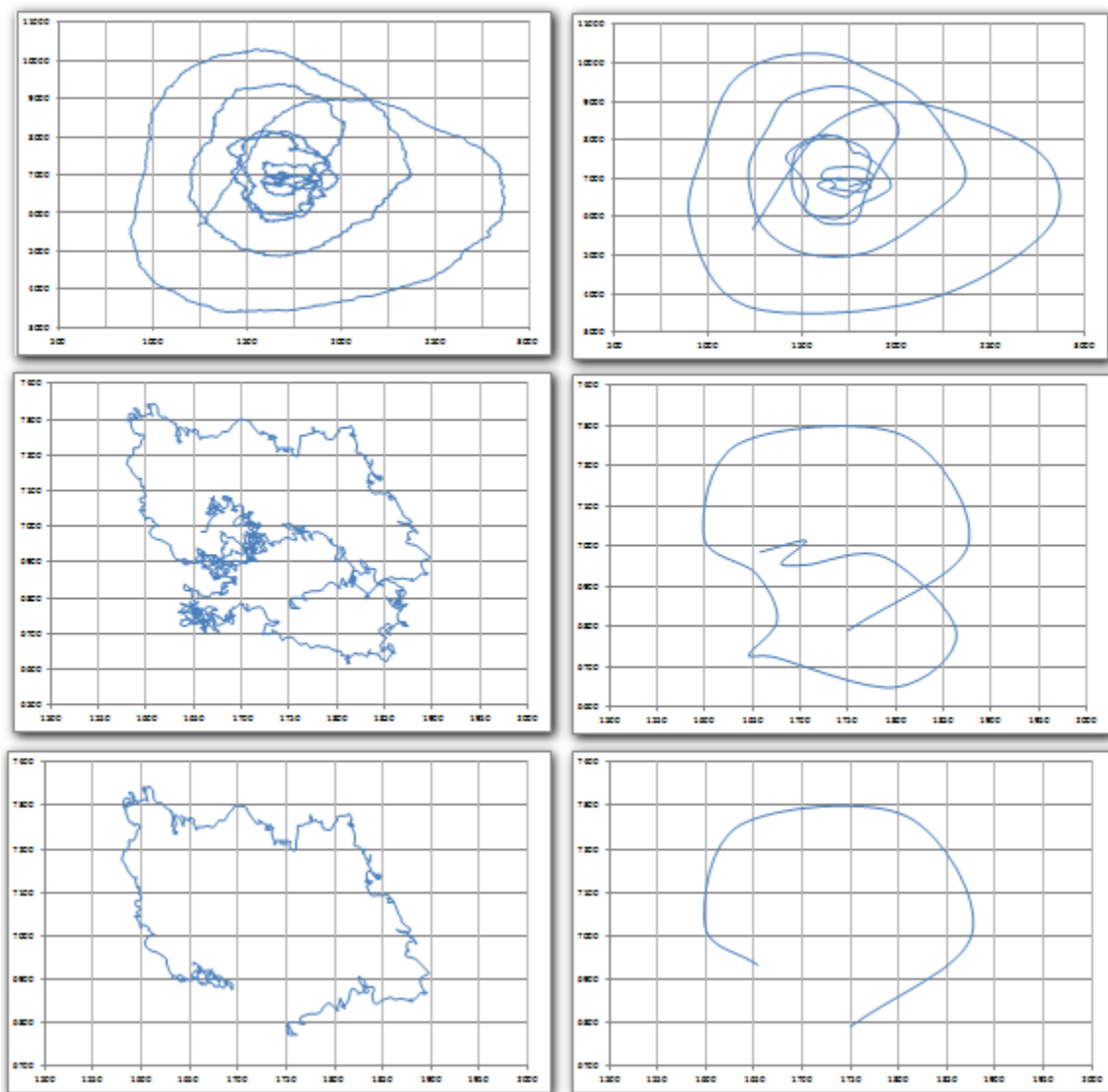
XY-diagramnak egy-egy érdekesebbnek vélt, vagy nem kellően kivehető részletét kiragadhatjuk, újrarajzoltathatjuk. „Ciklusokat” kereshetünk...



