EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

Szakdolgozat

Késleltetést tartalmazó differenciálegyenletek numerikus megoldása

Készítette:

Horváth Matilda Anna

Matematika BSc - matematikai elemző szakirány

Témavezető:

Dr. Csomós Petra

Alkalmazott Analízis és Számításmatematikai Tanszék

BUDAPEST, 2023

Köszönetnyilvánítás

Szeretném megköszönni konzulensemnek, Dr. Csomós Petrának a rengeteg támogatást és segítséget, amit a szakdolgozatom elkészítéséhez nyújtott.

Tartalomjegyzék

Bevezetés

A szakdolgozat témája a késleletetést tartalmazó közönséges differenciálegyenletek és azok numerikus megoldásai. A dolgozat célja rámutatni arra, hogyan változik a numerikus megoldási módszer, ha a differenciálegyenletünk késleltetést is tartalmaz. Ebből kifolyólag egyaránt találhatunk benne késleltetést nem tartalmazó és tartalmazó példákat is.

Az első két fejezetben bevezetjük a differenciálegyenletekhez tartozó alapfogalmakat, és a két példat, amiker a dolgozat további részeiben is visszatérünk. A dolgozat harmadik fejezetében taglaljuk a dolgozat fő témáját, a numerikus megoldási módszereket. Ezután a következő részben megtekintjük a már korábban bevezetett módszereket két populációs példa keretein belül.

A dolgozat hatodik fejezetében a dolgozat ábráihoz és számolásaihoz használt MATLAB kódok találhatók.

A dolgozat legfőbb forrása Alfredo Bellen and Marino Zennaro: Numerical Methods for Delay Differential Equations [1] c. könyve, amely a szakdolgozat témájával, azaz a késleltetést nem tartalmazó és tartalmazó differenciálegyenletekkel, azok numerikus megoldásaival és a közelítésekre használt módszerekkel foglalkozik.

1. Közönséges differenciálegyenletek

Ebben a fejezetben bevezetjük a közönséges differenciálegyenlet fogalmát, majd megvizsgáljuk a megoldás létezésének feltételeit. Itt kapnak helyet azon tételek és definíciók is, amelyek elengedhetetlenek a dolgozat további fejezeteihez.

Mikor differenciálegyenletekről beszélünk, akkor tulajdonképpen az ismeretlen függvény és annak deriváltja közti kapcsolatot vizsgáljuk. Ennek az összefüggésnek a vizsgálata nemcsak elméleti szempontból fontos és érdekes, hanem rengeteg gyakorlati haszna is van a matematika mellett még a természet-, műszaki és társadalomtudományok különböző területein is. Ez azért van, mert differenciálegyenletekkel könnyen modellezhetünk olyan folyamatokat, ahol az ismeretlen mennyiség megváltozására van valamilyen információnk.

A fejezet Tóth János és Simon L. Péter: Differenciálegyenletek (Typotex, Budapest, 2020) [2], Fekete Imre: Alkalmazott Analízis jegyzet [4] és [6] online jegyzet források alapján készült.

1.1. Definíció. Legyen $G \subset \mathbb{R} \times \mathbb{R}^n$ tartomány (összefüggő, nyílt halmaz), $U \subset \mathbb{R}$ nyílt halmaz, $f : G \to \mathbb{R}^n$ folytonos függvény és $(t, y(t)) \in G$ minden $t \in U$ esetén. Ekkor ha az $y : U \to \mathbb{R}^n$ differenciálható függvényre teljesül, hogy

$$y'(t) = f(t, y(t)), \quad t \in U, \tag{1}$$

akkor az y függvényt az f jobb oldalú elsőrendű explicit differenciálegyenlet megoldásának nevezzük.

1.1. Állítás. *Legyen az $F : \mathbb{R} \times (\mathbb{R}^n)^m \to \mathbb{R}^n$ folytonos függvény által meghatározott*

$$y^{(n)}(t) = F(t, y(t), y'(t), y''(t), \ldots, y^{(m-1)})$$

egy m-ed rendű explicit egyenlet. Ekkor az $y_1 = y, y_2 = \dot{y}, \ldots, y_m = y^{(m-1)}$, $y_i : \mathbb{R} \to \mathbb{R}^n$ új függvények bevezetésével az m-ed rendű egyenlet átírható elsőrendű egyenletrendszerré:

$$\begin{aligned} y_1'(t) &= y_2(t) \\ y_2'(t) &= y_3(t) \\ &\vdots \\ y_n'(t) &= F(t, y_1(t), y_2(t), \ldots, y_m(t)) \end{aligned}$$

1.2. Definíció. Legyen $G \subset \mathbb{R} \times \mathbb{R}^n$ egy tartomány, $(t_0, y_0) \in G$ egy adott pont és $f : G \to \mathbb{R}^n$ egy folytonos leképezés. Az

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0 \tag{2}$$

feladatot kezdetiérték-feladatnak, más szóval Cauchy-feladatnak nevezzük.

1.3. Definíció. Az olyan $y : I \to \mathbb{R} \times \mathbb{R}^n$ ($I \subset \mathbb{R}$ egy nyílt intervallum) folytonosan differenciálható függvényt, amelyre:

1. $\{(t, y(t)) : t \in I\} \subset G$
2. $y'(t) = f(t, y(t)) \quad \forall t \in I$
3. $t_0 \in I, \quad y(t_0) = y_0$

a (2) kezdetiérték-feladat megoldásának nevezzük.

1.4. Definíció. Legyen $G \subset \mathbb{R} \times \mathbb{R}^n$ egy tartomány. Az $f : G \to \mathbb{R}^n$ függvényt a második változójában Lipschitz-tulajdonságúnak nevezzük, ha létezik olyan $L \geq 0$, hogy minden (t, p_1), (t, p_2) $\in G$ esetén $|f(t, p_1) - f(t, p_2)| \leq L|p_1 - p_2|$.

1.5. Tétel (Picard–Lindelöf)**.** *Legyen $f : G \to \mathbb{R}^n$ folytonos függvény, ahol*

$$G = \{(t, y) \in \mathbb{R} \times \mathbb{R}^n : |t - t_n| \leq a, |y - y_0| \leq b\}$$

henger, $(t_0, y_0) \in \mathbb{R} \times \mathbb{R}^n$ és $0 < a < \infty, 0 < b < \infty$. Legyen $M = \max_{(t,y)\in G} |f(t, y)|$, továbbá tegyük fel, hogy az f függvény második változójában Lipschitz- tulajdonságú a G halmazon. Ekkor a kezdetiérték-feladatnak létezik egyértelmű megoldása a $[t_0 - \delta, t_0 + \delta]$ intervallumon, ahol $\delta = \min(a, \frac{b}{M})$.

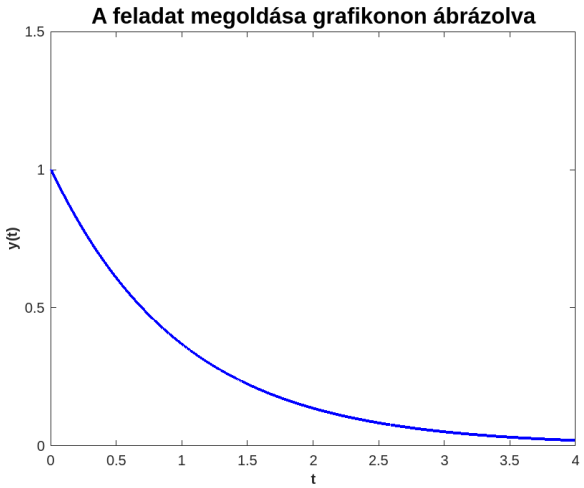
1.6. Definíció. Legyen $f : G \to \mathbb{R}^n$ folytonos függvény. Ezt az f függvényt, akkor nevezzük sima függvénynek, ha G intervallumon az f csak véges számú pontban nem folytonos, és a nem folytonos pontok között f deriváltja folytonos.

1.7. Példa. Keressük azt az $f : \mathbb{R} \to \mathbb{R}$ függvényt, melyre

$$y'(t) = -y(t), \quad y(0) = 1$$

A feladat egyértelmű megoldása $y(t) = e^{-t}$. Ezt úgy kapjuk meg, hogy az egyenletet beszorozzuk e^t -vel, majd felismerjük, hogy egy szorzat deriváltját kaptuk:

$$\begin{aligned} y'(t) &= -y(t) \\ y'(t) \cdot e^t &= -y(t) \cdot e^t \\ y'(t) \cdot e^t + y(t) \cdot e^t &= 0 \\ (y(t) \cdot e^t)' &= 0 \\ y(t) \cdot e^t &= 1 \\ y(t) &= 1 \cdot e^{-t} \end{aligned}$$



1. ábra: Az y függvény grafikonja

Az 1. ábrán láthatjuk az 1.7. példa esetén kapott megoldás függvényét $t \in [0, 4]$ esetén.

2. Késleltetést tartalmazó differenciálegyenletek

A legnagyobb különbség a késleltetést nem tartalmazó közönséges és a késleltetést tartalmazó differenciálegyenletek között az, hogy míg az előbbiben a megoldás csak egy korábbi y_0 értéktől függ, addig a késleltetést tartalmazó differenciálegyenletek esetében ezt egy $\varphi(t)$ függvény váltja fel. Ez azért van, mert a függvényt, mint a nevéből is sejthető, késleltetjük, ezt egy τ értékkel fogjuk jelölni. Erre a késleltetésre akkor van szükség, ha a függvénnyel egy korábbi adat deriváltja áll összefüggésben, nem pedig önmagának a deriváltja. Ekkor τ jelöli, hogy a mennyivel ezelőtti függvényérték az, amellyel összefüggésben áll. A korábban bevezetett definíciók és tételek eszerint módosulnak.

A fejezet Alfredo Bellen and Marino Zennaro: Numerical Methods for Delay Differential Equations [1] forrás 1. fejezete és Svantnerné Sebestyén Gabriella: Populációodinamikai modellek és matematikai vizsgálatuk [3] forrás alapján készült.

2.1. Állítás. *Legyen $G \subset [t_0, t_0+\beta] \times \mathbb{R}^n$ egy konvex tartomány és $0 \leq \tau_1 < \tau_2 < \tau_3, \cdots < \tau_m$. Tegyük fel, hogy $f : G \to \mathbb{R}^n$ folytonos G-n és $\varphi(t) : [t_0 - \tau_m] \to G$. Ekkor a*

$$\begin{cases} y'(t) = f(t, y(t-\tau_1), y(t-\tau_2), \ldots, y(t-\tau_m)) \\ y(t) = \varphi(t), \quad t_0 - \tau_m < t < t_0 \end{cases} \tag{3}$$

késleltetést tartalmazó differenciálegyenletnek egyértelmű megoldása van a $[t_0, t_0+\beta]$ intervallumon.

2.1. Példa. Keressük azt az $y : [-1, \infty] \to \mathbb{R}$ függvényt, melyre

$$\begin{cases} y'(t) = -y(t-1), \quad t \geq 0 \\ y(t) = \varphi(t) = 1, \quad t \in [-1, 0] \end{cases} \tag{4}$$

Ezt a 3. feladatból az alábbi változtatásból kapjuk:

$$\begin{aligned} t_0 &= 0 \\ m &= 1 \\ \tau_1 &= 1 \end{aligned}$$

azaz,

$$\begin{aligned} f(t, y(t-1)) &= -y(t-1) \\ \varphi(t) &= 1, \quad -1 \leq t < 1. \end{aligned}$$

A feladat megoldását az alábbiak alapján sejthetjük meg.

Mivel a feladat késleltetést tartalmaz, ezért a megoldását szakaszonként fogjuk tudni előállítani. A példában $\tau = 1$, így az ismeretlen y függvény deriváltja mindig az y függvény 1-gyel azelőtti értékétől függ, így 1 hosszúságú intervallumokban lépünk előre, hiszen az előző $y(t-1)$ időintervallumon a függvény már ismert.

Az y fügvvény a $t \in [0, 1]$ intervallum esetén az $y(t-1)$ értékből tudjuk előállítani, amely a $t \in [0, 1]$ miatt egyenlő lesz a kezdetiérték függvénnyel, azaz $\varphi(t-1)$ értékkel.

$t \in [0,1]$

$$y_1'(t) = -y_1(t-1) = \varphi(t) = -1$$
$$y_1(t) = \int -1dt = -t + c_1$$

A c_1 értékét abból kaphatjuk meg, hogy feltételezzük y folytonosságát:

$$y_0(0) = y_1(0)$$
$$1 = 0 + c_1.$$

Tehát

$$c_1 = 1$$
$$y_1(t) = -(t-1).$$

A továbbiakban ugyanezen logika mentén haladunk tovább, miközben szabályosságot keresünk, hogy fel tudjuk írni a példa általános megoldását.

$t \in [1,2]$

$$y_2'(t) = -y_2(t-1) = -y_1(t-1) = -(-((t-1)-1) = (t-1)-1$$
$$y_2(t) = \int t-1dt - \int -1dt = \frac{(t-1)^2}{2} - t + c_2$$

A c_2 értékét az előbbihez hasonlóan a megoldás folytonosságából kaphatjuk meg:

$$y_1(1) = y_2(1),$$

azaz

$$-(1-1) = \frac{(1-1)^2}{2} - 1 + c_2.$$

Tehát

$$c_2 = 1$$
$$y_2(t) = \frac{(t-1)^2}{2} - t + 1.$$

Itt már láthatjuk, hogy:

$$y_1(t) = -t + 1$$
$$y_2(t) = \frac{(t-1)^2}{2} - t + 1.$$

Ezek alapján azt várjuk, hogy a következő intervallumokban a megoldás úgy álljon elő, hogy mindig egy új tag adódjon hozzá az előzőekhez képest.

