

SZÁMÍTÓGÉPES SZIMULÁCIÓK

# Populációdinamika

## JEGYZŐKÖNYV

Jakobi Ádám

2021.04.29.



# Tartalomjegyzék

<b>1. Bevezető</b>	<b>1</b>
<b>2. Elméleti bevezetés</b>	<b>1</b>
2.1. Egy szabadon fejlődő faj . . . . .	1
2.1.1. Logisztikus egyenlet . . . . .	2
2.1.2. Logisztikus egyenlet stabilitása . . . . .	2
2.2. Fajok versengése közös erőforrását . . . . .	3
2.2.1. Csatolt logisztikus egyenlet . . . . .	3
2.3. Ragadozó-préda rendszerek . . . . .	3
2.3.1. Lotka-Volterra-modell . . . . .	4
2.3.2. Realisztikusabb Lotka-Volterra-modell . . . . .	4
2.3.3. Tápláléklánc 3 fajra . . . . .	5
<b>3. Megoldás módszerei</b>	<b>5</b>
3.1. Differenciálegyenlet megoldó módszerek . . . . .	5
3.1.1. Euler algoritmus . . . . .	5
3.1.2. Negyedrendű Runge-Kutta módszer (RK4) . . . . .	6
3.1.3. Adaptív negyedrendű Runge-Kutta módszer (ARK4) . . . . .	6
<b>4. Kiértékelés</b>	<b>7</b>
4.1. Logisztikus egyenlet . . . . .	7
4.2. Versengés . . . . .	7
4.3. Lotka-Volterra-modell . . . . .	7
4.4. Realisztikus Lotka-Volterra-modell . . . . .	7
4.5. Tápláléklánc . . . . .	9
<b>5. Diskusszió</b>	<b>9</b>
<b>6. Függelék</b>	<b>10</b>
6.1. Logisztikus egyenlet . . . . .	10
6.1.1. Euler-módszer . . . . .	10
6.1.2. Adaptív negyedrendű Runge-Kutta módszer . . . . .	11
6.2. Versengés . . . . .	12
6.2.1. Kompetitív kizárás . . . . .	12
6.2.2. Két faj együttélése . . . . .	13
6.3. Lotka-Volterra-modell . . . . .	14
6.4. Realisztikus Lotka-Volterra-modell . . . . .	15
6.5. Tápláléklánc . . . . .	16
6.5.1. Populációk létszámai . . . . .	16

6.5.2. Populációk fázistere . . . . .	17
---------------------------------------	----

# 1. Bevezető

Ebben a jegyzőkönyvben a számítógépes szimulációk tárgy hatodik beadandóját, a populációdinamika témakörét dolgozom fel. Vizsgálom a logisztikus egyenletet Euler- és adaptív negyedrendű Runge-Kutta-módszerrel, ezeket összevetem a logisztikus egyenlet analitikus megoldásával. Ezt követően két, egymással versengő faj szaporodását szimulálom csatolt logisztikus modell segítségével, példát mutatok a kompetitív kizárás törvényére illetve megmutatom két faj együttélésének szükséges feltételeit. Implementálom a Lotka-Volterra-modellt és kiegészítem úgy, hogy figyelembe vegye a véges táplálék illetve telítődés hatását. Ezekről populáció létszám és fázistér ábrákat készítek. Végül létre fogok hozni egy 3 fajból álló tápláléklánc modellt, ennek szemléltetésére is készítek populáció létszám és fázistér ábrákat.

## 2. Elméleti bevezetés

Newton eredetileg a fizikai rendszerek megértésére vezette be a differenciálegyenleteket, azonban a differenciálegyenletek az élet számos más területén is hasznosnak bizonyultak, például a kémia, populáció biológia, pénzügyi-gazdasági és társadalmi folyamatok leírása során. Fontos megjegyezni, hogy ezek a modellek csupán a valóság közelítései, de fontos összefüggések fedezhetők fel általuk.

### 2.1. Egy szabadon fejlődő faj

A populáció létszámának ( $n$ ) változását vizsgáljuk az idő ( $t$ ) függvényében. Ha más tényező nincs, akkor logikus feltételezni, hogy a szaporodás és a természetes halálozás a populáció létszámával arányos. A szaporodási ráta ( $a$ ) írja le, hogy mennyi utód jön létre, a halálozási ráta ( $d$ ) pedig az elhunyt egyedek számát adja meg egy adott  $\Delta t$  időintervallumra (például egy évre).

$$n(t + \Delta t) = n(t) + an(t) - dn(t) = n(t) + rn(t),$$

ahol  $r$  ráta egyenlő  $a - d$ -vel. Ha  $\Delta t$  kicsi, akkor  $r$ -t átskálázva a fenti egyenlet határesetben ( $\Delta t \rightarrow 0$ ) a következő differenciálegyenletet adja:

$$\frac{dn}{dt} = rn.$$

A folytonos egyenlet jó közelítés, ha a populáció mérete kellően nagy és a szaporodási ráta elfolytonosítása nem okoz gondot. Az eredeti diszkrét megoldás a kiértékelt folytonos egyenlet újra diszkrétizálásával történik. A valóságban ezzel vigyázni kell az erős éves szaporodási ciklusok miatt. Az egyenlet megoldása a jól ismert exponenciális növekedés:

$$n(t) = e^{rt}.$$

Vegyük figyelembe, hogy az erőforrások korlátosak, vagyis a rendelkezésre álló táplálék véges:

$$\frac{dn}{dt} = rnF(n).$$

Tegyük fel, hogy az erőforrások egy maximum  $k$  létszámú populációt képes fenntartani (kapacitás). F legegyszerűbb modellje  $n$ -ben lineáris, kis populáció esetében elhanyagolható a hatása, a kapacitás elérésekor pedig nem enged további szaporodást.

Ezt a feltételt kielégíti:

$$F(n) = 1 - \frac{n}{k}.$$

Így a differenciálegyenletünk:

$$\frac{dn}{dt} = rn\left(1 - \frac{n}{k}\right).$$

### 2.1.1. Logisztikus egyenlet

Az úgynevezett logisztikus egyenletet  $x = \frac{n}{k}$ -val átskálázva kapjuk:

$$\frac{dx}{dt} = rx(1 - x).$$

Ennek megoldása  $r$ -től és az  $x_0$  kezdeti feltételtől függően növekedő vagy csökkenő szigmoid jellegű görbe lesz:

$$x(t) = \frac{1}{1 + \left(\frac{1}{x_0} - 1\right)e^{-rt}}.$$

### 2.1.2. Logisztikus egyenlet stabilitása

A logisztikus egyenlet egy nemlineáris differenciálegyenlet. A nemlineáris differenciálegyenletek nem minden esetben oldhatók meg analitikusan. Az egyenlet fixpontja  $\frac{dx}{dt} = 0$  helyen található (a mi esetünkben  $x = 0$  és  $x = 1$ -nél). A fixpontok lehetnek stabilak vagy instabilak, annak függvényében, hogy a fixpontból történő kitérítés során a rendszer visszatér-e a fixpontjába, ahonnan kitérítettük (ha visszatér, akkor stabil, ha nem, akkor instabil). A stabilitás legegyszerűbben lineáris perturbációszámítással vizsgálható. Legyen a differenciálegyenlet a következő alakú:

$$\frac{dx}{dt} = f(x),$$

és egy fix pontja  $x^*$  ( $f(x^*) = 0$ ). Kis perturbációval kimozdítva a rendszert a megoldás lineáris közelítésben kereshető:

$$\frac{dx}{dt} = x^* + \epsilon(t).$$

Beírva az eredeti egyenletre és Taylor-sorba fejtve:

$$\frac{d\epsilon}{dt} = f(x^* + \epsilon) = f(x^*) + \epsilon f'(x^*) + \dots,$$

a magasabb rendű deriváltakat elhagyva:

$$\frac{d\epsilon}{dt} = \epsilon f'(x^*),$$

amit megold:

$$\epsilon(t) = \epsilon(0)e^{f'(x^*)t}.$$

A megoldás akkor stabil, ha  $\lim_{t \rightarrow \infty} \epsilon(t) = 0$ , azaz  $f'(x^*) < 0$  (a mi esetünkben  $r > 0$  esetén  $x^* = 0$  instabil és  $x^* = 1$  stabil).

## 2.2. Fajok versengése közös erőforrásért

Ha egy élőhelyen (niche) több faj küzd egyazon táplálékért, a véges erőforrásokon keresztül kölcsönhatásba kerülnek. Egymáshoz viszonyított szaporodási rátájuk és a környezet eltartóképessége függvényében a "rátermettebb" faj akár teljesen el is foglalhatja a nichet.

### 2.2.1. Csatolt logisztikus egyenlet

A két faj modelljében figyelembe vesszük, hogy az erőforrásért folytatott verseny során az egyik faj szaporodása a másik faj erőforrásainak csökkenését is jelenti, és fordítva.

$$\frac{dn_1}{dt} = r_1 n_1 \left(1 - \frac{n_1 + \alpha n_2}{k_1}\right),$$

$$\frac{dn_2}{dt} = r_2 n_2 \left(1 - \frac{n_2 + \beta n_1}{k_2}\right),$$

ahol  $\alpha$  és  $\beta$  dimenziótlan paraméterek azt fejezik ki, milyen mértékben hat egyik faj a másik erőforrásaira és viszont. A rendszer fixpontjai függenek az  $\alpha$ ,  $\beta$ ,  $r_1$ ,  $r_2$ ,  $k_1$ ,  $k_2$  értékeiktől.

## 2.3. Ragadozó-préda rendszerek

A fajok együttélése nem csak a közös erőforrásokért folytatott harcban merülhet ki, hanem ragadozó-préda kapcsolat is létrejöhet több egyéb lehetőség mellett. Ha sok prédaállat él, a ragadozók elszaporodhatnak. Ha a ragadozók elszaporodnak, túlvadásszák a prédaállatot, így azok száma csökken. Ha a prédaállat száma lecsökken, a ragadozók éhen maradnak és az ő számuk is csökkenni fog addig, amíg a prédaállatok újra elszaporodhatnak és kezdődhet az egész elölről. A következő részekben a ragadozókat az egyszerűség kedvéért rókának, a prédaállatokat pedig nyulaknak fogom venni.

### 2.3.1. Lotka-Volterra-modell

Hipotézisek: a nyulak (R) számára korlátlan táplálék áll rendelkezésre, így szaporodásukat az erőforrások nem korlátozzák. A rókák kapacitása korlátlan, vagyis a nyulak számától lineárisan függ, hogy egy róka hány nyulat tud megenni adott idő alatt.

$$\frac{dn_R}{dt} = an_R - bn_F n_R,$$

$$\frac{dn_F}{dt} = cn_R n_F - dn_F.$$

A paraméterek jelentése:

1.  $n_R$ : a nyulak száma,
2.  $n_F$ : a rókák száma,
3.  $a$ : a nyulak születési-halálozási rátája,
4.  $cn_R$ : a rókák szaporodási rátája,
5.  $bn_F$ : a nyulak pusztulási rátája,
6.  $d$ : a rókák pusztulási rátája.

### 2.3.2. Realisztikusabb Lotka-Volterra-modell

A Lotka-Volterra-modell realisztikusabbá tehető, ha figyelembe vesszük, hogy a nyulak számára rendelkezésre álló élelem korlátos, illetve, hogy egy róka nyúl fogyasztási kapacitása véges, hiába van belőle rengeteg.

A fentieket figyelembevéve két taggal lehet kiegészíteni az egyenleteket:

$$a \rightarrow a\left(1 - \frac{n_R}{k}\right),$$

$$n_R n_F \rightarrow \frac{n_R n_F}{1 + \frac{n_R}{s}},$$

ahol  $k$  a nyulak kapacitása és  $s$  a rókák telítődése.

Így a realisztikusabb Lotka-Volterra-modell:

$$\frac{dn_R}{dt} = a\left(1 - \frac{n_R}{k}\right)n_R - b\frac{n_R n_F}{1 + \frac{n_R}{s}},$$

$$\frac{dn_F}{dt} = c\frac{n_R n_F}{1 + \frac{n_R}{s}} - dn_F.$$

### 2.3.3. Tápláléklánc 3 fajra

Az utolsó feladat során válsztani lehetett két egymással versengő növényevő és egy rájuk vadászó ragadozó, vagy egy három főből álló tápláléklánc szimulációja közül. Én az utóbbit választottam, a Déli sarkon élő fókákat (F), pingvineket (P) és halakat (H) használva példaállatoknak. A modellem alapjául a realisztikusabb Lotka-Volterra-modell szolgál. Két kiegészítést tettem, az egyik, hogy a középén elhelyezkedő ragadozónak is van külön halálozási rátája (így akkor is kipusztul, ha a rá vadászó csúcsragadozók ugyan elfogynak, de nem marad élelme), a másik, hogy egyik fajt sem hagyom teljesen kipusztulni, minden körben a fajok elemszáma legalább egy (ez az egyenletekben nem jelenik meg, csak a programkódban, a ciklustörzsben minden kör során ellenőrzöm a fajok számát, és ha kisebb-egyenlő mint nulla, akkor egyre módosítom). Így a modellem a következő formába önthető:

$$\begin{aligned}\frac{dn_H}{dt} &= a\left(1 - \frac{n_H}{k}\right)n_H - b\frac{n_H n_P}{1 + \frac{n_H}{s_P}}, \\ \frac{dn_P}{dt} &= c\frac{n_H n_P}{1 + \frac{n_H}{s_P}} - d\frac{n_P n_F}{1 + \frac{n_P}{s_F}} - e n_P, \\ \frac{dn_F}{dt} &= f\frac{n_P n_F}{1 + \frac{n_P}{s_F}} - g n_F.\end{aligned}$$

## 3. Megoldás módszerei

A populációdinamika szimuláció numerikus megoldásához az Euler és az adaptív negyedrendű Runge-Kutta differenciálegyenlet megoldó módszereket fogom alkalmazni, melyeket a következő alfejezetben tárgyalok.