F1.7. ábra. Az F6. ábrán látható futás eredményét Excelbe bevíve, alkalmas beállításokkal a részletek jobban kivehetőkké tehetők. (felül) A középső kép az utolsó 300 000, az alsó az utolsó 100 000 időegységet emeli ki.

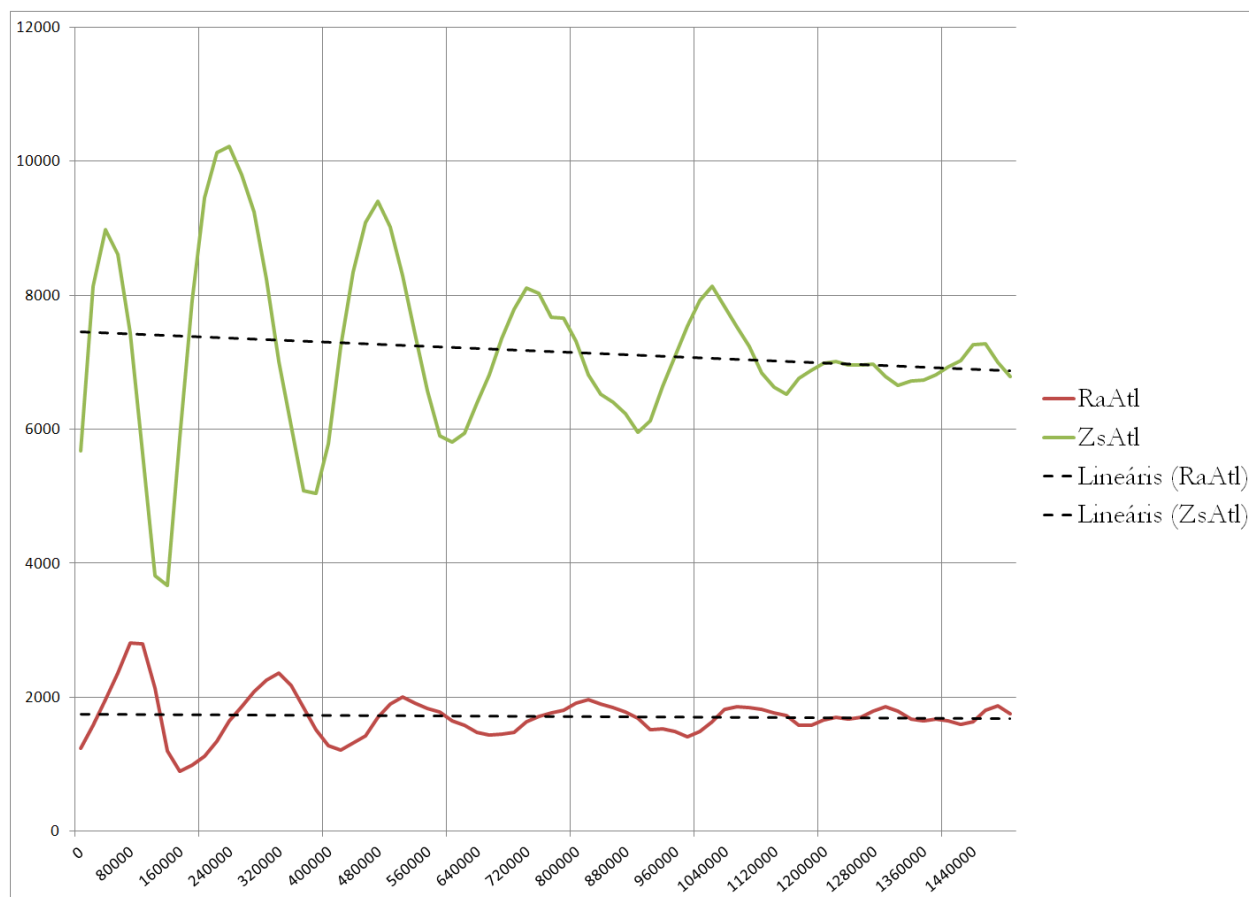
A ritkítás a lehető legmechanikusabb módja a „trend” előállításának. Ezt mutatja az alábbi ábra.