$t \in [2,3]$

$$y_3'(t) = -y_3(t-1) = -y_2(t-1) = \frac{((t-1)-1)^2}{2} + (t-1) - 1 = \frac{(t-2)^2}{2} + \frac{(t-1)^1}{1} - 1$$
$$y_3(t) = \int \frac{(t-2)^2}{2} dt + \int t - 1 dt - \int -1 dt = -\frac{(t-2)^3}{6} + \frac{(t-1)^2}{2} - t + c_3$$

A c_3 értékének kiszámítása:

$$y_2(2) = y_3(2)$$
$$\frac{(2-1)^2}{2} - 2 + 1 = -\frac{(2-2)^3}{6} + \frac{(2-1)^2}{2} - 2 + c_3.$$

Azaz

$$c_3 = 1$$
$$y_3(t) = -\left(\frac{(t-2)^3}{6} - \frac{(t-1)^2}{2} + t - 1\right).$$

$t \in [3,4]$

$$y_4'(t) = -y_4(t-1) = -y_3(t-1) = -\left(-\left(\frac{((t-1)-2)^3}{6} - \frac{((t-1)-1)^2}{2} + (t-1) - 1.\right)\right) =$$
$$= \frac{(t-3)^3}{6} - \frac{(t-2)^2}{2} + (t-1) - 1$$
$$y_4(t) = \int \frac{(t-3)^3}{6} dt - \int \frac{(t-2)^2}{2} dt + \int t - 1 dt - \int -1 dt =$$
$$= \frac{(t-3)^4}{4!} - \frac{(t-2)^3}{3!} + \frac{(t-1)^2}{2!} - t + c_4$$

A c_4 értékét az előbbiekhez hasonlóan úgy kaphatjuk meg, hogy

$$y_3(3) = y_4(3)$$
$$-\left(\frac{(3-2)^3}{6} - \frac{(3-1)^2}{2} + 3 - 1\right) = \frac{(3-3)^4}{4!} - \frac{(3-2)^3}{3!} + \frac{(3-1)^2}{2!} - 3 + c_4$$
$$c_4 = 1$$
$$y_4(t) = \frac{(t-3)^4}{4!} - \frac{(t-2)^3}{3!} + \frac{(t-1)^2}{2!} - t + 1.$$

Összegezve az eddig kapott megoldásainkat, azt láthatjuk, hogy:

$$y_1(t) = -t + 1$$
$$y_2(t) = \frac{(t-1)^2}{2} - t + 1$$
$$y_3(t) = -\frac{(t-2)^3}{6} + \frac{(t-1)^2}{2} - t + 1$$
$$y_4(t) = \frac{(t-3)^4}{4!} - \frac{(t-2)^3}{3!} + \frac{(t-1)^2}{2!} - t + 1$$

Ezek alapján a sejtésünk tehát az lesz, hogy $t \in [k-1,k]$ esetén,

$$y_k(t) = (-1)^k \cdot \frac{(t-(k-1))^k}{k!} + y_{k-1}(t), \quad k = 1,2,\dots$$

amely összegképlettel kifejezve

$$y_k(t) = \sum_{j=0}^{k} \left((-1)^j \cdot \frac{(t-(j-1))^j}{j!} \right), \quad k = 1,2,\dots$$

A példa grafikonja $t = 4$-ig tehát így néz ki:

2. ábra: Az y függvény grafikonja

2.2. Állítás. *A 2.1. feladat megoldása*

$$y_k(t) = \sum_{j=0}^{k} (-1)^j \cdot \frac{(t-(j-1))^j}{j!} \tag{5}$$

$k = 1,2,\dots$ *és* $t \in [k-1,k]$ *esetén.*

Bizonyítás. Az állítás bizonyítását két részre kell bontanunk, mert egyszer igazolnunk kell:

$$y_1(t) = \varphi(t),$$

másodszor pedig:

$$y_k'(t) = -y_k(t-1)$$

egyenleteket.

Az állítás azon részét, hogy

$$y_1(t) = \varphi(t)$$

könnyen bizonyíthatjuk, hiszen ezt már a feladat kiköti számunkra. Mivel $y_1(t)$ esetén $t \in [0,1]$, így ebben az esetben $t-1 \in [-1,0]$. A (4) feladatot megnézve láthatjuk, hogy:

$$y(t) = \varphi(t), \quad t \in [-1,0].$$

Most tekintsük a bizonyítás

$$y_k'(t) = -y_k(t-1)$$

részét:

$$y_k'(t) = \left(\sum_{j=0}^{k} (-1)^j \cdot \frac{(t-(j-1))^j}{j!} \right)' = \sum_{j=0}^{k} (-1)^j \cdot \frac{(t-(j-1))^{j-1}}{j!} \cdot j.$$

Ebben az esetben könnyen láthatjuk, hogy $j=0$ esetén egy konstanst kapunk, hiszen

$$-1^0 \cdot \frac{(t-(0-1))^{0-1}}{0-1!} = 1$$

ennek a deriváltja 0, így a szummát átindexelhetjük $j=1$-től és k-ig, azaz

$$y_k'(t) = \sum_{j=1}^{k} (-1)^j \cdot \frac{(t-j-1)^{j-1}}{j-1!},$$

és ez egyenlő

$$y_k'(t) = \sum_{j=0}^{k-1} (-1)^{j+1} \cdot \frac{(t-j)^j}{j!}.$$

Emellett felhasználjuk, hogy $y_k(t-1) = y_{k-1}(t-1)$, ami pedig az alábbiak szerint alakítható át:

$$-y_{k-1}(t-1) = -\left(\sum_{j=0}^{k-1} (-1)^j \cdot \frac{((t-1)-(j-1))^j}{j!} \right) = -\left(\sum_{j=0}^{k-1} (-1)^j \cdot \frac{(t-j)^j}{j!} \right) =$$
$$= \sum_{j=0}^{k-1} (-1)^{j+1} \cdot \frac{(t-j)^j}{j!}.$$

Azaz ezzel megmutattuk hogy

$$y_k'(t) = -y_{k-1}(t-1),$$

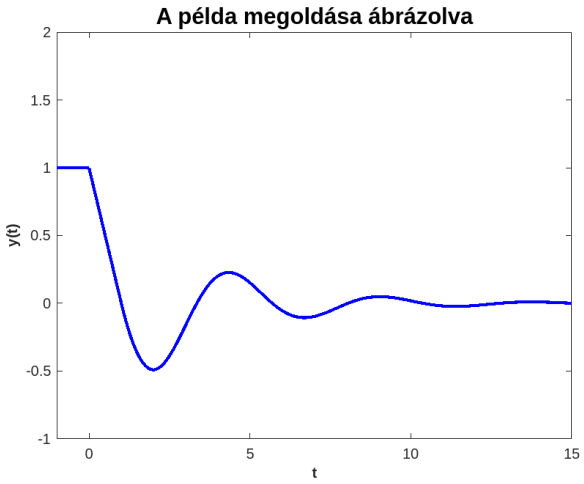
és az állítást bebizonyítottuk.

□

Ezek alapján a 2.1. példa megoldása:

$$y_k(t) = \sum_{j=0}^{k} (-1)^j \cdot \frac{(t-(j-1))^j}{j!}, \quad k = 1,2,\ldots,$$

melynek grafikonja pedig a megadott paraméterekre így néz ki:



3. ábra: Az *y* függvény grafikonja

3. Numerikus módszerek

A fejezet Alfredo Bellen and Marino Zennaro: Numerical Methods for Delay Differential Equations [1] és Faragó István – Horváth Róbert: Numerikus módszerek [5] források alapján készült.

A numerikus matematika a folytonos matematikai problémák megoldásához gyárt modelleket, és azokat elemezve igyekszik a legjobb közelítő eljárást megtalálni. Erre azért van szükség, mert a folytonos matematikai problémák sok esetben nem megoldhatók, vagy pedig megoldásuk rengeteg időt vesz igénybe.

A folyamat alatt, amíg eljutunk a folytonos matematikai problémától a megoldásig, több ponton is felléphetnek hibaforrások. Az ebben a dolgozatban található példák és feladatok során a következők hibafajták léphetnek fel:

1. Modellhiba: A legtöbb esetben a tudományos modellek nem tükrözik teljesen a valóságot. Sok esetben bizonyos körülményeket és adatokat ki kell hagynunk a modellből ahhoz, hogy az kezelhető legyen.
2. Képlethiba: Ebben az esetben azért léphet fel hiba, mert a képlet kezelhetősége érdekében sokszor leegyszerűsítjük azt, esetleg bizonyos részeit elhagyjuk.
3. Diszkretizációs hiba: A folytonos feladatok numerikus módon történő megoldása során az adott függvényt ún. rácsfüggvényekkel közelítjük, így tudjuk megadni a függvény értékét bizonyos pontokban. Azonban mivel ez csak egy közelítés, ez nem tükrözi hibátlanul a folytonos függvényt.
4. Kerekítési hiba: A számítógéppel történő számolás során bevitt adatokat a számítógép kerekíti, különböző számrendszerekben számol, így ebből is adódhatnak hibák.

Fontos az, hogy az általunk létrehozott modell hibáit mérni tudjuk. Ez garantálja, hogy össze tudunk hasonlítani különböző modelleket, és elmezni tudjuk őket.

3.1. Definíció. Adott $y \in \mathbb{R}$ esetén, amelyet egy v skalár értékkel közelítünk, abszolút hibát tudunk mérni. Abszolút hibának az $|y - v|$ értéket tekintjük.

A dolgozatban MATLAB programcsomaggal készült számítások és árázolások láthatók.

3.1. Közönséges differenciálegyenletek numerikus megoldásai

A következő alfejezetekben bevezetjük az Euler- és a Runge–Kutta-módszereket késleltetést nem tartalmazó differenciálegyenletek esetén.

3.1.1. Az explicit Euler-módszer

Az explicit Euler-módszerrel a feladat megoldását rekurzív módon állíthatjuk elő. Ez a módszer elsőrendű közelítés, így a hibára adott felsőbecsés a h zsugorítása esetén h-val csökken. Ezt az (1) elsőrendű explicit differenciálegyenlet esetén a következő sorokban látható módon tudjuk előállítani.

Először felírjuk az y függvény t helyen vett differenciálhányadosát:

$$y'(t) = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h}.$$

Legyen $h > 0$ rögzített és elegendően kicsi. Ekkor igaz lesz a következő összefüggés:

$$y(t+h) \approx y(t) + h \cdot y'(t)$$

Emelett tudjuk, hogy az f függvényre igaz, hogy:

$$f(t, y(t)) = y'(t).$$

Ekkor a korábbi közelítő egyenlőséget fel tudjuk írni úgy, hogy:

$$y(t+h) \approx y(t) + h \cdot f(t, y(t)).$$

Ezek után már felírható az Euler-módszer, hiszen

$$\begin{aligned} t_k &= k \cdot h, \quad k = 1, 2, \dots \\ y(t_k) &\approx y_k \\ y(t_k + h) &\approx y_{k+1} \end{aligned}$$

Tehát a közelítésre felírható az alábbi módszer:

$$y_{k+1} = y_k + h \cdot f(t, y_k), \quad k = 1, 2, \dots$$

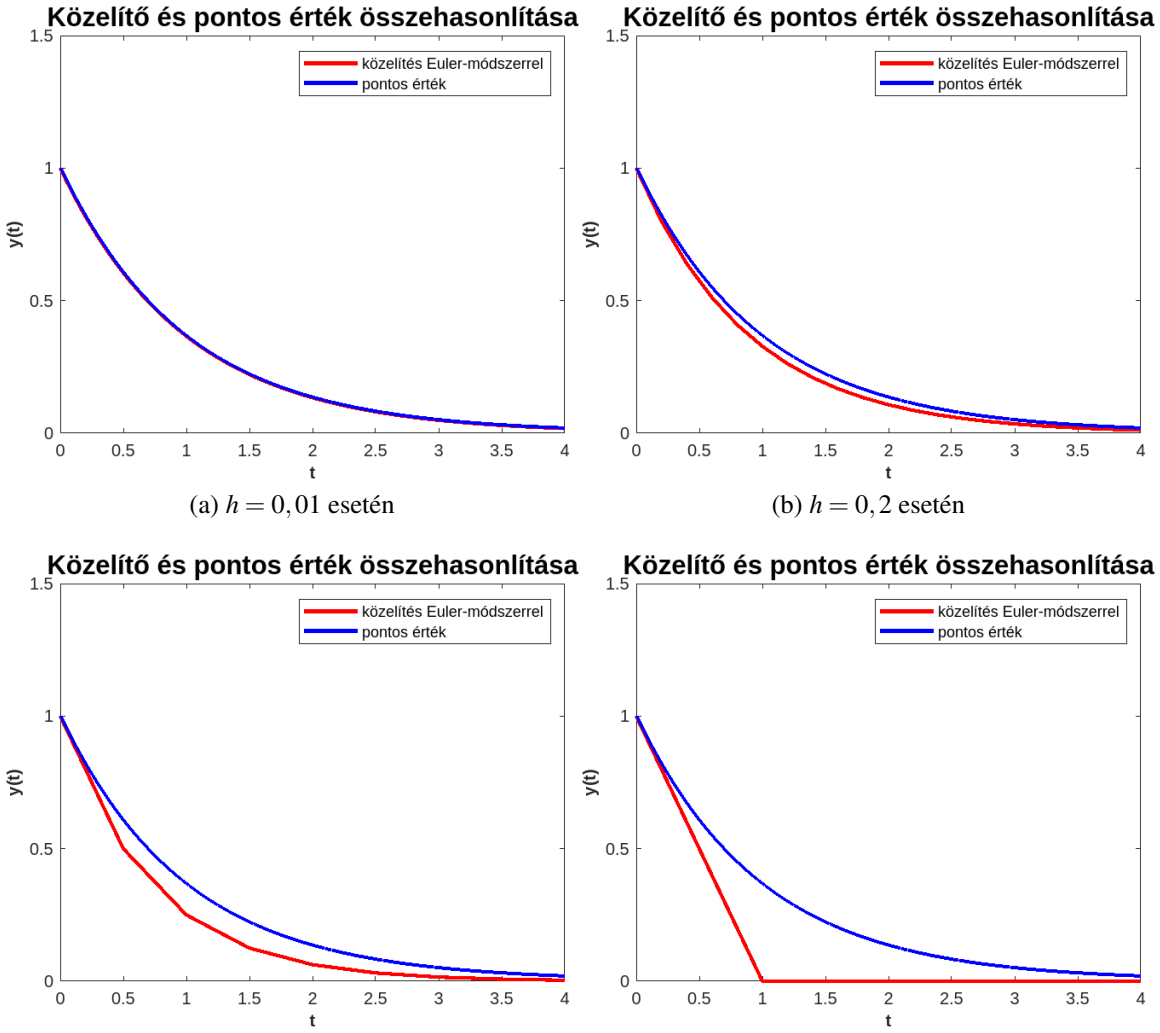
A megoldás a fentiek szerint az 1.7. példa esetében a következőképpen fog kinézni:

$$y_{k+1} = y_k + h \cdot f(t_k), \quad k = 1, 2, \cdots$$

ahol $h > 0$ a lépésköz és

$$f(t_k) = -y_k.$$

Különböző h értékek esetén különböző közelítéseket kaptunk. Megmutatható, hogy a módszer konvergens, tehát minél kisebb lépésközzel dolgozunk, az eredmény annál pontosabb lesz. A függvény értékét jelenleg nem folytonosan számoljuk ki, hanem diszkerét módon. Ezért minél több pontban számoljuk ki a közelítést, annál sűrűbben lesznek a kiszámolt pontok, és így az eredmény is pontosabb lesz. Ezt szemléletesen a 4. ábrán tekinthetjük meg, ahol pirossal a különböző h értékek esetén kapott eredményt láthatjuk, a kék görbe pedig a pontos megoldás grafikonja.