### 3.1. Differenciálegyenlet megoldó módszerek

#### 3.1.1. Euler algoritmus

Az elsőrendű, explicit Runge-Kutta módszert hívjuk Euler algoritmusnak, lokális csonkolási hibája (LTE)  $\mathcal{O}(\delta t^3)$ , a globális csonkolási hibája pedig  $\mathcal{O}(\delta t^2)$ . A feladat során használt módszerek közül az Euler algoritmus numerikusan a legkevésbé stabil. Az algoritmus az inga esetére nézve a következőképpen írható fel:

$$\begin{aligned}t_{n+1} &= t_n + \delta t \\ \theta_{n+1} &= \theta_n + \omega_n \delta t \\ \omega_{n+1} &= \omega_n + \beta_n \delta t \\ \beta_n &= -\frac{g}{l} \sin(\theta) - q\omega + F_D \sin(\omega_D t)\end{aligned}$$



### 3.1.2. Negyedrendű Runge-Kutta módszer (RK4)

A negyedrendű Runge-Kutta módszer explicit Runge-Kutta módszer, lokális csonkolási hibája (LTE)  $\mathcal{O}(\delta t^5)$ , a globális csonkolási hibája pedig  $\mathcal{O}(\delta t^4)$ , numerikusan stabil. Felépítése a következő:

$$\begin{aligned}k_1 &= f(t_n, y_n) \\k_2 &= f\left(t_n + \frac{\delta t}{2}, y_n + \frac{k_1}{2}\right) \\k_3 &= f\left(t_n + \frac{\delta t}{2}, y_n + \frac{k_2}{2}\right) \\k_4 &= f(t_n + \delta t, y_n + k_3) \\y_{n+1} &= y_n + \frac{\delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}$$

,

ahol  $f$  a léptetéshez szükséges deriváltfüggvény az adott helyen és időben. Az implementáció során az idő, elmozdulás és sebesség mennyiségeket vektorként kezeltem, az  $f$  derivált függvény pedig az adott vektorkomponensekhez tartozó deriváltakat adja vissza a következő módon:

$$\begin{aligned}\frac{dt}{dt} &= 1 \\ \frac{dr_x}{dt} &= v_x \\ \frac{dr_y}{dt} &= v_y \\ \frac{dv_x}{dt} &= -Gm_{Sun} \frac{r_x}{(r_x^2 + r_y^2)^{3/2}} \\ \frac{dv_y}{dt} &= -Gm_{Sun} \frac{r_y}{(r_x^2 + r_y^2)^{3/2}}\end{aligned}$$

### 3.1.3. Adaptív negyedrendű Runge-Kutta módszer (ARK4)

Az adaptív negyedrendű Runge-Kutta módszer annyival tér el a nem adaptív változattól, hogy lépéshosszát a lokális hiba szerint változtatja: ha bármelyik mennyiség becsült hibája meghaladja a pontossági paramétert, a lépéshossz lecsökken, majd az adott lépést és hibáját újra számolja az algoritmus az új lépéshosszal. Ugyanez a helyzet akkor is, ha a lépéshez tartozó hiba nagyon alacsony értéket ad, csak akkor a lépéshosszt nem csökkenteni, hanem növelni fogjuk, amíg a hiba megfelelő tartományba nem kerül. Minél meredekebb a derivált, annál kisebb lépéshosszra van szükség ugyanazon pontosság eléréséhez. A becsült hiba kiszámítási módja az, hogy minden lépést kétszer végzünk el: első verzióban, az egyszerű negyedrendű Runge-Kuttához hasonlóan kiszámolunk egy teljes lépést, a második verzióban pedig két fél lépést teszünk meg, majd megnézzük a két verzió közötti eltérést, mely megadja a becsült lokális csonkolási hibát az adott lépésre.

## 4. Kiértékelés

### 4.1. Logisztikus egyenlet

Az első feladat során a logisztikus egyenletet szimuláltam Euler- és adaptív negyedrendű Runge-Kutta-módszerek segítségével. Több paramétert is kipróbáltam, ezekről ábra a függelékben található.

Azt tapasztaltam, hogy az adaptív Runge-Kutta módszer segítségével mindössze pár pont felhasználásával is nagyon pontos eredményt kaphatunk, szemben az Euler-módszerrel, ahol az állandó lépéshossz miatt pontjaink száma sokszorosa a Runge-Kutta módszer által használt mennyiségnek, az eredményünk mégis pontatlanabb. A jelenlegi felbontásban ez így is alig észrevehető. Megfigyelhető még, hogy minél távolabb vagyunk a fix ponttól, az eltérés annál nagyobb. Ennek oka valószínűleg az, hogy a fix pont közelében kisebbek a deriváltak, így ott a hiba is kisebb.

### 4.2. Versengés

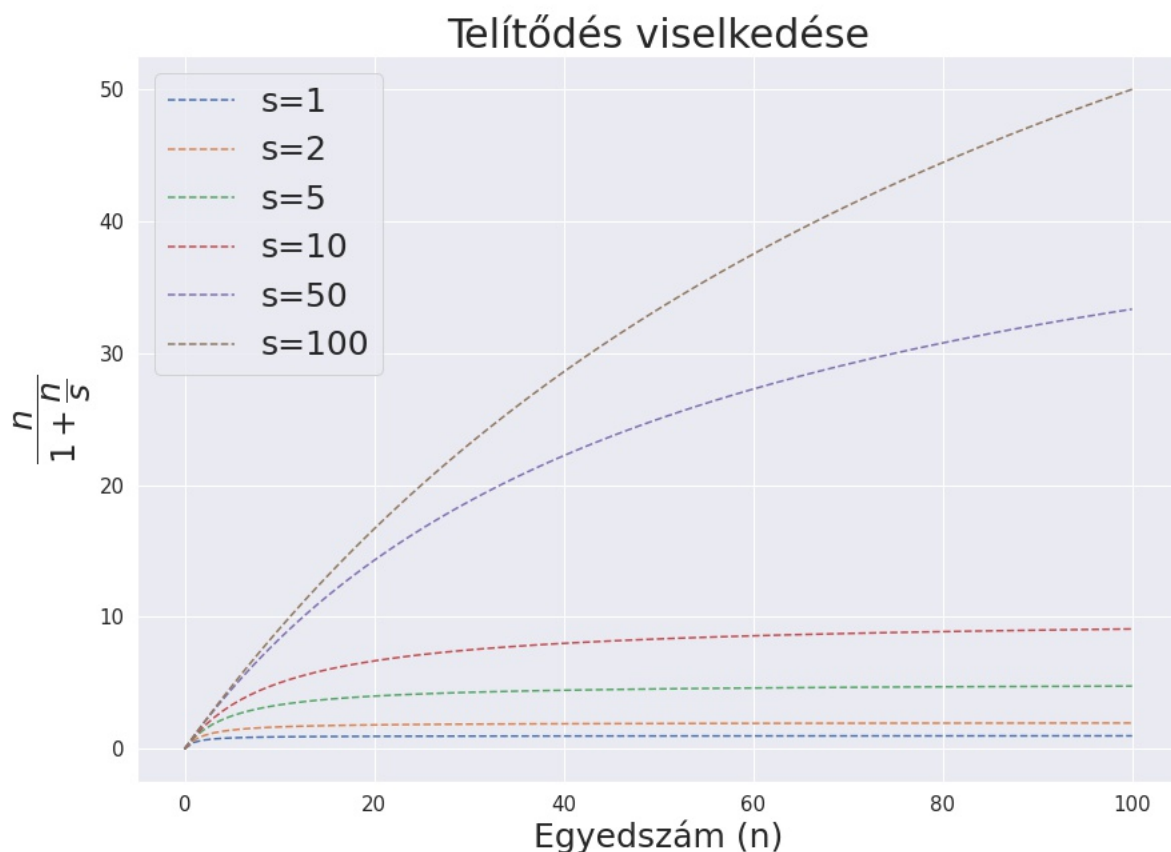
A második feladat folyamán két egymással versengő faj populációváltozását szimuláltam, itt már csak az adaptív negyedrendű Runge-Kutta-módszert használtam. Először egy példával bemutattam a kompetitív kizárás törvényét, mely  $\alpha = \beta = 1$  esetben fordul elő. Ez esetben a nagyobb kapacitással rendelkező faj teljesen ki fogja szorítani a másikat. Ezt követően numerikus úton bebizonyítottam, hogy két faj együttélése csak  $\alpha k_1 < k_2$  és  $\beta k_1 < k_2$  esetén lehetséges. Ehhez az  $\alpha$  és  $\beta$  paraméterek változtatásával gyártottam példákat, majd ábrázoltam a hosszútávon megmaradt fajok számát az  $\alpha k_2 - k_1$  -  $\beta k_1 - k_2$  síkon. A függelékben elhelyezett ábra igazolja a fenti összefüggést, jól látható, hogy csak ott marad meg mind a két faj egyszerre, ahol mindkét tengelynek negatív az értéke.