Zsákmányszerzési modell



F1.8. ábra. Ritkítással drasztikusan egyszerűsödnek a fenti ábrák: balra az eredeti, jobbra a 100-adára ritkított.

A „hagyományos” vonal-diagrammal a létszámok alakulás látható a következő ábrán. Ebből (is) látszik, hogy az idő múlásával csillapodik mind a ragadozók, mind a zsákmányok száma. Némi fázis eltolódás végig látható.



F1.9. ábra. A teljes szimuláció létszámának alakulása az idő függvényében. Ritkított adatok alapján. Piros a ragadozókhoz, zöld a zsákmányokhoz tartozik.

Alkosson további lényegkiemelő módszereket a táblázatkezelő képességeire építve!

2. Függelék – További megfontolások

Figyelmünket összpontosítsuk az elméletre! Az általunk vizsgált probléma legegyszerűbb –és legkorábbi– matematikai modelljére, a jól ismert Lotka-Volterra modellre. Ez az alábbi differenciál-egyenletrendszerre épül:

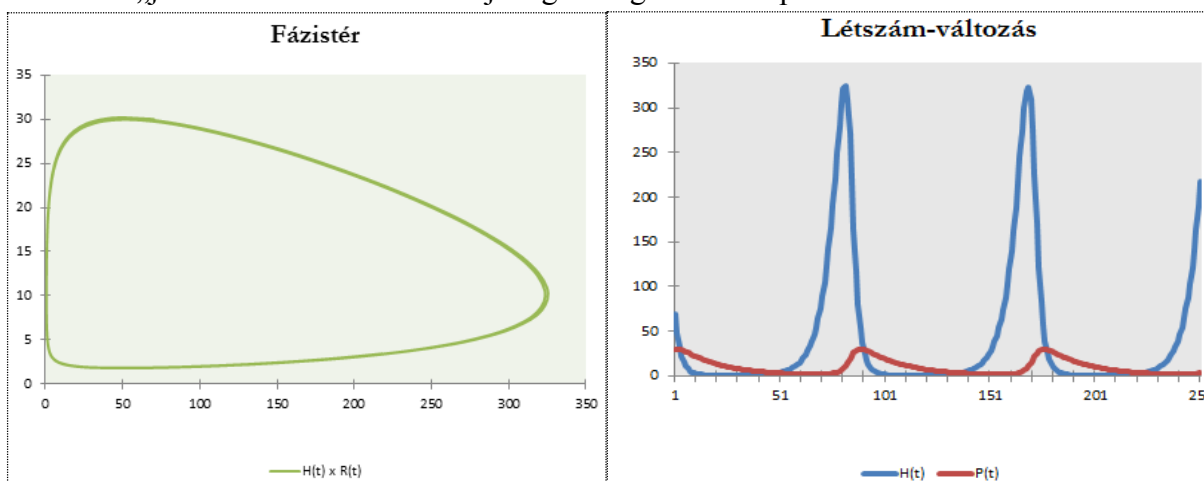
$$\frac{dH}{dt} = rH - aPH \quad (1)$$

$$\frac{dP}{dt} = bPH - mP \quad (2)$$

Átvettük most Alexei Sharov „On Line Lectures” c. anyagának ² 10.2. fejezetének jelöléseit. Az egyenletből a következők látszanak: H és P a zsákmány, ill. a ragadozók száma; r és b a zsákmány, ill. ragadozók születési „hajlandósága”; a és m pedig a zsákmány, ill. a ragadozók halálási esélye. Az egyenlet „működése” így interpretálható: A zsákmányok számának változása csak saját létszámukból adódó (halálózson felüli) születéstől függ pozitíve ($r \sim$ születési ráta és halálázási ráta különbsége), valamint a saját és a ragadozók számának szorzatától negatíve. A pozitív rész nyilvánvaló, a negatív azzal magyarázható, hogy nyilván annál nagyobb az esélye (így a várható értéke is) annak, hogy zsákmány áldozatul esik a ragadozónak, minél több van