(a) $h = 0,01$ esetén

(b) $h = 0,2$ esetén

(c) $h = 0,5$ esetén

(d) $h = 1$ esetén

4. ábra: Euler-módszer által számolt megoldások különböző h értékekre

3.1.2. A Runge–Kutta-módszer

A negyedrendű Runge–Kutta-módszerrel az (1) elsőrendű explicit differenciálegyenlet kezdeti-értékfeladat megoldásának negyedrendben konvergens közelítését kaphatjuk meg. Negyedrendű közelítés esetén a h zsugorításával h^4-nel csökken a hibára adott felső becslés is.

Először tekintsük meg az algoritmust az (1) esetén, ekkor a módszerrel kapott közelítésnek a alakja az alábbi:

$$y_{k+1} = y_k + \frac{h}{6} \cdot (F_1 + 2 \cdot F_2 + 2 \cdot F_3 + F_4), \quad k = 1,2,\ldots$$

ahol $h > 0$ a lépésköz és

$$\begin{aligned} F_1 &= f(t_k, y_k) \\ F_2 &= f(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} \cdot F_1) \\ F_3 &= f(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} \cdot F_2) \\ F_4 &= f(t_k + h, y_k + h \cdot F_3). \end{aligned}$$

Láthatjuk, hogy a közelítés módja hasonló az Euler-módszerhez, tulajdonképpen annak egy javított változata. Mivel itt a meredekségeket is becsüljük és azokat súlyozzuk, ezért pontosabb megoldást kapunk, mint az Euler-módszernél.

A módszert az 1.7. példa esetén a következőképpen írhatjuk fel:

$$y_{k+1} = y_k + \tfrac{h}{6} \cdot (F_1 + 2 \cdot F_2 + 2 \cdot F_3 + F_4), \quad k = 1, 2, \ldots$$

ahol h a lépésköz és

$$\begin{aligned} F_1 &= f(y_k) \\ F_2 &= f(y_k + \tfrac{h}{2} \cdot F_1) \\ F_3 &= f(y_k + \tfrac{h}{2} \cdot F_2) \\ F_4 &= f(y_k + h \cdot F_3). \end{aligned}$$

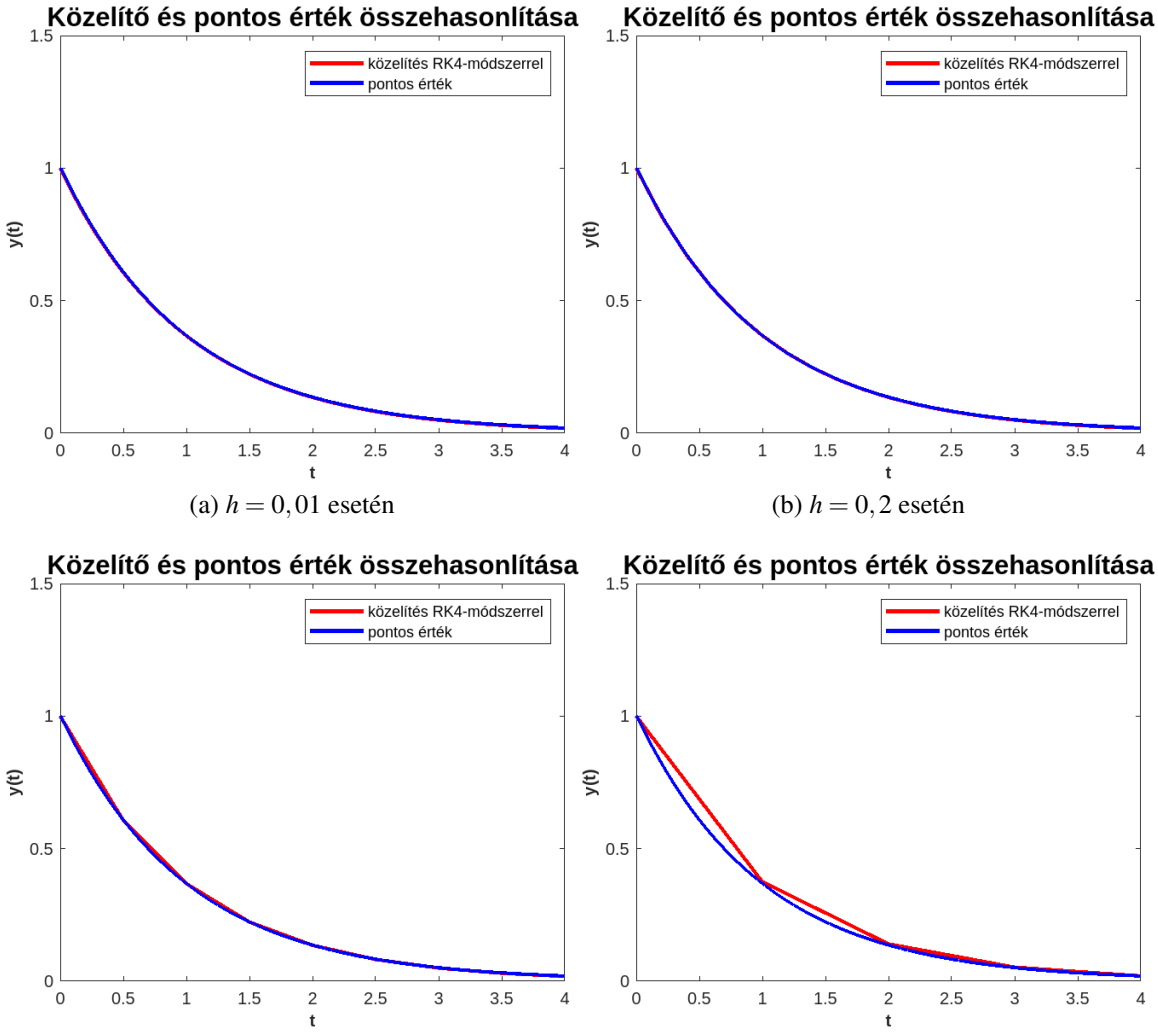
Ahol a az 1.7. példában található f függvényt a korábbiakhoz hasonlóan ismét

$$f(t, y) = -y$$

képlettel tudjuk kiszámolni, tehát

$$\begin{aligned} F_1 &= -y_k \\ F_2 &= -(y_k + \tfrac{h}{2} \cdot F_1) \\ F_3 &= -(y_k + \tfrac{h}{2} \cdot F_2) \\ F_4 &= -(y_k + h \cdot F_3). \end{aligned}$$

Az 5. ábrán itt is láthatjuk, hogy különböző h értékekre mennyire pontos közelítést kaphatunk. Megfigyelhetjük, hogy a Runge–Kutta-módszerrel való közelítés láthatóan sokkal jobb eredményt ad a különböző h értékekre, akkor is ha a h nem kicsi. Ez azért van, mert a Runge–Kutta-módszer több meredekség becslése alapján számol, így sokkal jobban meg tudja becsülni az eredeti függvény alakját.



(a) $h = 0,01$ esetén

(b) $h = 0,2$ esetén

(c) $h = 0,5$ esetén

(d) $h = 1$ esetén

5. ábra: Runge–Kutta-módszer által számolt megoldások különböző h értékekre

3.2. Késleltetést tartalmazó differenciálegyenletek numerikus megoldása

Az alfejezetben megnézzük, hogy a fent bevezetett módszereket egy kis változtatással a késleltetést tartalmazó differenciálegyenletek esetén is tudjuk-e használni úgy, hogy azok jó közelítést adjanak. A különbség a korábbiakhoz képest az, hogy a késleltetés miatt nem csak az adott t időponttól fog függeni az ismeretlen függvény deriváltja, hanem $t - \tau$ időponttól is. Ekkor τ a késleltetés mértékét jelöli.

3.2.1. Az explicit Euler-módszer

A (3) késleltetett differenciálegyenlet esetén az explicit Euler-módszer alakja a következő lesz:

$$y_{k+1} = y_k + h \cdot f(t_k, y_{k-\bar{k}}), \quad k = 1, 2, \ldots$$

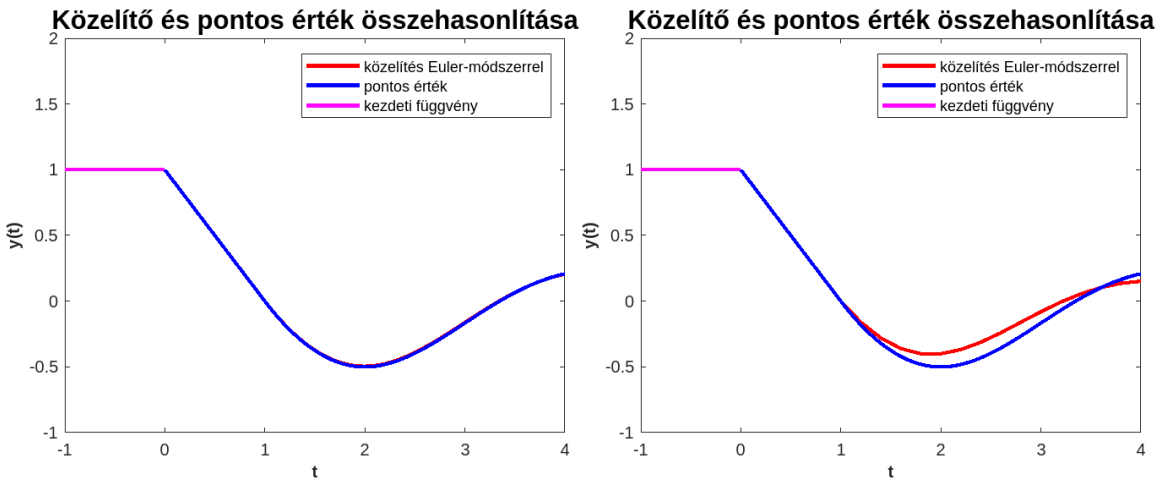
ahol $h > 0$ lépésköz mellett $\bar{k}$ olyan, hogy

$$\tau = \bar{k} \cdot h.$$

Ekkor a 2.1. példa esetén az $f(t_k, y_{k-\bar{k}})$ úgy néz ki, hogy:

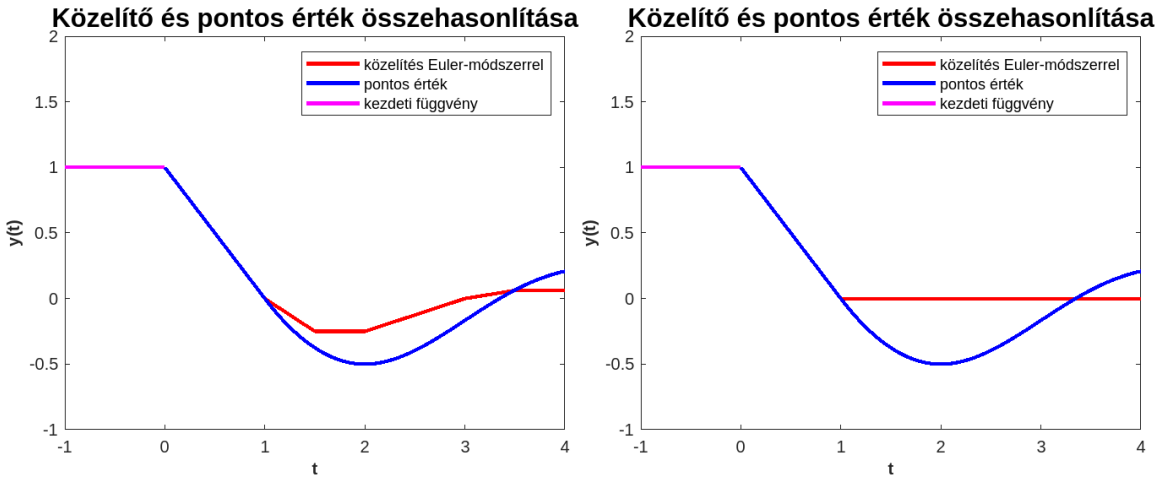
$$f(t_k, y_{k-\bar{k}}) = -y_{k-\bar{k}}$$

A jelenlegi példában $\tau = 1$. A késleltetés miatt szükségünk van még egy segédfüggvényre is, mivel jelen esetben az f függvénynél nem az y_k, hanem az $y_{k-\bar{k}}$ helyen felvett értékét használjuk a közelítéshez. Ehhez az előző $\bar{k}$ értéket el kell tárolnunk egy másik vekotorban.



(a) $h = 0,01$ esetén

(b) $h = 0,2$ esetén



(c) $h = 0,5$ esetén

(d) $h = 1$ esetén

6. ábra: Euler-módszer által számolt megoldások különböző h értékekre

A 6. ábrán ismét megtekinthetjük a megoldást különböző h értékek esetén. Láthatjuk, hogy a helyes h megválasztása nagyon fontos, mert csak úgy kapunk pontos közelítést, ha elég kis lépésközt használunk. Megfigyelhetjük, hogy minél bonyolultabb a függvényünk, annál fontosabb, hogy h elég kicsi legyen, mivel a bonyolult függvények gyakrabban változtatnak meredekséget.