### 4.3. Lotka-Volterra-modell

Ennél a feladatnál az egyszerű Lotka-Volterra modellt implementáltam és ábrázoltam a populációk létszámát és a populációk fázisterét, az ábrák megtalálhatók a függelékben.

### 4.4. Realisztikus Lotka-Volterra-modell

A harmadik feladat után most a relativisztikus Lotka-Volterra-modellt implementáltam és a harmadik feladathoz hasonló ábrákat készítettem, melyek a függelékben megtekinthetők. Készítettem egy értelmező ábrát a rókák telítődési számához ( $s$ ), ez lejjebb látható:



1. ábra

Minél kisebb  $s$  értéke, annál kevésbé tudnak a ragadozók többet fogyasztani a prédákból annak ellenére, hogy a prédák létszáma nő.

A függelék ábrái alapján a következő következtetéseket tudtam még levonni:

ha nem veszem számításba a nyulak véges táplálékát és a rókák telítődési száma megfelelő (vagyis nem halnak meg a rókák a nyulak kezdeti populáció létszáma mellett), akkor a nyulak és rókák populáció létszáma egymással szinkronban oszcillálva divergálni kezdenek a végtelenbe (hol felszökik a létszámuk, hol nulla közelébe esik vissza, és minden felszökés exponenciálisan növekszik az előzőekhez képest). Ha számításba veszem a nyulak véges táplálékmennyiségét és a rókák telítődési száma megfelelő, akkor a két faj tud egymás mellett létezni, mindkét faj populációs száma tart egy-egy adott értékhez. Ha a telítődési szám túl kicsi, a rókák kihalnak, a nyulak pedig beállnak a kapacitásuk értékére. Tehát a triviális (a rókák kipusztulnak) esetet leszámítva, a fixpontok a nyulak korlátozott kapacitása és a rókák megfelelően beállított telítődési száma mellett tudnak létrejönni, a lehetséges értékek közül pedig a kisebb telítődési számok mellett áll be gyorsabban az egyensúly.

## 4.5. Tápláléklánc

Az utolsó feladatban, ahogy azt már az elmélet végén is írtam, egy három fajból álló táplálékláncot szimuláltam. Az antarktisz élővilágából merítettem, a táplálékláncom planktonevő halak, pingvinek, és a pingvinekre vadászó fókákból áll. Felteszem, hogy a halak csak planktonokat, a pingvinek csak halakat, a fókák pedig csak pingvineket fogyasztanak. Az ezekhez tartozó ábrák szintén a függelékben lelhetők fel.