² <http://home.comcast.net/~sharov/PopEcol/popecol.html>

belőle és a ragadozóból. A ragadozók növekedése annál nagyobb, minél nagyobb az esélye annak, hogy egy ragadozó zsákmányra bukkan, azaz ezek számának szorzatával arányos, a fogyást kizárólag a halálozás okozza, ami annál nagyobb, minél többen vannak. A modellel „játszva” a számalakulásra jellegzetes görbéket kapunk.



F2.1. ábra. A Lotka-Volterra modell megoldása, és a megoldásgörbék kirajzolása. A balra a „szokásos” XY-diagram, a jobbra az idődiagram.

Az egyenlet paramétereit változtatgatva gyakorta jutunk a fenti ábra által mutatott periodikus eredményhez.

Összevetve az eddigi eredményeinkkel elgondolkodtató különbséget fedezhetünk föl: a mi modelljeink kivétel nélkül kisebb-nagyobb szabálytalansággal egy egyensúlyi állapot felé konvergálnak. Mi lehet a magyarázat?

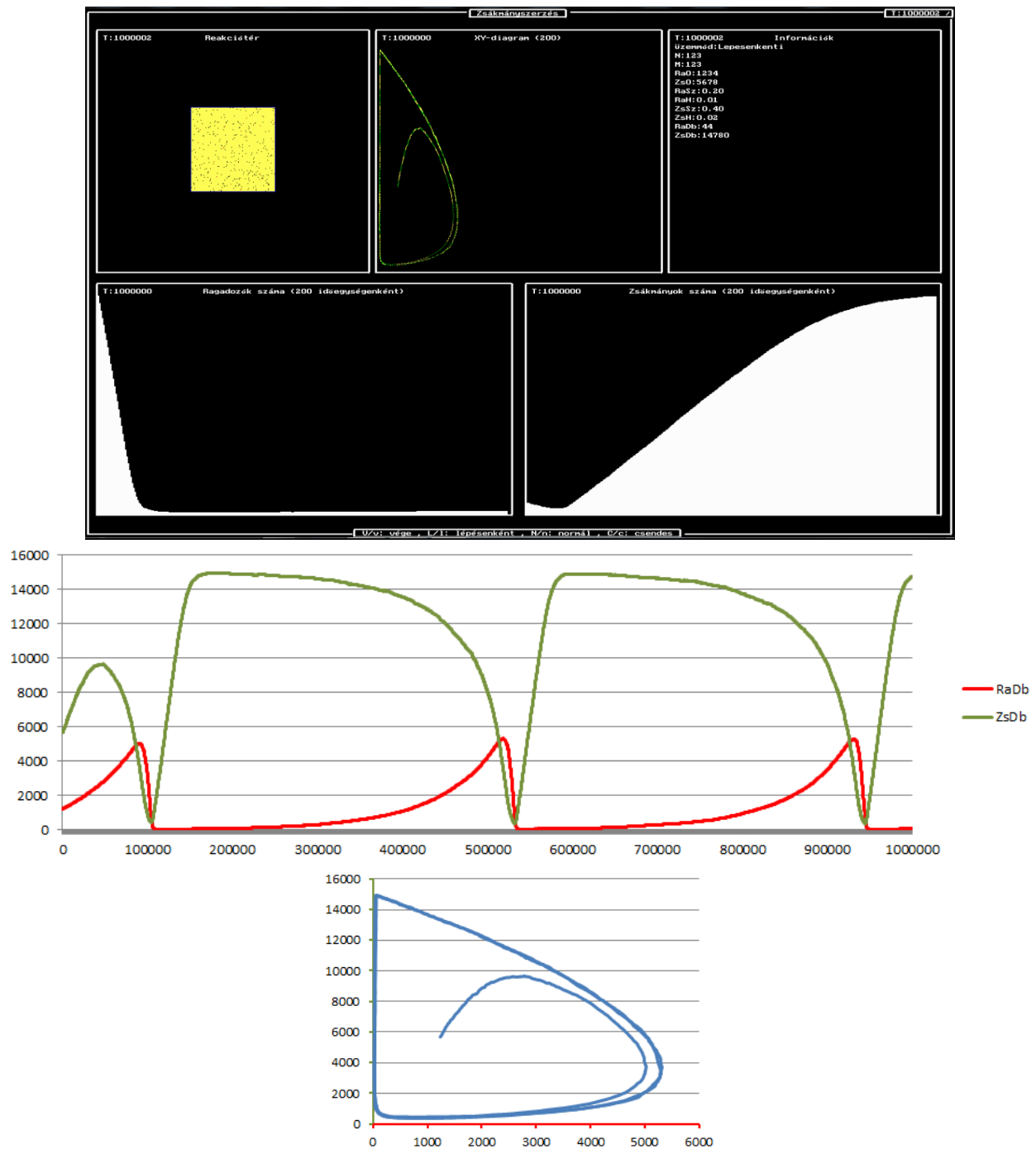
A kérdés annál is meglepőbb, mert a szimulációs részt megvizsgálva azt találjuk, hogy ugyanúgy kellene függnie a változásnak a pillanatnyi létszámtól, mint ahogy azt fentebb leírtuk. Az a megfigyelés vezet el a magyarázathoz, hogy a szimuláció legelején még egészen úgy változik az XY-diagram, ahogy az elméleti modellben látszik, csak később válik a görbe szűkülővé és egyre beljebb konvergálóbbá. Kérdés tehát, mi változik a számokon túl az idővel a szimulációs modellben?