3.2.2. A Runge–Kutta-módszer

A (3) késleltetett differenciálegyenlet esetén a módszer úgy fog változni a korábbiakhoz képest, hogy

$$y_{k+1} = y_k + \frac{h}{6} \cdot (F_1 + 2 \cdot F_2 + 2 \cdot F_3 + F_4), \quad k = 1, 2, \ldots$$

ahol továbbra is $h > 0$ a lépésköz és

$$\begin{aligned} F_1 &= f(t_{k-\bar{k}}, y_{k-\bar{k}}) \\ F_2 &= f(t_{k-\bar{k}} + \tfrac{h}{2}, y_{k-\bar{k}} + \tfrac{h}{2} \cdot F_1) \\ F_3 &= f(t_{k-\bar{k}} + \tfrac{h}{2}, y_{k-\bar{k}} + \tfrac{h}{2} \cdot F_2) \\ F_4 &= f(t_{k-\bar{k}} + h, y_{k-\bar{k}} + h \cdot F_3). \end{aligned}$$

Mivel a 2.1. feladatban is

$$f(x) = -x,$$

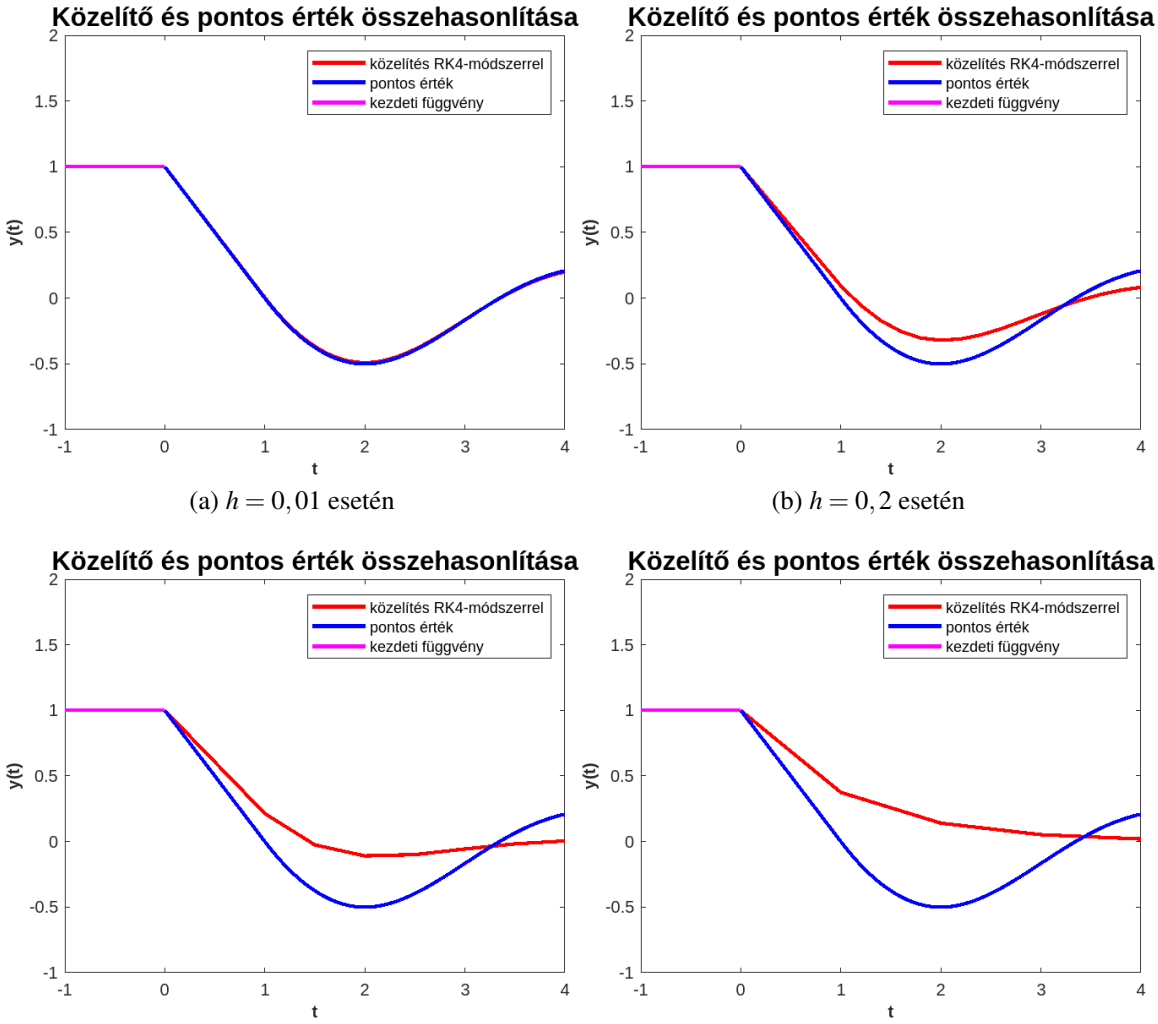
és Euler-módszerhez hasonlóan itt is teljesül, hogy:

$$\tau = \bar{k} \cdot h.$$

ezért hasonlóan az előzőekhez:

$$\begin{aligned} F_1 &= -y_{k-\bar{k}} \\ F_2 &= -(y_{k-\bar{k}} + \tfrac{h}{2} \cdot F_1) \\ F_3 &= -(y_{k-\bar{k}} + \tfrac{h}{2} \cdot F_2) \\ F_4 &= -(y_{k-\bar{k}} + h \cdot F_3). \end{aligned}$$

A 7. ábrán megfigyelhetjük, hogy nagy $h > 0$-k esetén a Runge–Kutta-közelítés látszólag rosszabb eredményt ad, mint az Euler-módszer. A késleltetést nem tartalmazó példánkban ez éppen fordítva történt. Elegendően kis $h > 0$ esetén az ábrából még nem tudunk következtetéseket levonni, ehhez szükségünk lesz az abszolút hiba kiszémolására.



(a) $h = 0,01$ esetén

(b) $h = 0,2$ esetén

(c) $h = 0,5$ esetén

(d) $h = 1$ esetén

7. ábra: Runge–Kutta-módszer által számolt megoldások különböző h értékekre

3.3. Példák és módszerek összehasonlítása

3.3.1. Az 1.7. példa és a 2.1. példa összehasonlítása

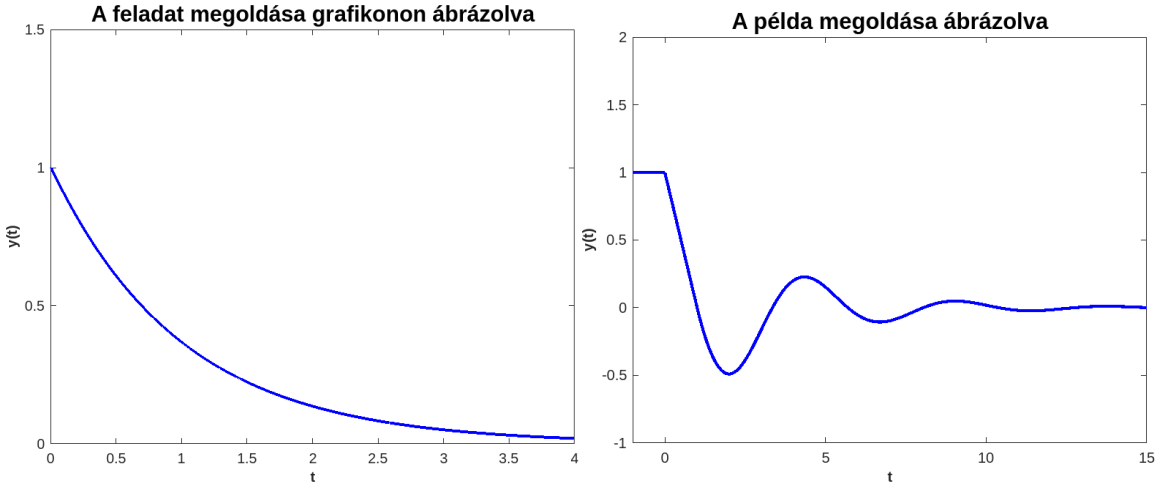
Könnyen látható, hogy az 1.7. példa és a 2.1. példa igen hasonlóak egymáshoz. A két példa

$$y'(t) = -y(t)$$
$$y'(t) = -y(t-1)$$

felépítése ugyanaz és kezdeti függvényük is megegyezik. Emellett mindkét megoldás függvény a nullához konvergál.

A különbség a kettő között az, hogy míg az 1.7. példában késleltetést nem tartalmazó közönséges differenciálegyenletről beszélünk, ahol a függvényünk a kezdetiértéktől és egy t időponttól függ, addig a 2.1. példában a késleltetést tartalmazó függvényünk a kezdeti függvénytől, és a $t-1$ időponttól függ. Ez kezdetben egy kis változtatásnak tűnhet, de a 8. ábrán láthatjuk, hogy

mekkora változást okoz a késleltetés még két ilyen egyszerű függvény esetén is. A másik nagyon szembetűnő különbség az, hogy míg az 1.7. példa esetén a kapott függvény sima, addig a 2.1. példa esetén nem az. A késleltetés miatt a függvényben töréspontok találhatók, és a függvény simasága a t-vel együtt szakaszonként nő. Így tehát $t = 0$ esetén a függvény megoldás nem deriválható, $t = 1$ esetén egyszer, $t = 2$ esetén kétszer stb. . . deriválható.



(a) Az 1.7. példa megoldása

(b) A 2.1. példa megoldása

8. ábra: Példák megoldásainak összehasonlítása

3.3.2. Az 1.7. példa hibafüggvénye

Ebben az alfejezetben az 1.7. példára korábban bemutatott közelítések hibáit vizsgáljuk meg. A hibafüggvények kiszámolására használt MATLAB kód a függelékben tekinthető meg.

A két közelítés összehasonlítására abszolút hibát mérünk a következő módon:

$$hiba_{E,k} = y_{E,k} - y_{pontos,k}$$
$$hiba_{RK,k} = y_{RK,k} - y_{pontos,k}$$

amelyeket úgy számolunk ki, hogy:

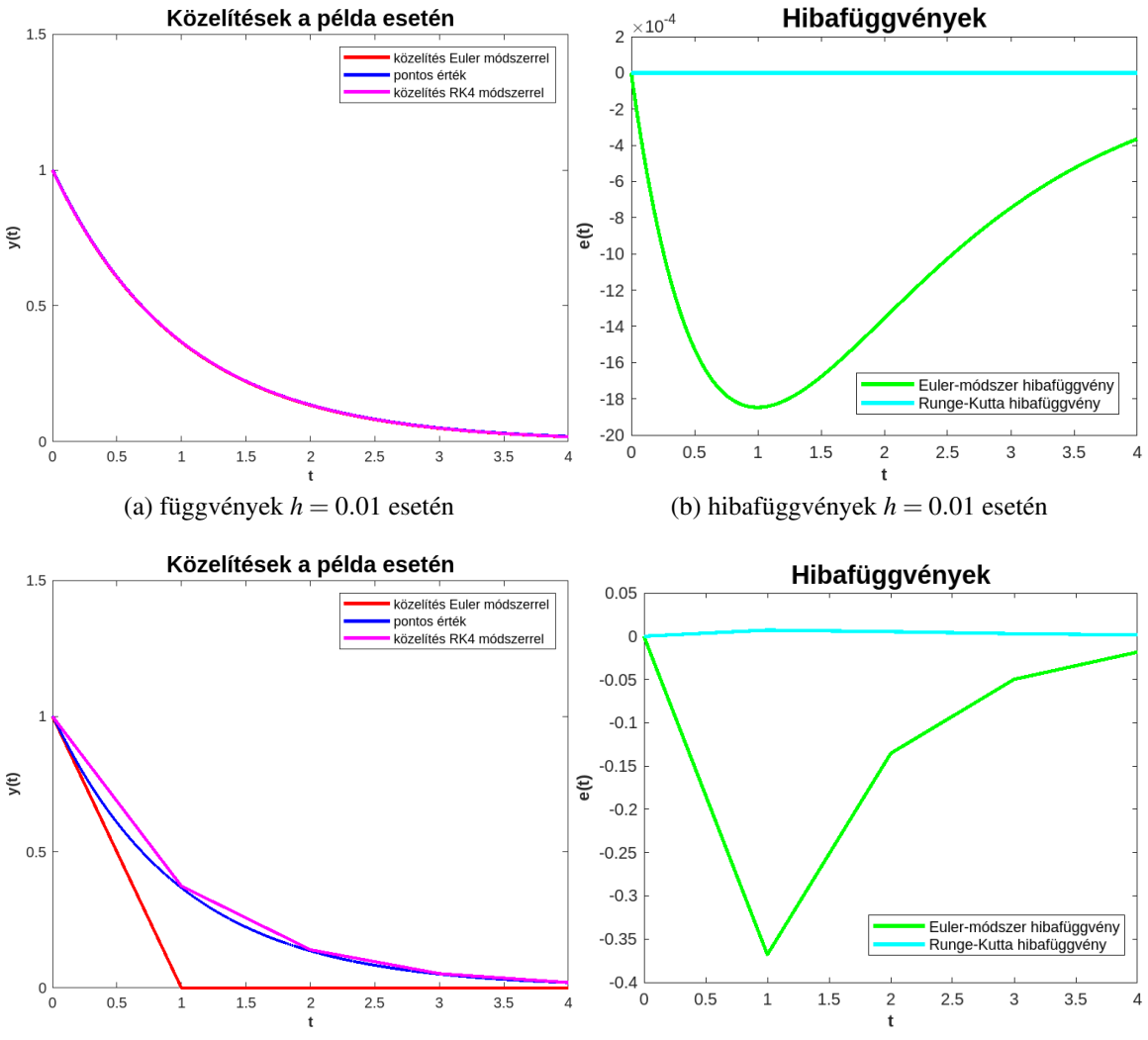
$$\begin{aligned} hiba_{E,k}(1) &= y_{E,k}(1) - y_{pontos,k}(1) \\ hiba_{E,k}(2) &= y_{E,k}(2) - y_{pontos,k}(2) \\ hiba_{E,k}(3) &= y_{E,k}(3) - y_{pontos,k}(3) \\ &\vdots \\ hiba_{E,k}(n) &= y_{E,k}(n) - y_{pontos,k}(n) \end{aligned}$$

és

$$\begin{aligned} hiba_{RK,k}(1) &= y_{RK,k}(1) - y_{pontos,k}(1) \\ hiba_{RK,k}(2) &= y_{RK,k}(2) - y_{pontos,k}(2) \\ hiba_{RK,k}(3) &= y_{RK,k}(3) - y_{pontos,k}(3) \\ &\vdots \\ hiba_{RK,k}(n) &= y_{RK,k}(n) - y_{pontos,k}(n) \end{aligned}$$

A két hibafüggvényt $h = 0,01$ és $h = 1$ értékekre is megtekinthetjük a 9. ábrán. Itt láthatjuk a közelítéseket és azok hibafüggvényeit egymás mellett. Mint azt a korábbiakban is megfigyelhettük, a közelítések kisebb lépésköz, azaz h esetén egyre pontosabb eredményt adnak. Ennek következtében a hibájuk is kisebb lesz. Míg $h = 0,01$ esetén a hiba olyan kicsi, hogy az 10^{-4} nagyságrendű, addig $h = 1$ esetén az Euler-módszer hibája akár $0,35$ is lehet bizonyos értékekre.