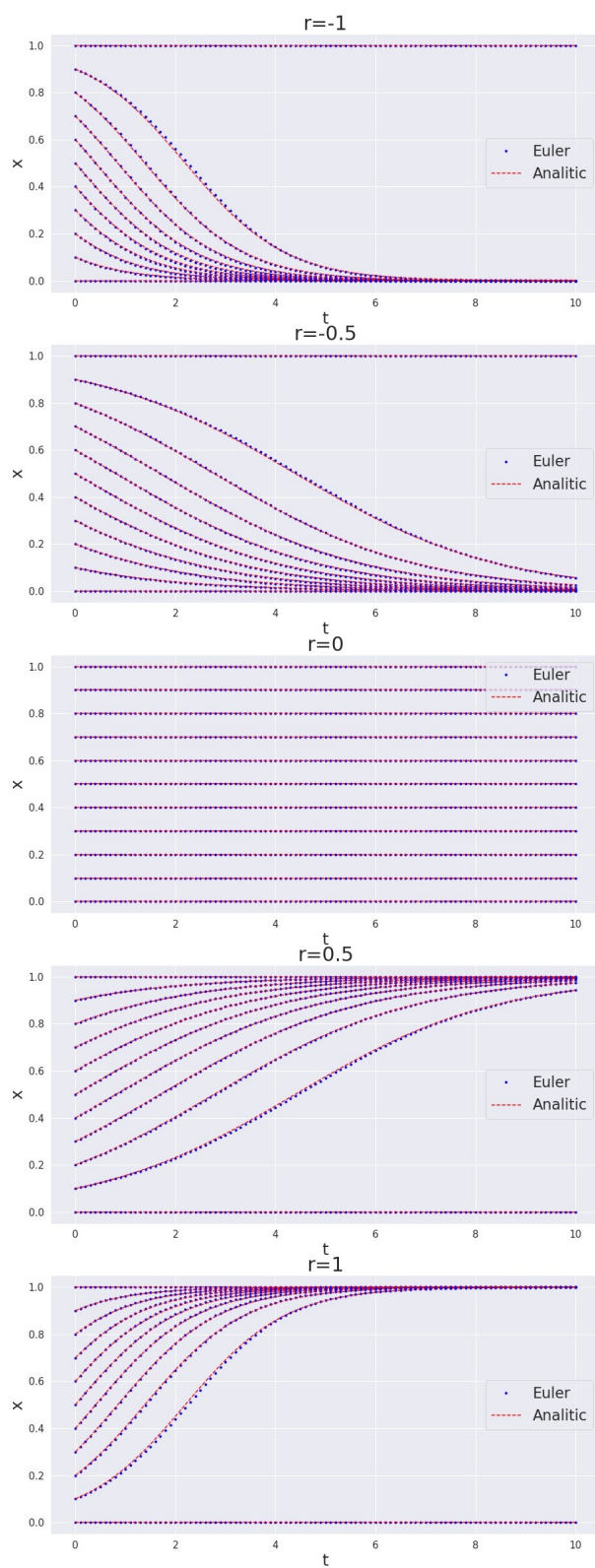
## 5. Diskusszió

Ebben a jegyzőkönyvben a számítógépes szimulációk tárgy populációdinamika témakörét dolgoztam fel. Megvizsgáltam a logisztikus egyenletet Euler- és adaptív negyedrendű Runge-Kutta-módszerrel, ezeket összevettem a logisztikus egyenlet analitikus megoldásával. Ezt követően két, egymással versengő faj szaporodását szimuláltam a csatolt logisztikus modell segítségével, példát mutattam a kompetitív kizárás törvényére illetve megmutattam két faj együttélésének szükséges feltételeit. Implementáltam a Lotka-Volterra-modellt és kiegészíttem úgy, hogy figyelembe vegye a véges táplálék illetve telítődés hatását. Ezekről populáció létszám és fázistér ábrákat készítettem. Végül létrehoztam egy három fajból álló tápláléklánc modelljét, ennek szemléltetésére is készítettem populáció létszám és fázistér ábrát.

## 6. Függelék

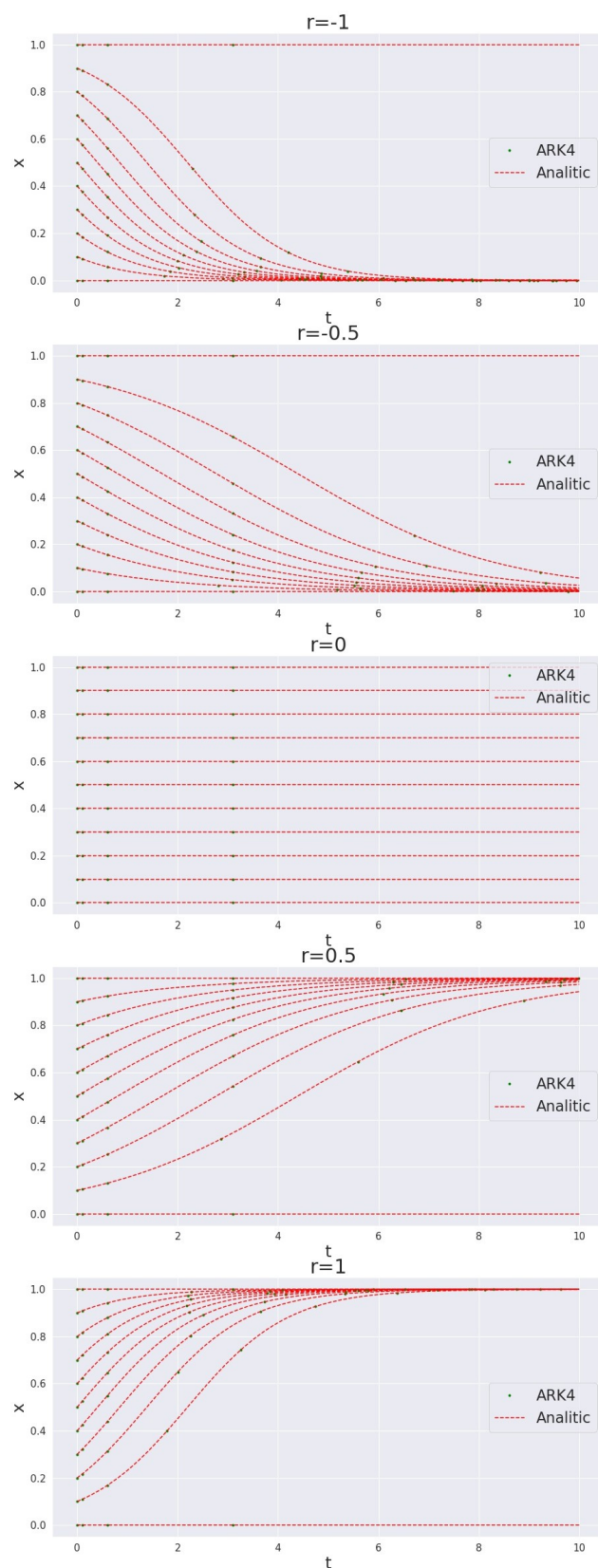
### 6.1. Logisztikus egyenlet

#### 6.1.1. Euler-módszer



2. ábra. A logisztikus egyenlet numerikus (Euler) és analitikus megoldásai. Az időbeli lépésköz  $dt = 0.1$  volt.

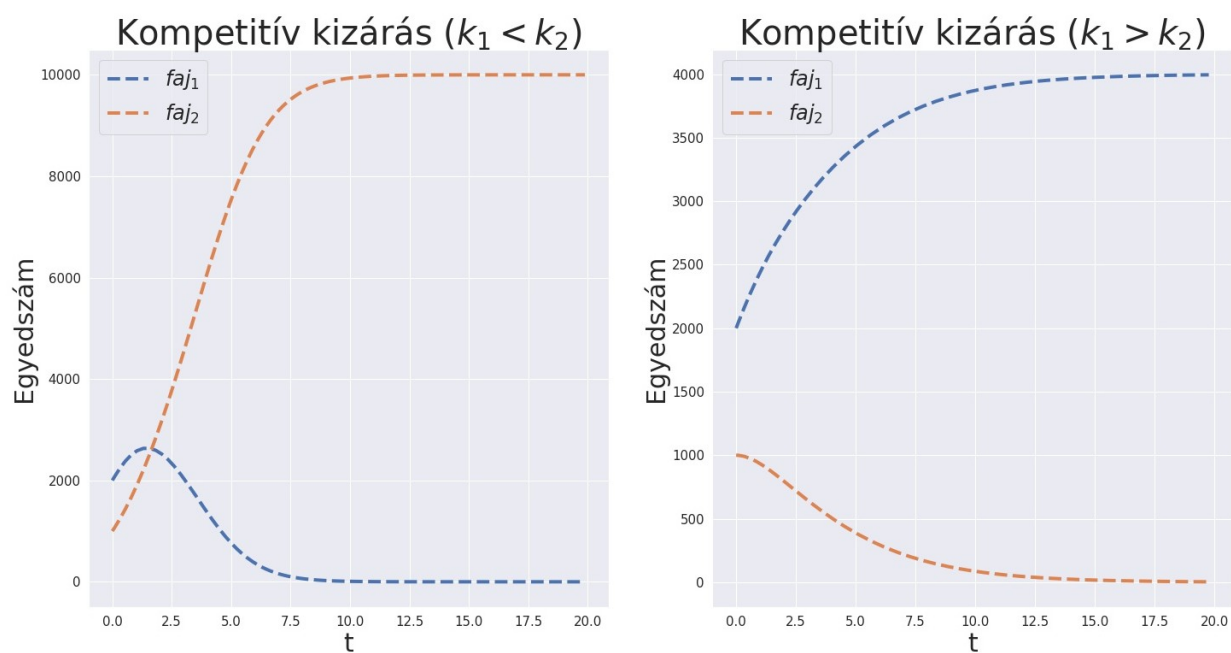
### 6.1.2. Adaptív negyedrendű Runge-Kutta módszer



3. ábra. A logisztikus egyenlet numerikus (adaptív negyedrendű Runge-Kutta (ARK4)) és analitikus megoldásai. Az ARK4-hez tartozó pontosság  $accuracy = 0.01$  volt.