Még egyszer elemezve a szimulációs lépést, rájövünk, hogy minden esemény, ami növekedést hoz, az annak az elemnek a közelében született egyedek által következik be, vagyis lokálisan érinti a következő „generációt”. Ugyanez a lokalitás a territórium mintázatán is kiválóan érzékelhető. Az a gondolat vetődhet föl bennünk, hogy az egyedek kezdetben még megvolt „egyenletességének” drasztikus elromlása áll a jelenség háttérében.

Az egyenletekre tekintve az érvelés jogosnak tűnik. A matematikai elemzés helyett most empirikusan ellenőrizzük sejtésünket! Ha igaz a sejtésünk, akkor csak a kiinduláskori egyenletességet kell fenntartanunk a szimuláció során. Erre több mód is kínálkozik: 1) gyakorta „keverjük össze” az egyedeket; 2) a keveredés esetleg nem ilyen „mesterkélt” is, valószerűbben is elérhető, ha az egyedek intelligensen a számukra kellemetlen szomszédától elfelé menekülnek, illetve a fennmaradás szempontjából fontos szomszédság felé mozdulnak, emellett kicsit megengedőbbek leszünk a ragadozókhoz azzal, hogy több sikertelen zsákmányolási kísérletet is megengedünk (újabb paraméter).

Az 1) elképzelés egy lehetséges megvalósításának eredményeit mutatja a következő ábra.

Zsákmányszerzési modell



F2.2. ábra. Egy közepes méretű territórium, többé-kevésbé reális paraméterek melletti futás végállapotú képe (fent); a teljes szimuláció létszámváltozása (középen), ugyanez XY-diagrammal (alul).

Megnyugtató a látvány, mely szerint a kezdeti „bizonytalankodás” után a szimuláció eredménye is közel periodikussá válik.