Megfigyelhető az is, hogy a feladatokra a Runge–Kutta-módszer sokkal jobb közelítést ad mindkét h esetén. Ebből kifolyólag az 1.7. példamegoldására ezt a közelítést lesz célszerű használni, mert így pontosabb eredményt kaphatunk.



(a) függvények $h = 0.01$ esetén

(b) hibafüggvények $h = 0.01$ esetén

(c) függvények $h = 1$ esetén

(d) hibafüggvények $h = 1$ esetén

9. ábra: Euler-módszer által számolt közelítések hibafüggvényei az 1.7. példa esetén

3.3.3. A 2.1. példa hibafüggvénye

A 2.1. példa esetén a hibafüggvényeket ismét a korábban már levezetett

$$hiba_{E,k} = y_{E,k} - y_{pontos,k}$$
$$hiba_{RK,k} = y_{RK,k} - y_{pontos,k}$$

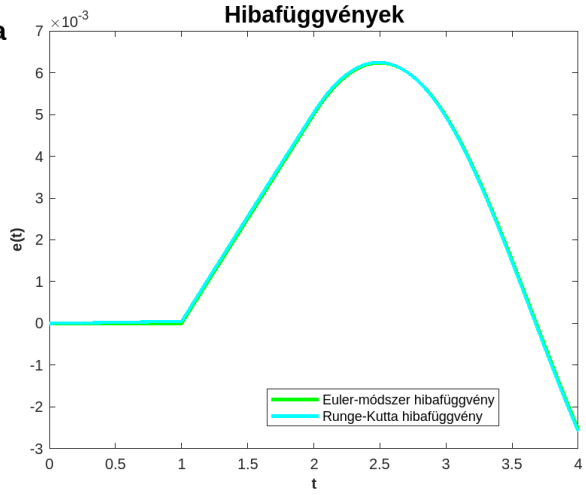
módon számítjuk ki.

A hibát ismét $h = 0,01$ és $h = 1$ esetén nézzük meg. A 10. ábrán láthatjuk, hogy a késleltetést tartalmazó differenciálegyenlet esetében is nagyon jó közelítéseket kapunk az általunk használt módszerek által. Ebből arra következtethetünk, hogy mikor a függvényünk késleltetést tartalmaz, akkor elég egyszerűen átalakítani úgy, hogy a késleltett értékkel számoljon és nincs szükség további változtatásokra.

Láthatjuk, hogy míg a közönséges differenciálegyenlet példánknál megfelelő h esetén a hibafüggvényünkben nem találunk töréspontokat, addig a jelenlegi feladatban minden esetben tartalmaz ilyet. Ez azért van, mert $t = 0$ időpontban már az eredeti függvényben is mindenképp található egy töréspont.

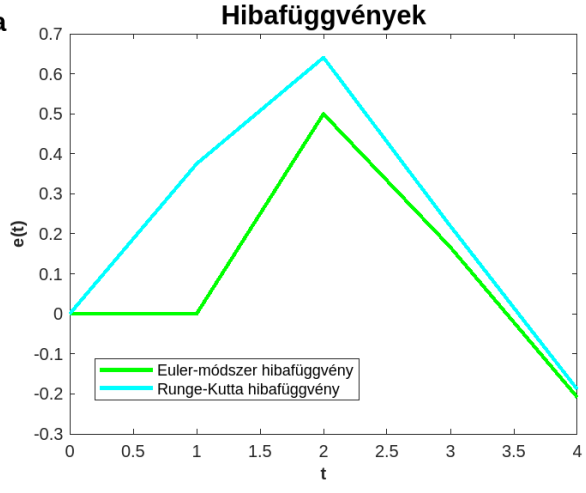
Emelett érdekes megfigyelés az is, hogy a Runge–Kutta-módszer nagy $h > 0$ esetén rosszabbul, megfelelően kis $h > 0$ esetén pedig jobban teljesít, mint az Euler-módszer.

(a) függvények $h = 0,01$ esetén



(b) hibafüggvények $h = 0,01$ esetén

(c) függvények $h = 1$ esetén



(d) hibafüggvények $h = 1$ esetén

10. ábra: Euler-módszer által számolt közelítések hibafüggvényei a 2.1. példa esetén

4. Populációs modellek

A fejezet Svantnerné Sebestyén Gabriella: Populációdinamikai modellek és matematikai vizsgálatuk [3] forrás alapján készült.

Különféle élőlények populációdinamikáját differenciálegyenletekkel tudjuk modellezni, ahol a különböző paraméterek más-más a populációra ható tényezőket írnak le.

Egy adott csoport egyedszáma rengeteg különböző tényezőtől függ, ilyen befolyásolható körülmény lehet például a jelenlegi egyedszám, a születési ráta, az élőhely eltartóképessége, különböző vírusok elterjedése, természeti katasztrófák, a vele azonos környezetben megtalálható más csoportok egyedszáma és tulajdonképpen bármi, ami hatással van az adott csoportra.

4.1. Verhulst-féle modell

A jelenlegi fejezetben egy olyan populációs modellt fogunk megtekinteni késleltetés nélkül és késleltetéssel is, amelyet a populáció mérete, a születési ráta és az élőkörnyezet eltartóképessége befolyásol. Ebben a fejezetben a közelítéshez az Euler-módszerrel számolunk.

Ezt a populációs modellt a logisztikai növekedés, vagy Verhulst-féle modellnek nevezzük. A modell során figyelembe vesszük azt, hogy a populáció véges eltartóképességgel rendelkezik, emelett a növekedés zárt környezetben zajlik.

A következőkben 4, a fenti paraméterekkel rendelkező pupulációs modellt fogunk megtekinteni, ezek közül 1 nem tartalmaz késleltetés, a másik 3 pedig különböző módokon tartalmaz. A 3 késleltetést tartalmazó modell közötti különbség az, hogy az adott modell melyik változója tartalmaz késleltetést.

A populációs modell ezen paraméterek esetén késleltetés nélkül így fog kinézni:

4.1. Példa. Tekintsük

$$\begin{aligned} y'(t) &= a \cdot y(t) \cdot (K - y(t)), \\ y(0) &= \alpha, \end{aligned}$$

ahol $y(t)$ a populáció mérete, $a > 0$ a születési ráta és $K > 0$ a terület eltartóképessége. Ez a modell nem realisztikus, hiszen nem veszi figyelembe azt, hogy a szaporodásnak és az élőterület kihasználásának is vannak időbeli késleltetései, mivel a legtöbb populációnak az új egyedek születéséhez és a táplálék feldolgozásához is időre van szüksége.

Erre az egyenletre az Euler-módszerrel való közelítés képlete:

$$\begin{aligned} y_{k+1} &= y_k + h \cdot f(y_k, y_{k-\bar{k}}), \quad k = 1, 2, \dots \\ f(y_k, y_{k-\bar{k}}) &= a \cdot y_k \cdot (K - y_k), \end{aligned}$$

ahol $h > 0$ lépésköz.

A három késleltetést tartalmazó modell pedig:

4.2. Példa. Tekintsük

$$y'(t) = a \cdot y(t) \cdot (K - y(t-\tau)), \quad t > 0$$
$$y(t) = \varphi(t) = 1; \quad t \in [-1,0]$$

ahol $\tau = 1$ a késleltetés mértéke. Ebben az változatban a táplálék feldolgozási idő kapcsán alkalmazunk késleltetést, így figyelembe véve azt, hogy a táplálék feldolgozásához időre van szükség, ezért a terület eltartóképességéből, nem $y(t)$, hanem $y(t-\tau)$ értékét vontuk ki.

Erre az egyenletekre az Euler-módszer a következőképpen fog kinézni:

$$y_{k+1} = y_k + h \cdot f(y_k, y_{k-\bar{k}}), \quad k = 1,2,\ldots$$
$$f(y_k, y_{k-\bar{k}}) = a \cdot y_k \cdot (K - y_{k-\bar{k}}).$$

Ahol $h > 0$ lépésköz és mellette $\bar{k}$ olyan, hogy

$$\tau = \bar{k} \cdot h.$$

4.3. Példa. Tekintsük

$$y'(t) = a \cdot y(t-\tau) \cdot (K - y(t)), \quad t > 0$$
$$y(t) = \varphi(t) = 1; \quad t \in [-1,0]$$

ahol $\tau = 1$ a késleltetés mértéke. Ebben a változatban azért alkalmazunk kséleltetést, mert a szaporodás mértéke függ a terhességi/vemhességi időtől.

Erre az egyenletekre az Euler-módszer a következőképpen fog kinézni:

$$y_{k+1} = y_{k-\bar{k}} + h \cdot f(y_k, y_{k-\bar{k}}), \quad k = 1,2,\ldots$$
$$f(y_k, y_{k-\bar{k}}) = a \cdot y_{k-\bar{k}} \cdot (K - y_k).$$

Ahol $h > 0$ lépésköz és mellette $\bar{k}$ olyan, hogy

$$\tau = \bar{k} \cdot h.$$

4.4. Példa. Tekintsük

$$y'(t) = a \cdot y(t-\tau) \cdot (K - y(t-\tau)), \quad t > 0$$
$$y(t) = \varphi(t) = 1; \quad t \in [-1,0]$$

ahol $\tau = 1$ a késleltetés mértéke. Ebben a változatban mindkét korábbi késleltetést egyszerre alaklmazzuk, ugyanakkora τ értékkel. A négy egyenlet közül ezzel kaphatjuk a legrealisztikusabb képet. Konkrét populációra nézve az egyenletet úgy kaphatunk valósághű képet, ha két különböző τ értékkel számolunk, hiszen a két változónak az eltolodása nem feltétlen azonos.

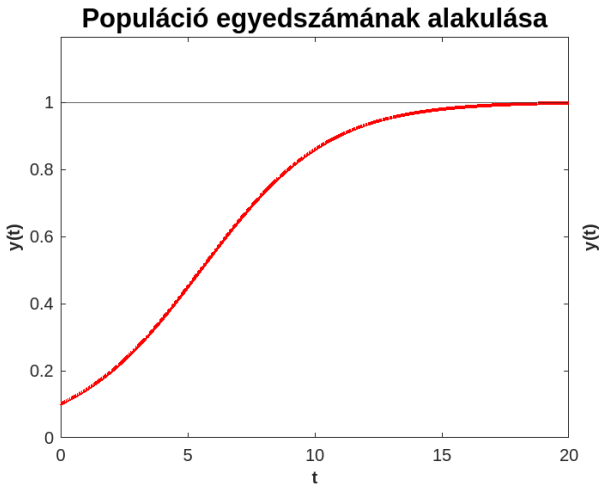
Erre az egyenletekre az Euler-módszer a következőképpen fog kinézni:

$$y_{k+1} = y_{k-\bar{k}} + h \cdot f(y_k, y_{k-\bar{k}}), \quad k = 1,2,\ldots$$
$$f(y_k, y_{k-\bar{k}}) = a \cdot y_{k-\bar{k}} \cdot (K - y_{k-\bar{k}}).$$

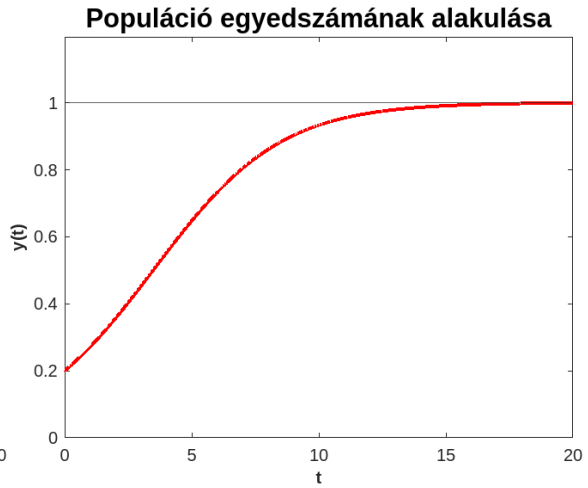
Ahol $h > 0$ lépésköz és mellette $\bar{k}$ olyan, hogy

$$\tau = \bar{k} \cdot h.$$

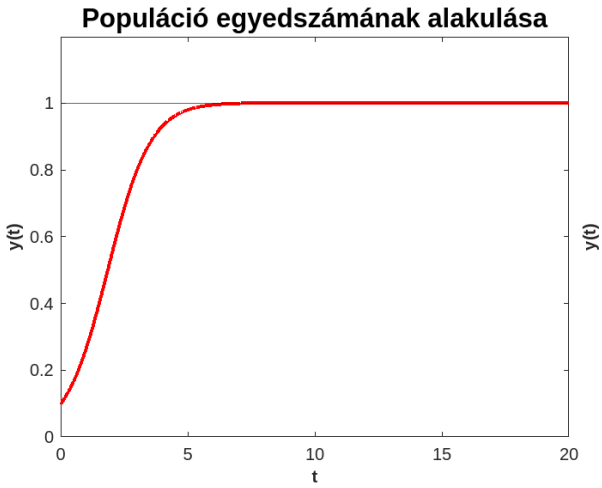
A 11. ábrán láthatjuk a 4.1. populációs modell megoldását különböző paraméterek esetén. Láthatjuk, hogy a görbe α változtatásával egybevágó marad; olyan, mintha az x-tengelyen eltolnánk a függvényt a késleltetés, azaz $\tau = 1$ értékkel. A születési ráta változtatásával láthajuk, hogy minél nagyobb az érték, annál gyorsabban érjük el a maximum K értéket.