## 6.2. Versengés

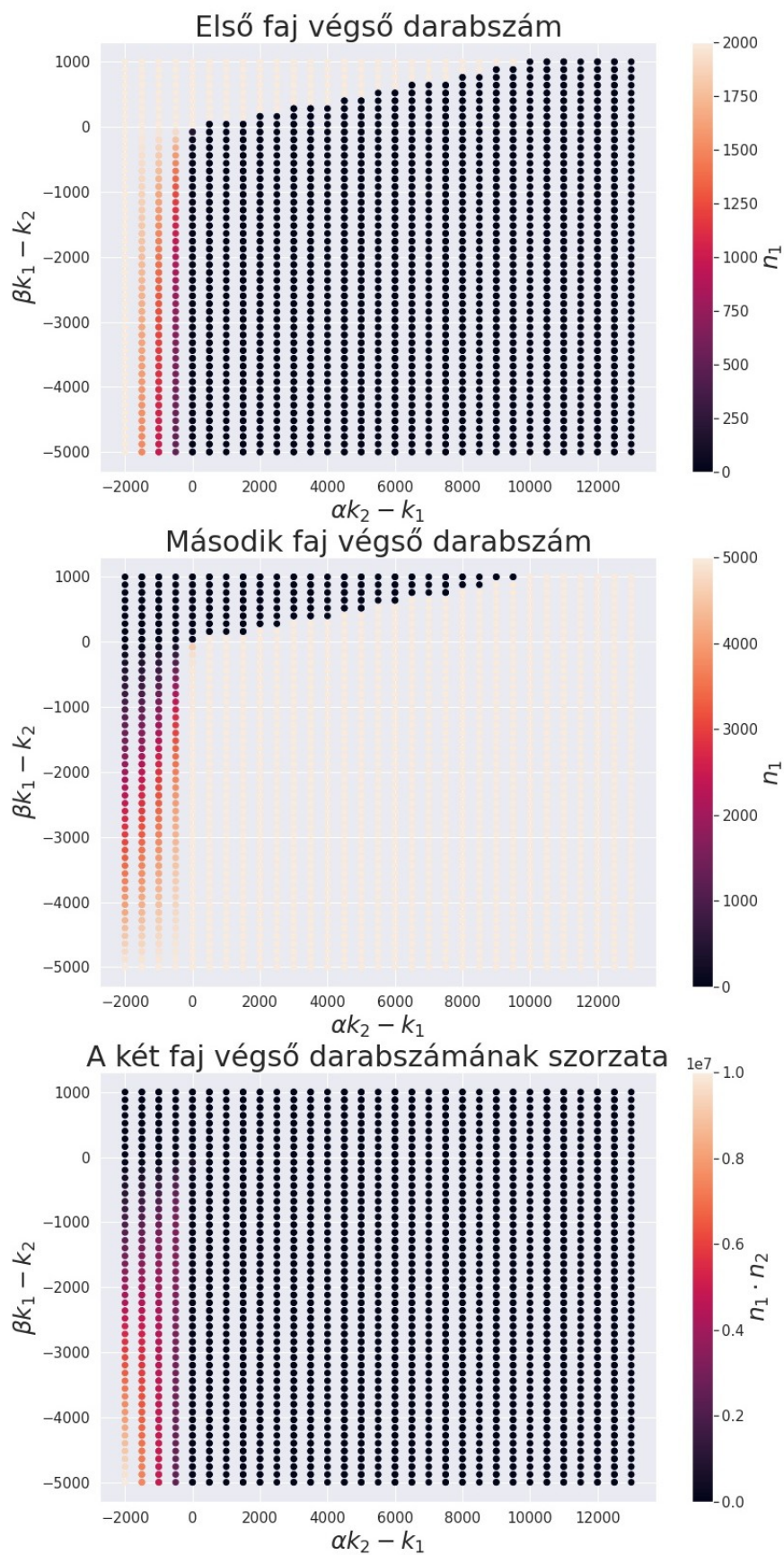
### 6.2.1. Kompetitív kizárás



4. ábra. Kompetitív kizárás,  $\alpha = \beta = 1$  (adaptív negyedrendű Runge-Kutta).  $r_1 = 1$ ,  $r_2 = 1$ ,  $accuracy = 1e - 6$ , első ábrán  $k_1 = 5000$ ,  $k_2 = 10000$ , második ábrán  $k_1 = 4000$ ,  $k_2 = 3000$ .