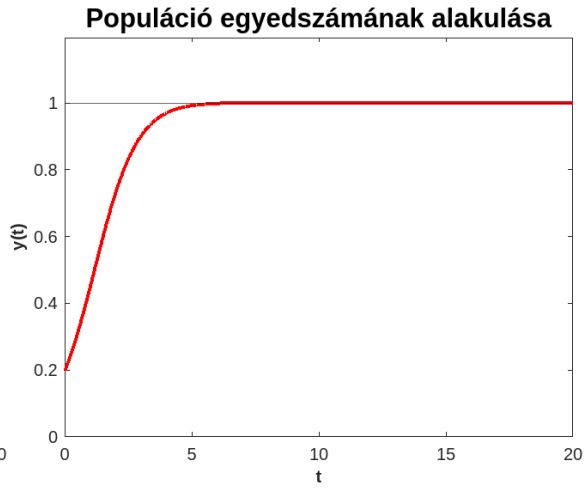
(a) $y_0 = 0,1; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(b) $y_0 = 0,2; t = 20; h = 0,01; a = 0,4; K = 1$ esetén

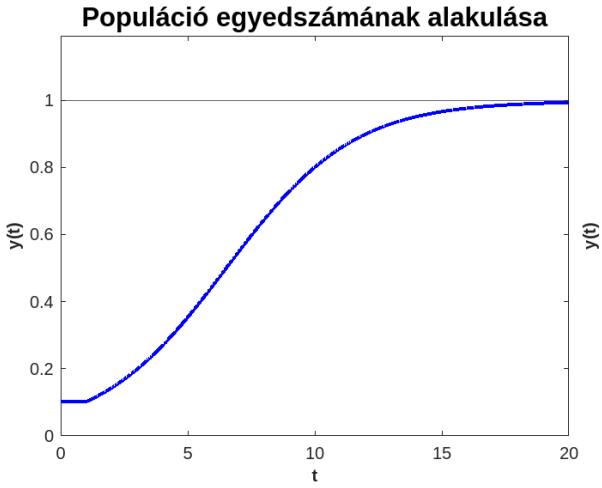


(c) $y_0 = 0,1; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

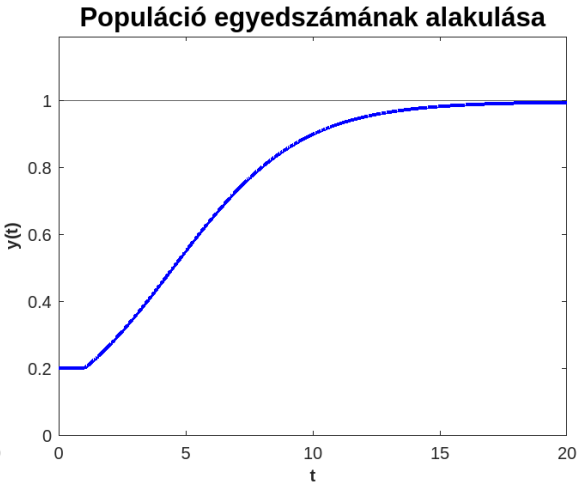


(d) $y_0 = 0,2; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

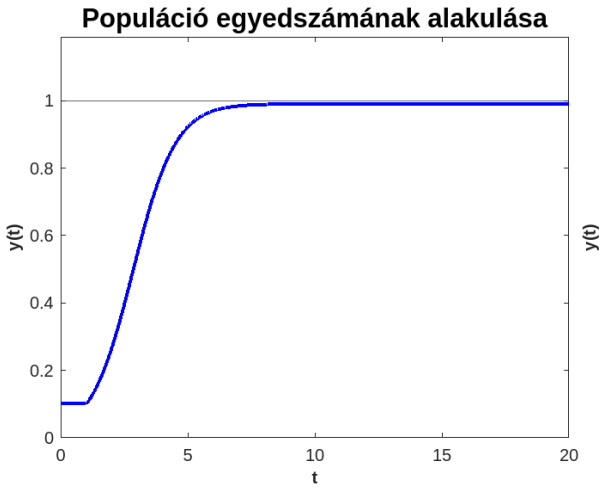
11. ábra: A 4.1. populációs modell megoldása különböző paraméterekre



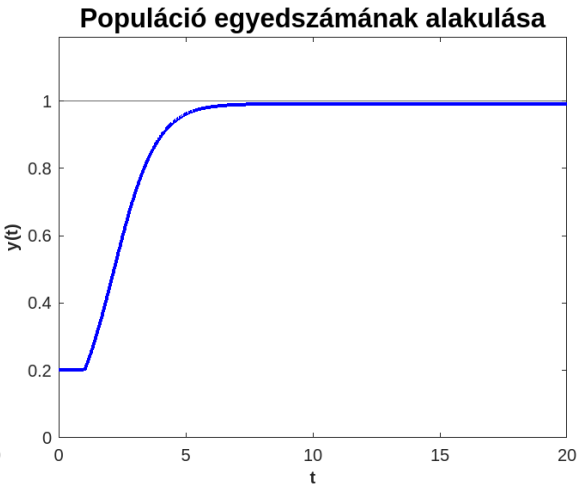
(a) $y_0 = 0,1; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(b) $y_0 = 0,2; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(c) $y_0 = 0,1; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

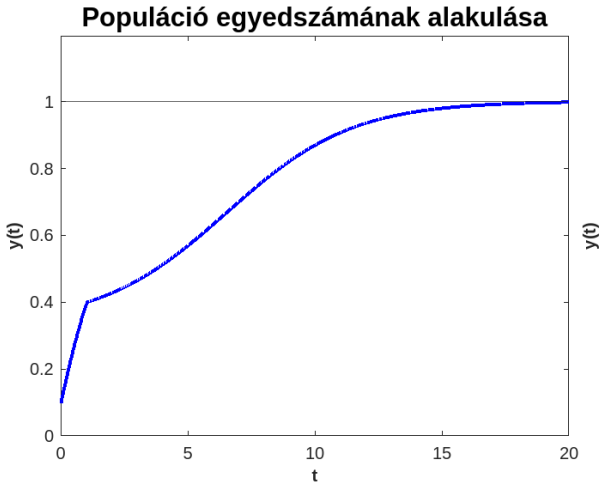


(d) $y_0 = 0,2; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

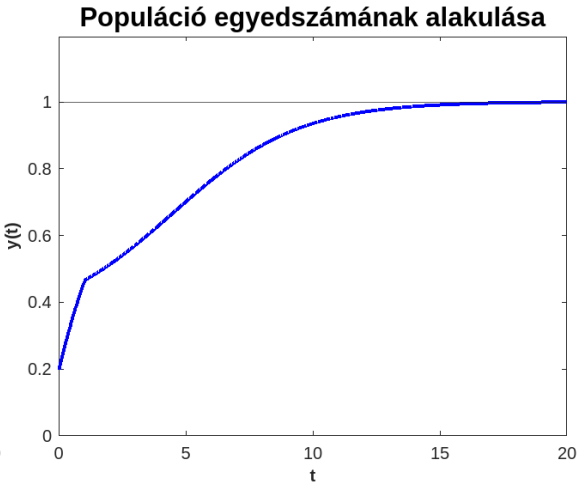
12. ábra: A 4.2. populációs modell megoldása különböző paraméterekre

A 4.2. modell esetén a 12. ábrán láthatjuk, hogy ebben az esetben a megoldás nem változik sokat a 4.1. modellhez képest. A különbség az, hogy itt a késeltetés miatt a $[0,1]$ intervallumon megjelenik a függvényben egy új szakasz, majd a töréspontot követően a modell megoldása előző esetekkel egybevágó, így tehát a megoldás t szerint mindig el lesz tolva 1-el.

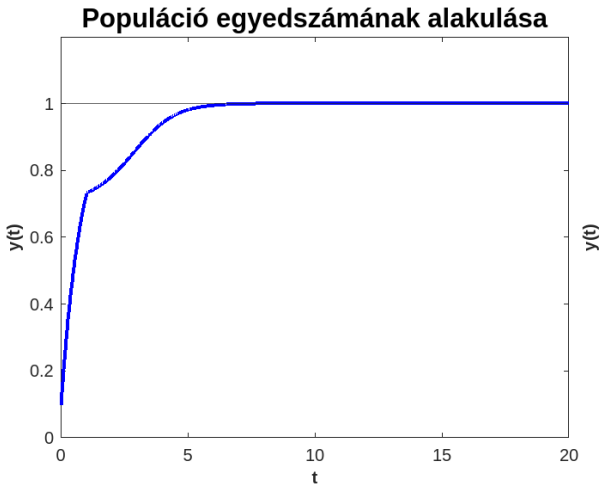
A 13. ábrán igazán érdekes függvényeket láthatunk. Érdekes megfigyelés, hogy a populáció méretét mennyire befolyásolja a függvény ilyen módon történő késleltetése. A megoldás a töréspont után hasonlít a korábbiakhoz, de egyik szakaszban sem veszi fel a 4.1. modell megoldásának alakját.



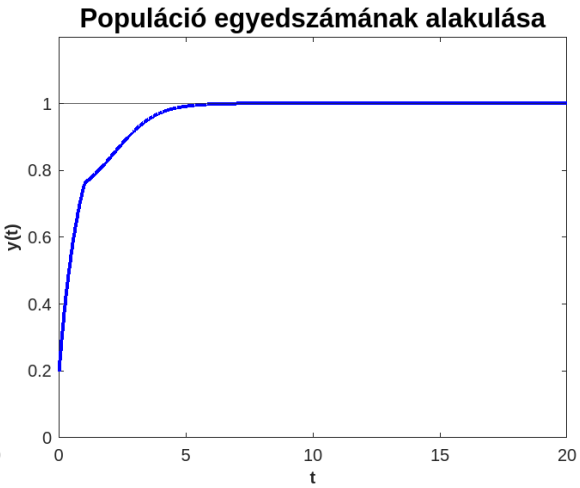
(a) $y_0 = 0,1; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(b) $y_0 = 0,2; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



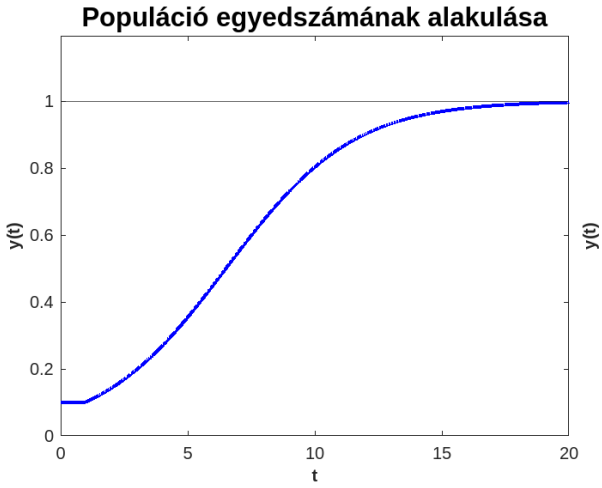
(c) $y_0 = 0,1; t = 20; h = 0,01; a = 1,2; K = 1$ esetén



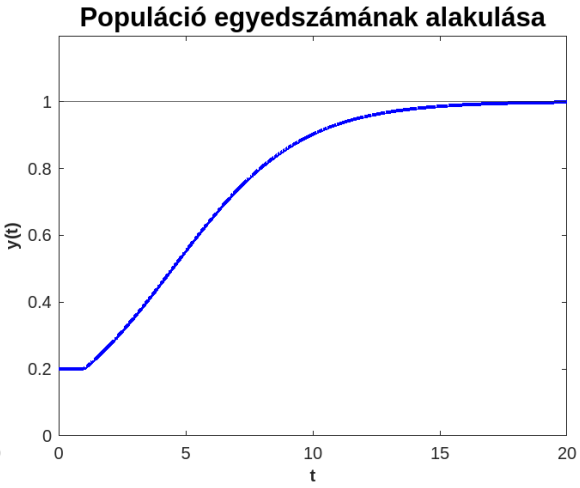
(d) $y_0 = 0,2; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

13. ábra: A 4.3. populációs modellek különböző paraméterekre

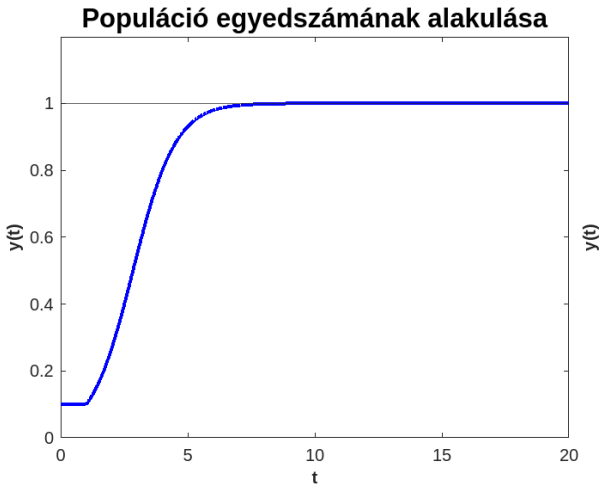
A 14. ábrán láthatjuk azt a modellt, amleyben a függvényünk csak a τ-val ezelőtti időpont beli értéktől függ. Itt tehát figyelembe vesszük mindkét korábban említett paraméter időbeli eltolódását. A mindkét helyen késleltetést tartalmazó modell jobban hasonlít a 4.1. és a 4.2. modellekre, mint a 4.3. modellre.



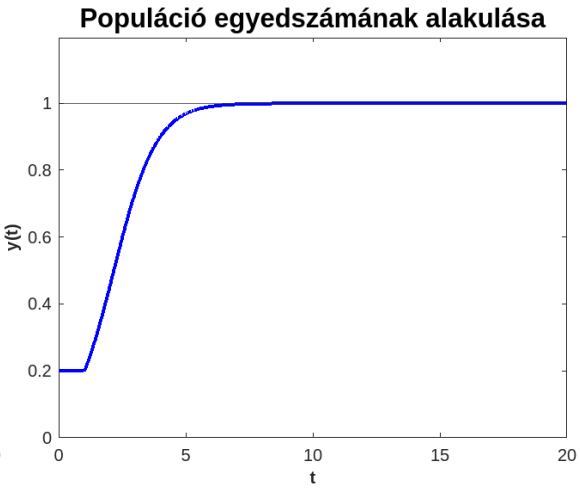
(a) $y_0 = 0,1; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(b) $y_0 = 0,2; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(c) $y_0 = 0,1; t = 20; h = 0,01; a = 1,2; K = 1$ esetén



(d) $y_0 = 0,2; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

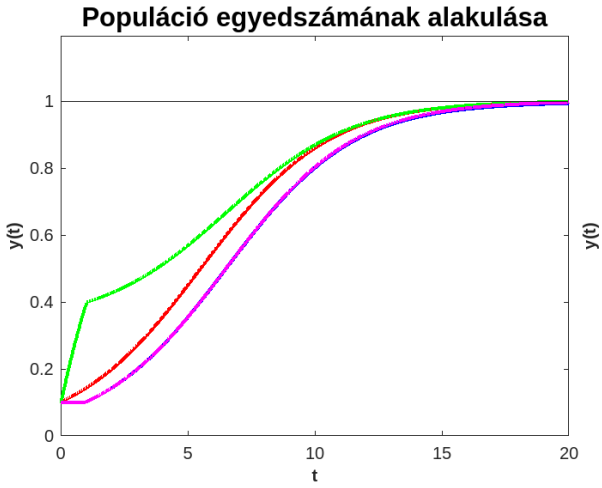
14. ábra: A 4.4. populációs modell megoldásai különböző paraméterekre

A 15. ábrán megtekinthetjük, hogy a különböző példák megoldásai, hogy néznek ki azonos paraméterek esetén egymáshoz képest. Zölddel láthatjuk a 4.3., pirossal a 4.1., kékkel a 4.2. és rózsaszínnel a 4.4. modellt.

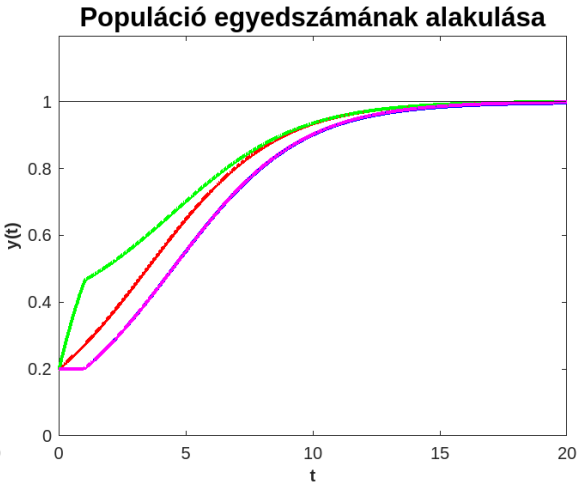
Megfigyelhetjük, hogy a kisebb modellbeli változatások ellenére az összes függvény körülbelül ugyanabban a t időpontban éri el a maximális egyedszámot. Abban viszont, hogy a függvény hogy viselkednek ezelőtt a t időpont előtt, vannak különbségek.

A késleltetést tartalmazó változatokban minden esetben található egy töréspont $t = 1$ időpontban, ez $\tau = 1$ miatt van így. A függvényeink a töréspontok után mind simák.

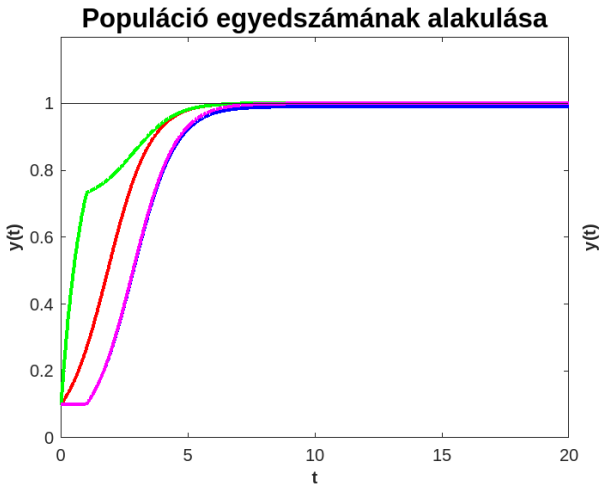
Érdekes, hogy a maximális egyedszámot minden esetben a 4.3. késleltetést tartalmazó modell éri el, hiszen azt várnánk, hogy a késleltetést tartalmazó modelleknek az eredménye később éri el a maximumot a késleltetést nem tartalmazó modellekhez képest, mert növekedő egyedszámú modellek esetén a korábbi értékekből való számolás általában kisebb értékekkel való számolást jelent.



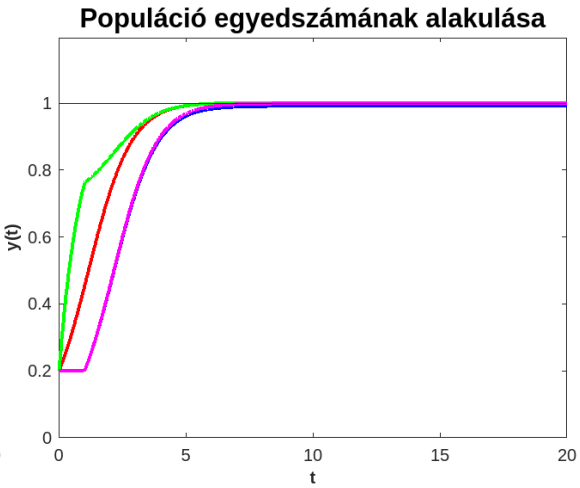
(a) $y_0 = 0,1; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(b) $y_0 = 0,2; t = 20; h = 0,01; a = 0,4; K = 1$ esetén



(c) $y_0 = 0,1; t = 20; h = 0,01; a = 1,2; K = 1$ esetén



(d) $y_0 = 0,2; t = 20; h = 0,01; a = 1,2; K = 1$ esetén

15. ábra: A 4 példa megoldásainak összehasonlítása azonos paraméterek esetén

4.2. Lotka–Volterra-féle zsákmány-ragadozó-modell

A Lotka–Volterra-féle zsákmány-ragadozó-modell megoldásához nem csak szimpla differenciálegyenletekre, hanem differenciálegyenlet rendszerekre van szükség. Ez azért van, mert ebben az esetben két populáció egymásra gyakorolt hatását vizsgáljuk, így két összefüggő változónk van. Ebben a fejezetben a megoldások közelítését Runge–Kutta-módszerrel számoljuk ki.

Először nézzük meg a populációs modellt késleltetést nem tartalmazó differenciálegyenlet esetén.

4.5. Példa. Keressük annak az egyenletrendszernek a megoldását, ahol:

$$\begin{aligned} u'(t) &= g(t,u(t),v(t)) \\ v'(t) &= l(t,u(t),v(t)) \\ g(t,u(t),v(t)) &= a\cdot u(t) - b\cdot u(t)\cdot v(t) \\ l(t,u(t),v(t)) &= c\cdot u(t)\cdot v(t) - c\cdot v(t) \\ u(0) &= u_0 \\ v(0) &= v_0, \\ a,b,c,d &> 0 \\ u,v &: \mathbb{R}\to\mathbb{R} \\ g,l &: \mathbb{R}\times\mathbb{R}^2\to\mathbb{R}. \end{aligned}$$

A 4.5. feladat numerikus megoldásához vezessük be az alábbiakat:

$$y(t) := \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}, \quad y : \mathbb{R}\to\mathbb{R}^2$$

$$f(t,y) := \begin{pmatrix} g(t,u,v) \\ l(t,u,v) \end{pmatrix}, \quad y : \mathbb{R}\times\mathbb{R}^2\to\mathbb{R}^2$$

$$y(0) = \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} \in \mathbb{R}.$$

Ekkor a 4.5. példa átírható az alábbi alakba:

$$\begin{aligned} y'(t) &= f(t,y(t)) \\ y(0) &= y_0. \end{aligned}$$

Innentől könnyű dolgunk van, hiszen az előző fejezetkben már található numerikus módszer az ilyen alakú egyenletekre. A különbség csupán annyi, hogy y_n jelen esetben egy vektor.

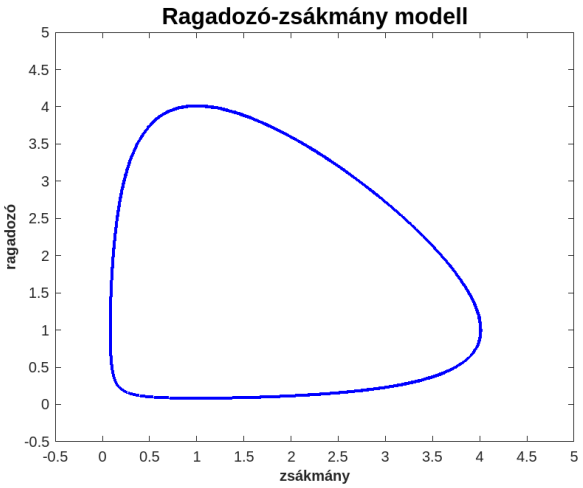
A feladatra felírva a Runge–Kutta-módszert:

$$y_{k+1} = y_k + \tfrac{h}{6}\cdot(F_1 + 2\cdot F_2 + 2\cdot F_3 + F_4), \quad k = 1,2,\ldots$$

ahol $h > 0$ a lépésköz és

$$
\begin{aligned}
F_1 &= f(t_k, y_k) \\
F_2 &= f(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} \cdot F_1) \\
F_3 &= f(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2} \cdot F_2) \\
F_4 &= f(t_k + h, y_k + h \cdot F_3)
\end{aligned}
$$

megkapjuk a 4.5. példa megoldását, amelyet a 16. ábrán láthatunk. Az ábra a ragadozó és a zsákmány közötti viszonyt mutatja meg. Az idő előrehaladtával nem térünk le a már ismert göbéről, ugyanis a megoldás periodikus és mindig megvan az egyensúly a ragadozó és a zsákmány között.



(a) $y_0 = 0,2; t = 50; h = 0,01$ esetén

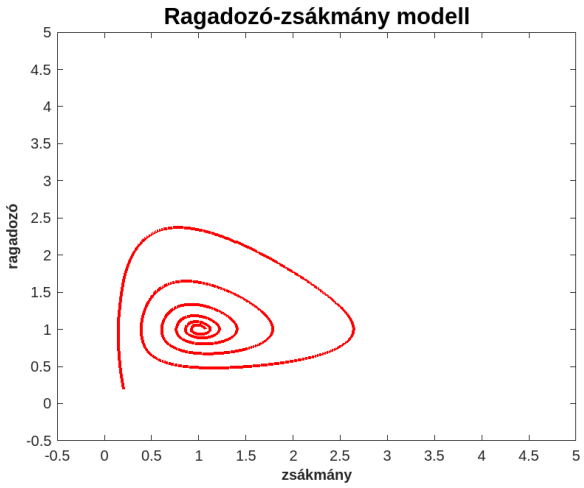
16. ábra: A 4.5. feladat megoldása fázistérben ábrázolva

Azért, hogy a modell pontosabb legyen, érdemes bizonyos részeit késleltetéssel leírni. A következőkben tekintsünk egy olyan modellt, ahol figyelembe vesszük a zsákmány populáció szaporodási idejét is.

4.6. Példa. Keressük annak az egyenletrendszernek a megoldását, ahol:

$$
\begin{aligned}
u'(t) &= g(t, u(t), v(t)) \\
v'(t) &= l(t, u(t), v(t)) \\
g(t, u(t), v(t), u(t-1)) &= a \cdot u(t-1) - b \cdot u(t) \cdot v(t) \\
l(t, u(t), v(t)) &= c \cdot u(t) \cdot v(t) - c \cdot v(t) \\
u(0) &= u_0 \\
v(0) &= v_0,
\end{aligned}
$$

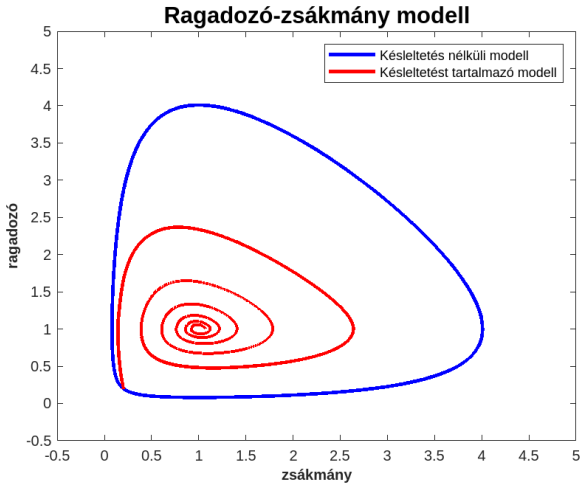
A fentebb már bemutatott módon erre a feladatra is alkalmazzuk a Runge–Kutta-módszert. A 17. ábrán láthatjuk a 4.6. példa megoldását. Láthatjuk, hogy a késleltetés miatt már nem alakul ki periodikus megoldás, mint a korábbi esetben.



(a) $y_0 = 0,2; t = 50; h = 0,01$ esetén

17. ábra: A 4.6. feladat megoldása fázistérben ábrázolva

Egy ábrára helyezve a két modell megoldását, megtekinthetjük, hogy azonos paraméterekkel mennyire különöbznek egymástól. A 18. ábrán láthatjuk, hogy a két modell igencsak eltérő értékeket ad.



(a) $y_0 = 0,2; t = 50; h = 0,01$ esetén

18. ábra: A két modell megoldása fázistérben ábrázolva

Összességében elmondhatjuk, hogy a késleltetést tartalmazó differenciálegyenletek fontos részei a modellalkotásnak és semmiképp sem szabad megfeledkezni róluk, mivel segítségükkel a környezetünket sokkal pontosabban tudjuk modellezni.

5. Összefoglalás

A dolgozatban arra a kérdésre kerestem a választ, hogy a késleltetést tartalmazó differenciálegyenletek esetén hogyan változnak a numerikus megoldási módszerek.

Ehhez először bevezettem az alapfogalmakat, majd egyszerű példák esetén megnéztem a különbségeket a késleltetést tartalmazó és a késleltetést nem tartalmazó differenciálegyenletek között. A dolgozatban az Euler- és a Runge–Kutta-módszereket használtam a függvények közelítésére. Az egyik példa esetén a pontos értékét is kiszámoltam, hogy a későbbiekben mérhessem a közelítési módszerek hibáját.

Késleltetés esetén a módszereken annyit változtattam, hogy a $y(t)$ helyett $y(t-\tau)$-val számoltam. Ez az átalakítás pozitív eredményeket adott, ugyanis a hibafüggvények kiszámolása és ábrázolása során arra a következtetésre jutottam, hogy elégséges ezt a két numerikus módszert a fenti módon átírni, hogy jó közelítéseket kapjunk.

A dolgozat végén bemutattam és numerikusan megoldottam két populációs modellt késleltetés nélkül és késleltetés esetén is. Ezekből az egyik egy bonyolultabb két ismeretlenes egyenletrendszert tartalmazott.

A dolgozatban található ábrákat saját kódokkal készítettem, amelyek a függelékben találhatók.

6. Függelék - MATLAB kódok és leírásuk

A függelékben megtalálhatóak azok a MATLAB kódot, amelyekkel a közelítéseket, a hibafüggvényeket és az ezeket bemutató ábrákat készítettem. A dolgozathoz sok különböző kódot készítettem, az itt megjelenített kódok kommentek ki és be kapcsolásával az összes elkészített kódot lefedik.