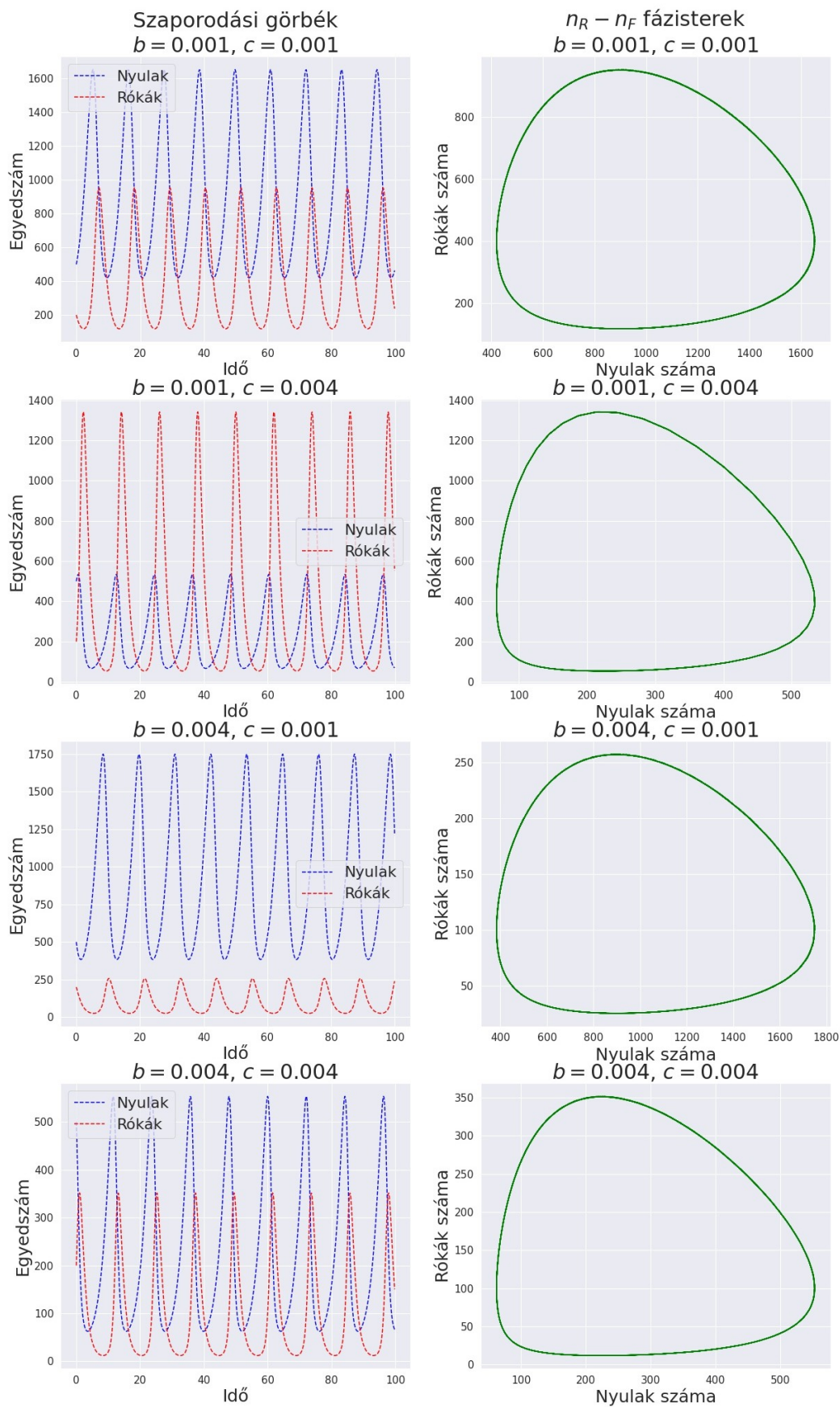
## 6.2.2. Két faj együttélése



5. ábra. Két faj együttélésének vizsgálata (adaptív negyedrendű Runge-Kutta).  $r_1 = 1$ ,  $r_2 = 1$ ,  $k_1 = 2000$ ,  $k_2 = 5000$   $accuracy = 1e - 6$ .

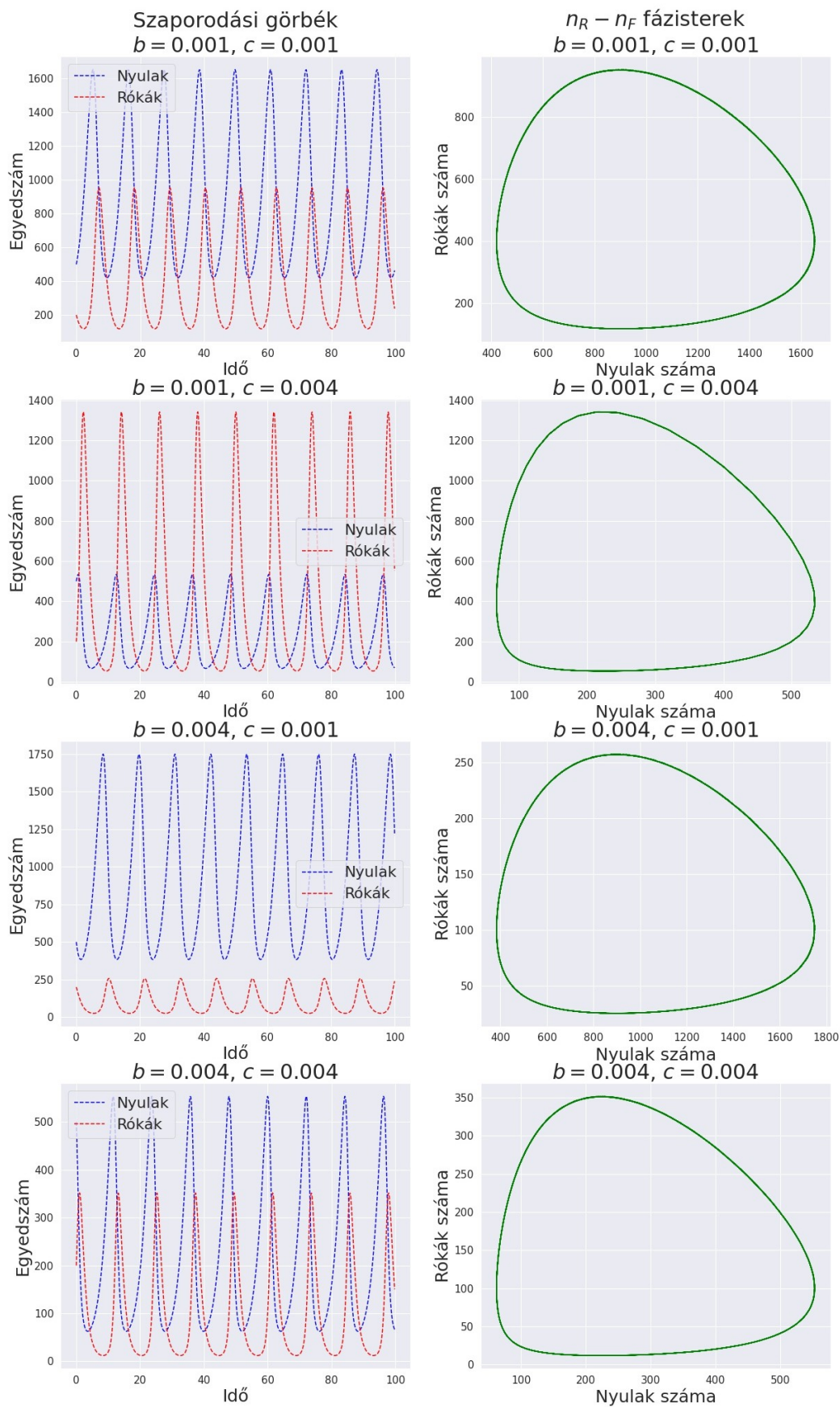


### 6.3. Lotka-Volterra-modell



6. ábra. Lotka-Volterra-modell ábrái (adaptív negyedrendű Runge-Kutta).  $a = 0.4, d = 0.9$ ,  $accuracy = 1e - 6$ .