6.1. Közönséges példa

A kódban két függvényt találhatunk, az első kiszámolja az 1.7. példa esetén a pontos függvényt, annak Euler- és a Runge–Kutta-módszerekkel való közelítését, és ezek hibafüggvényeit.

A hibafüggvényt két konkrét h esetre írtam fel, ez azért van, mert azt a „problémát”, hogy közelítések esetén $t = 0$-tól $t = 4$-ig kevesebb adatom van, mint a pontos függvény esetén – így a vektoraim nem egyforma hosszúak – úgy hidaltam át, hogy a vektorhosszúság különbségekre számoltam ki egyéni hibaszámítási módszereket.

A második függvény a közelítésekhez szükséges segédfüggvényt tartalmazza.

A kód:

```
%Kozonseges differencialegyenlet
function E = hiba_sima(x0,t,h)

    x(1) = x0;
    z(1) = x0;
    b(1) = 0;
    a(1) = 0;

    N   = floor(t/h)+1;
    tex = linspace(0,t,4001);
    e   =  x0*exp(-tex);

    for i = 2:N
        x(i) = x(i-1) + h * fgv_sima(x(i-1));

        k_1 = h * fgv_sima(z(i-1));
        k_2 = h * fgv_sima(z(i-1)+k_1*h/2);
        k_3 = h * fgv_sima(z(i-1)+k_2*h/2);
        k_4 = h * fgv_sima(z(i-1)+k_3*h);
        z(i) = z(i-1) + (1/6)*(k_1+2*k_2+2*k_3+k_4);
    %{
        %h=0.01
        a(i) = x(i) - e((10)*(i-1)+1);
        b(i) = z(i) - e((10)*(i-1)+1);
    %}
    end
    %{
```

```
28          %h=1
29          a(2) = x(2)-e(1001);
30          a(3) = x(3)-e(2001);
31          a(4) = x(4)-e(3001);
32          a(5) = x(5)-e(4001);
33
34          b(2) = z(2)-e(1001);
35          b(3) = z(3)-e(2001);
36          b(4) = z(4)-e(3001);
37          b(5) = z(5)-e(4001);
38      %}
39  %{
40      %hibafuggvenyek
41      plot([0:h:t],a,"green",'LineWidth',2)
42      axis([0 t -20*10^-4 2*10^-4])
43      hold on
44      plot([0:h:t],b,"cyan",'LineWidth',2)
45      hold on
46
47      title('Hibafuggvenyek','fontsize',15)
48      xlabel('t','fontsize',10,'fontweight','bold')
49      ylabel('e(t)','fontsize',10,'fontweight','bold')
50      legend('Euler-modszer hibafuggveny','Runge-Kutta
           hibafuggveny')
51  %}
52  %{
53      %Kozelitesek
54      plot([0:h:t],x,"red",'LineWidth',2)
55      %axis([0 t -1 x0+0.5])
56      hold on
57      plot(tex,x0*exp(-tex),"blue",'LineWidth',2)
58      hold on
59      plot([0:h:t],z,"magenta",'LineWidth',2)
60
61      title('Kozelitesek a pelda eseten','fontsize',15)
62      xlabel('t','fontsize',10,'fontweight','bold')
63      ylabel('y(t)','fontsize',10,'fontweight','bold')
64      legend('kozelites Euler-modszerrel','pontos ertek','
           kozelites RK4-modszerrel')
65  %}
66
67  end
68
69  function dy= fgv_sima(y)
70  dy = -y;
71  end
72
73  % hiba_sima(1,4,0.01)
```

```
74 % hiba_sima(1,4,1)
```

6.2. Késleltetett példa

A kódban két függvényt találhatunk, az első kiszámolja a 2.1. példa esetén a pontos függvényt, annak Euler- és a Runge–Kutta-módszerekkel való közelítését, és ezek hibafüggvényeit.

A hibafüggvényt két konkrét h esetre írtam fel, ez azért van, mert azt a „problémát”, hogy közelítések esetén $t = 0$-tól $t = 4$-ig kevesebb adatom van, mint a pontos függvény esetén – így a vektoraim nem egyforma hosszúak – úgy hidaltam át, hogy a vektorhosszúság különbségekre számoltam ki egyéni hibaszámítási módszereket.

A második függvény a közelítésekhez szükséges segédfüggvényt tartalmazza.

A kód:

```
1  %Kesleltetest tartalmazo differencialegyenlet
2  function E = hiba_delay(x0,t,h)
3
4      N = floor(t/h+1);
5      x(1) = x0;
6      z(1) = x0;
7
8      xn = ones(1,1/h);
9      zn = ones(1,1/h);
10
11     b(1) = 0;
12     a(1) = 0;
13
14     b1  =linspace(0,1,1001);
15     b2  =linspace(1,2,1001);
16     b3  =linspace(2,3,1001);
17     b4  =linspace(3,4,1001);
18     e1 =  -(b1-1);
19     e2 =  (((b2-1).*(b2-1)) / 2)-b2+1;
20     e3 = -((b3-2).^3)/6 + (((b3-1).*(b3-1)) / 2)-b3+1;
21     e4 = +((b4-3).^4)/24-((b4-2).^3)/6 + (((b4-1).*(b4-1)) / 2)
          -b4+1;
22
23     for i = 2:N
24         x(i) = x(i-1) + h * fgv_delay2(xn(1));
25         xn = circshift(xn,-1);
26         xn(end) = x(i);
27
28         k_1 = h * fgv_delay2(zn(1));
29         k_2 = h * fgv_delay2(zn(1)+h*k_1/2);
30         k_3 = h * fgv_delay2(zn(1)+h*k_2/2);
31         k_4 = h * fgv_delay2(zn(1)+h*k_3);
```

```
        z(i) = z(i-1) + (1/6)*(k_1+2*k_2+2*k_3+k_4);
        zn = circshift(zn,-1);
        zn(end) = z(i);
    %{
        %h=0.01
        if i <=100
        a(i) = x(i) - e1((10)*(i-1)+1);
        b(i) = z(i) - e1((10)*(i-1)+1);

        elseif i > 100 && i <=200
        a(i) = x(i) - e2((10)*(i-101)+1);
        b(i) = z(i) - e2((10)*(i-101)+1);

        elseif i >= 200 && i <=300
        a(i) = x(i) - e3((10)*(i-201)+1);
        b(i) = z(i) - e3((10)*(i-201)+1);

        elseif i>=300
        a(i) = x(i) - e4((10)*(i-301)+1);
        b(i) = z(i) - e4((10)*(i-301)+1);

        end
    %}
    end
    %{
        % h=1
        a(2) = x(2)-e1(1001);
        a(3) = x(3)-e2(1001);
        a(4) = x(4)-e3(1001);
        a(5) = x(5)-e4(1001);

        b(2) = z(2)-e1(1001);
        b(3) = z(3)-e2(1001);
        b(4) = z(4)-e3(1001);
        b(5) = z(5)-e4(1001);

    %}
%{
    %hibafuggvenyek
    plot([0:h:t],a,"green",'LineWidth',2)
    hold on
    plot([0:h:t],b,"cyan",'LineWidth',2)
    hold on
    title('Hibafuggvenyek','fontsize',15)
    xlabel('t','fontsize',10,'fontweight','bold')
    ylabel('e(t)','fontsize',10,'fontweight','bold')
    legend('Euler-modszer hibafuggveny','Runge-Kutta
       hibafuggveny')
```

```
%}
%{
    %kozelitesek
    plot([0:h:t],x,"red",'LineWidth',2)
    hold on
    axis([-1 t -1 x0+1])
    plot([-1 0],[1 1],"blue",'LineWidth',2)
    hold on
    plot([0:h:t],z,"magenta",'LineWidth',2)
    hold on
    plot([-1 0],[1 1],"blue",'LineWidth',2)
    hold on
    plot(b1,e1,"blue",'LineWidth',2)
    hold on
    plot(b2,e2,"blue",'LineWidth',2)
    hold on
    plot(b3,e3,"blue",'LineWidth',2)
    hold on
    plot(b4,e4,"blue",'LineWidth',2)

    title('Kozelito es pontos ertek osszehasonlitasa','fontsize
       ',15)
    xlabel('t','fontsize',10,'fontweight','bold')
    ylabel('y(t)','fontsize',10,'fontweight','bold')
    legend('kozelites Euler-modszerrel','pontos ertek','
       kozelites Runge-Kutta-modszerrel')
%}

end

function r = fgv_delay2(y)
    r =  - y;
end

% hiba_delay(1,4,1)
% hiba_delay(1,4,0.01)
```

6.3. Verhulst-féle modell

A kódban 5 függvényt találunk, az első kiszámolja és ábrázolja a megadott paraméterekkel a dolgozatban található Verhulst-féle modellhez tartózó értékeket. Az értékek meghatározására Euler-módszeres közelítést alkalmazunk.

A többi függvény a közelítésekhez szükséges segédfüggvényt tartalmazza.

```
function r = populacio(x0,t,h,a,K)

    x(1) = x0;
```

```
    N = floor(t/h)+1;
    hossz = linspace(0,t,t/h+1);
    zn = ones(1,1/h);
    z(1) = x0;

    for i = 2:N

        x(i) = x(i-1) + h * fgv_pop(a,x(i-1),K);
        x(i) = x(i-1) + h * fgv_pop(a,x(i-1),K);

        z(i) = z(i-1) + h * fgv_pop3(a,z(i-1),zn(1),K);
        zn = circshift(zn,-1);
        zn(end) = x(i);

    end

    plot(hossz,x,"red",'LineWidth',2)
    hold on
    plot(hossz,z,"blue",'LineWidth',2)
    %plot(hossz,z,"green",'LineWidth',2)
    %plot(hossz,z,"magenta",'LineWidth',2)
    hold on
    axis([0 t 0 z(end)+0.2])
    yline(K)

    title('Populacio egyedszamanak alakulasa','fontsize',15)
    xlabel('t','fontsize',10,'fontweight','bold')
    ylabel('y(t)','fontsize',10,'fontweight','bold')

end

function dy = fgv_pop(a,y,K)
    dy = a * y * (K-y);
end

function dy = fgv_pop1(a,y,tau,K)
    dy = a * y * (K-tau);
end

function dy = fgv_pop2(a,y,tau,K)
    dy = a * tau * (K-y);
end

function dy = fgv_pop3(a,y,tau,K)
    dy = a * tau * (K-tau);
end
```

```
52
53 % 1 - populacio(0.1,20,0.01,0.4,1)
54 % 2 - populacio(0.1,20,0.01,1.2,1)
55 % 3 - populacio(0.2,20,0.01,0.4,1)
56 % 4 - populacio(0.2,20,0.01,1.2,1)
```

6.4. Lotka–Volterra-féle zsákmány-ragadozó-modell

A kódban 3 függvényt találunk, az első kiszámolja és ábrázolja a megadott paraméterekkel a Lotka–Volterra-féle zsákmány-ragadozó-modellhez tartózó értékeket. Az értékek meghatározására Runge–Kutta-módszeres közelítést alkalmazunk.

A többi függvény a közelítésekhez szükséges segédfüggvényt tartalmazza.

```
1  unction r = rzs(u0,v0,t,h)
2      %u0 - zsakmany, v0 - ragadozo
3      y(1,1)=u0;
4      y(2,1)=v0;
5      z(1,1)=u0;
6      z(2,1)=v0;
7      N = floor(t/h)+1;
8      zn = ones(2,1/h);
9
10     for i = 2:N
11
12         k1 = func(y(:,i-1));
13         k2 = func(y(:,i-1)  + h/2 * k1);
14         k3 = func(y(:,i-1)  + h/2 * k2);
15         k4 = func(y(:,i-1)  + h   * k3);
16
17         y(:,i) = y(:,i-1) + h/6 * (k1 + 2 * k2 + 2 * k3 + k4);
18         %
19         k_1 = func_(z(:,i-1),zn(:,1));
20         k_2 = func_(z(:,i-1)+k_1*h/2,zn(:,1)+k_1*h/2);
21         k_3 = func_(z(:,i-1)+k_2*h/2,zn(:,1)+k_2*h/2);
22         k_4 = func_(z(:,i-1)+h * k_3,zn(:,1)+h * k_3);
23
24         z(:,i) = z(:,i-1) + (h/6)*(k_1+2*k_2+2*k_3+k_4);
25
26         zn(1,:) = circshift(zn(1,:),-1);
27         zn(2,:) = circshift(zn(2,:),-1);
28         zn(:,end) = z(:,i);
29
30     end
31      plot(y(2,:),y(1,:),"blue",'LineWidth',2);
32      axis([-0.5 5 -0.5 5])
33      hold on
34      %plot(z(2,:),z(1,:),"red",'LineWidth',2);
```

```
35      %axis([-0.5 5 -0.5 5])
36
37      title('Ragadozo-zsakmany modell','fontsize',15)
38      xlabel('zsakmany','fontsize',10,'fontweight','bold')
39      ylabel('ragadozo','fontsize',10,'fontweight','bold')
40      %legend('Kesleltetes nelkuli modell','Kesleltetest
          tartalmazo modell')
41  end
42
43  function dy = func(y)
44      dy(1,1) =  y(1,:) - y(1,:) * y(2,:);
45      dy(2,1) =  y(1,:) * y(2,:) - y(2,:);
46  end
47
48  function dy = func_(z,zn)
49      dy(1,1) =  zn(1,:) - z(1,:) * z(2,:);
50      dy(2,1) =  z(1,:) * z(2,:) - z(2,:);
51  end
52
53
54
55  % rzs(0.2,0.2,50,0.01)
```

Hivatkozások

[1] Alfredo Bellen and Marino Zennaro: Numerical Methods for Delay Differential Equations (Universitá di Trieste, Italy, 2003)

[2] Tóth János és Simon L. Péter: Differenciálegyenletek (Typotex, Budapest, 2020)

[3] Svantnerné Sebestyén Gabriella: Populációdinamikai modellek és matematikai vizsgálatuk, PHD dolgozat (ELTE, 2018)

[4] Fekete Imre: Alkalmazott Analízis órai jegyzet (ELTE, 2023)

[5] Faragó István – Horváth Róbert: Numerikus módszerek (Typotex, 2016)

[6] Online jegyzet a Konvergenciatulajdonságokról (2023.12.25.) https://gyires.inf.unideb.hu/GyBITT/02/ch05s09.html