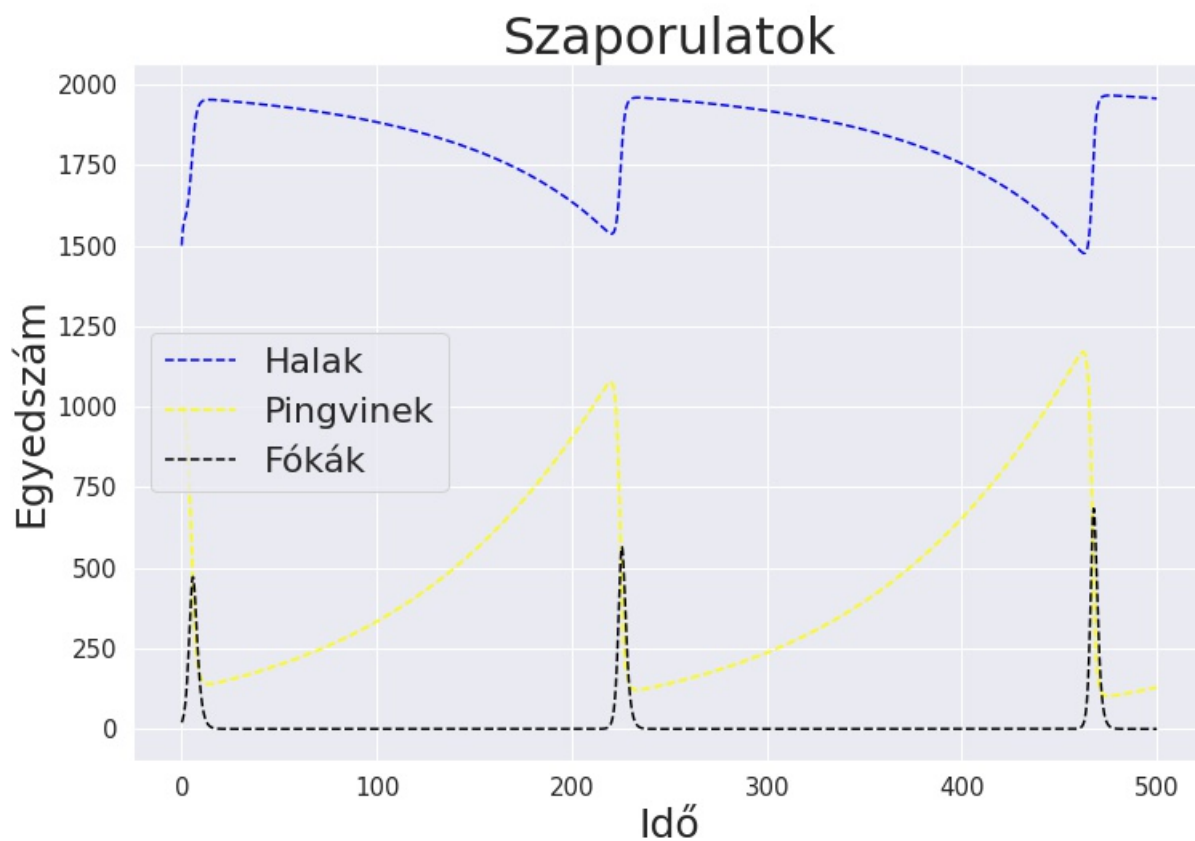
## 6.4. Realisztikus Lotka-Volterra-modell



7. ábra. Realisztikus Lotka-Volterra-modell ábrái (adaptív negyedrendű Runge-Kutta).  $a = 0.4$ ,  
 $b = 0.004, c = 0.004, d = 0.9, accuracy = 1e - 6$ .

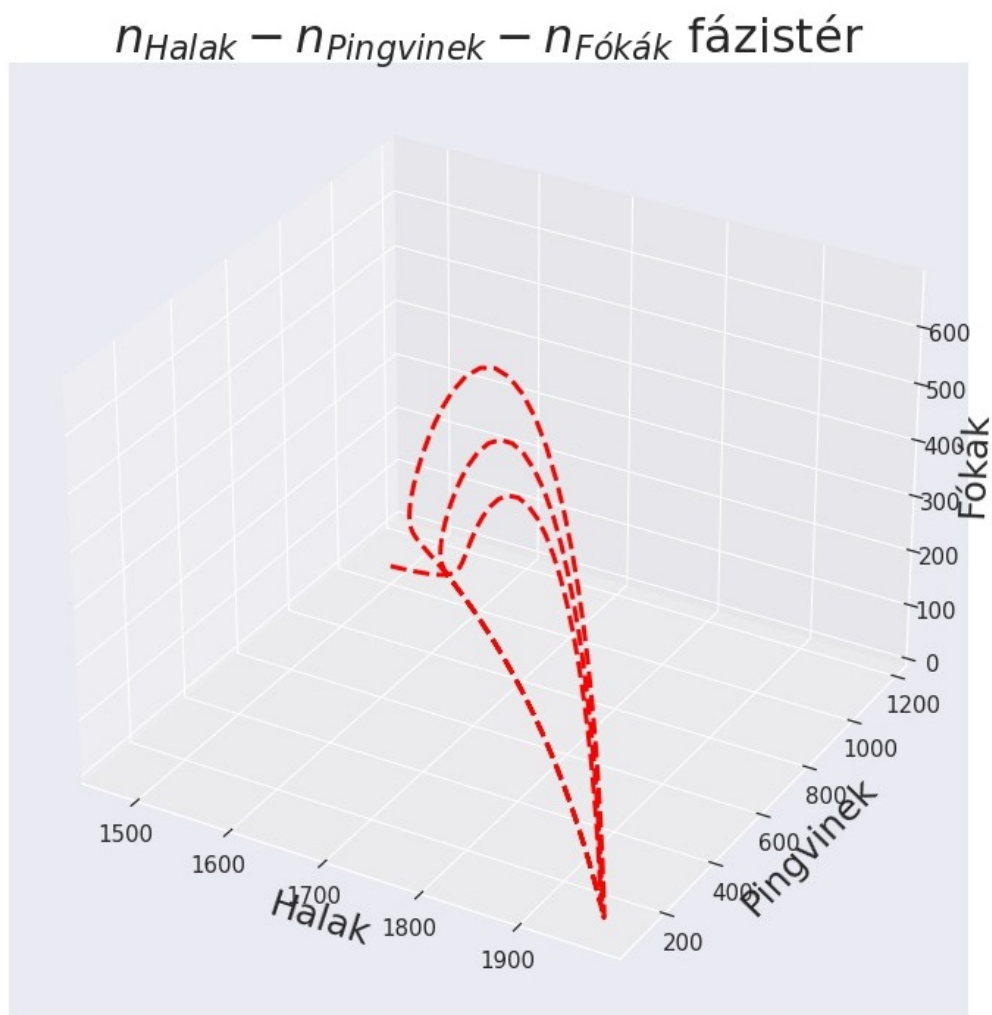
## 6.5. Tápláléklánc

### 6.5.1. Populációk létszámai



8. ábra. Tápláléklánc populációk létszámai (adaptív negyedrendű Runge-Kutta).  $a = 3$ ,  $b = 0.1$ ,  $c = 0.09$ ,  $d = 0.001$ ,  $e = 0.885$ ,  $f = 0.0025$ ,  $g = 0.9$ ,  $k = 2000$ ,  $sP = 10$ ,  $sF = 2000$ ,  $T = 500$ ,  $accuracy = 1e - 6$ .

### 6.5.2. Populációk fázistere



9. ábra. Tápláléklánc populációk fázistere (adaptív negyedrendű Runge-Kutta).  $a = 3$ ,  $b = 0.1$ ,  $c = 0.09$ ,  $d = 0.001$ ,  $e = 0.885$ ,  $f = 0.0025$ ,  $g = 0.9$ ,  $k = 2000$ ,  $sP = 10$ ,  $sF = 2000$ ,  $T = 500$ ,  $accuracy = 1e - 6$ .